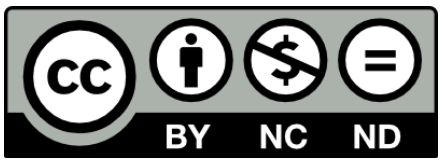
Modelos de Computación

Los Del DGIIM, losdeldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada



Modelos de Computación

Los Del DGIIM, losdeldgiim.github.io

Arturo Olivares Martos

Granada, 2024-2025

Índice general

1. Relaciones de Problemas

1.1. Introducción a la Computación

Ejercicio 1.1.1. Sea la gramática $G = (V, T, P, S)$ dada por:

$$V = \{S, X, Y\}$$
$$T = \{a, b\}$$
$$S = S$$

1. Describe el lenguaje generado por la gramática teniendo en cuenta que P viene descrito por:

$$S \to XYX$$
$$X \to aX \mid bX \mid \varepsilon$$
$$Y \to bbb$$

Sea $L = \{ubbbv \mid u, v \in \{a, b\}^*\}$. Demostraremos mediante doble inclusión que $L = \mathcal{L}(G)$.

$\subset$) Sea $w \in L$. Entonces, $w = ubbbv$ con $u, v \in \{a, b\}^*$. Veamos que $S \overset{*}{\Longrightarrow} w$:

$$S \Longrightarrow XYX \Longrightarrow XbbbX$$

Además, es fácil ver que la regla de producción $X \to aX \mid bX \mid \varepsilon$ nos permite generar cualquier palabra $u \in \{a, b\}^*$. Por tanto, tenemos que $X \overset{*}{\Longrightarrow} u$ y $X \overset{*}{\Longrightarrow} v$; teniendo así que $S \overset{*}{\Longrightarrow} ubbbv$.

$\supset$) Sea $w \in \mathcal{L}(G)$. Veamos la forma de w:

$$S \Longrightarrow XYX \Longrightarrow XbbbX \Longrightarrow ubbbv \mid u, v \in \{a, b\}^*$$

donde en el último paso hemos empleado lo visto en el apartado anterior de la regla de producción $X \to aX \mid bX \mid \varepsilon$. Por tanto, $w \in L$.

2. Describe el lenguaje generado por la gramática teniendo en cuenta que P viene descrito por:

$$S \to aX$$
$$X \to aX \mid bX \mid \varepsilon$$

Sea $L = \{au \mid u \in \{a, b\}^*\}$. Demostraremos mediante doble inclusión que $L = \mathcal{L}(G)$.

$\subset$) Sea $w \in L$. Entonces, $w = au$ con $u \in \{a,b\}^*$. Veamos que $S \overset{*}{\Longrightarrow} w$:

$$S \Longrightarrow aX \Longrightarrow au$$

donde en el último paso hemos empleado lo visto respecto a la regla de producción $X \to aX \mid bX \mid \varepsilon$. Por tanto, $w \in \mathcal{L}(G)$.

$\supset$) Sea $w \in \mathcal{L}(G)$. Veamos la forma de w:

$$S \Longrightarrow aX \Longrightarrow au \mid u \in \{a,b\}^*$$

donde en el último paso hemos empleado lo visto respecto a la regla de producción $X \to aX \mid bX \mid \varepsilon$. Por tanto, $w \in L$.

3. Describe el lenguaje generado por la gramática teniendo en cuenta que P viene descrito por:

$$S \to XaXaX$$
$$X \to aX \mid bX \mid \varepsilon$$

Sea $L = \{uavawa \mid u,v,w \in \{a,b\}^*\}$. Demostraremos mediante doble inclusión que $L = \mathcal{L}(G)$.

$\subset$) Sea $z \in L$. Entonces, $z = uavawa$ con $u,v,w \in \{a,b\}^*$. Veamos que $S \overset{*}{\Longrightarrow} z$:

$$S \Longrightarrow XaXaX \Longrightarrow uavawa$$

donde en el último paso hemos empleado lo visto respecto a la regla de producción $X \to aX \mid bX \mid \varepsilon$. Por tanto, $z \in \mathcal{L}(G)$.

$\supset$) Sea $z \in \mathcal{L}(G)$. Veamos la forma de z:

$$S \Longrightarrow XaXaX \Longrightarrow uavawa \mid u,v,w \in \{a,b\}^*$$

donde en el último paso hemos empleado lo visto respecto a la regla de producción $X \to aX \mid bX \mid \varepsilon$. Por tanto, $z \in L$.

4. Describe el lenguaje generado por la gramática teniendo en cuenta que P viene descrito por:

$$S \to SS \mid XaXaX \mid \varepsilon$$
$$X \to bX \mid \varepsilon$$

Sea el lenguaje $L = \{b^iab^jab^k \mid i,j,k \in \mathbb{N} \cup \{0\}\}$. Demostraremos mediante doble inclusión que $L^* = \mathcal{L}(G)$.

$\subset$) Sea $z \in L^* = \bigcup_{i \in \mathbb{N}} L^i$. Sea n el menor número natural tal que $z \in L^n$. Notando por $n_a(z)$ al número de a's en z, tenemos que $n_a(z) = 2n$. Entonces, $z \in L \cdot \ldots \cdot L$ (n veces), por lo que existen $i_1, j_1, k_1, \ldots, i_n, j_n, k_n \in \mathbb{N} \cup \{0\}$ tales que $z = b^{i_1}ab^{j_1}ab^{k_1} \cdot \ldots \cdot b^{i_n}ab^{j_n}ab^{k_n}$. Veamos que $S \overset{*}{\Longrightarrow} z$:

- Para conseguir el número de a's deseado, empleamos la regla de producción $S \to SS$ y reemplazamos una de las S por $XaXaX$. Esto lo hacemos n veces.
- Posteriormente, cada X la sustituiremos tantas veces como sea necesario por bX para conseguir el número de b's deseado en cada posición, y finalizaremos con $X \to \varepsilon$.

$\supset$) Sea $z \in \mathcal{L}(G)$, y sea $n_a(z)$ el número de a's en z. Entonces, como el número de a siempre aumenta de dos en dos, tenemos que $n_a(z) = 2n$ para algún $n \in \mathbb{N} \cup \{0\}$. Veamos la forma de z:

- Para llegar a z, hemos tenido que emplear la regla de producción $S \to SS \to SXaXaX$ n veces. Una vez llegados aquí, para eliminar la S (ya que habremos llegado a $n_a(z)$ a's), empleamos la regla de producción $S \to \varepsilon$.
- Posteriormente, para cada X, tan solo podemos emplear la regla de producción $X \to bX \mid \varepsilon$ para conseguir el número de b's deseado en cada posición.

Por tanto, es directo ver que $z \in L^n \subseteq L^*$.

Ejercicio 1.1.2. Sea la gramática $G = (V, T, P, S)$. Determinar en cada caso el lenguaje generado por la gramática.

1. Tenga en cuenta que:

$$\begin{aligned} V &= \{S, A\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{rcl} S & \to & abAS \mid a \\ abA & \to & baab \\ A & \to & b \end{array} \right\} \end{aligned}$$

Sea $L = \{ua \mid u \in \{abb, baab\}^*\}$. Demostraremos mediante doble inclusión que $L = \mathcal{L}(G)$.

$\subset$) Sea $w \in L$. Entonces, $w = ua$ con $u \in \{abb, baab\}^*$. Veamos que $S \overset{*}{\Longrightarrow} w$. Para ello, sabemos que $u \in \{abb, baab\}^* = \bigcup_{i \in \mathbb{N}} \{abb, baab\}^i$. Sea n el menor número natural tal que $u \in \{abb, baab\}^n$, es decir, es una concatenación de n subcadenas, cada una de las cuales es o bien abb o bien $baab$. Veamos que S produce ambas subcadenas:

- Para producir abb, tenemos que $S \to abAS \to abbS$.
- Para producir $baab$, tenemos que $S \to abAS \to baabS$.

Como vemos, en cada caso podemos concatenar la subcadena necesaria, pero siempre nos quedará una S al final. Usamos la regla de producción $S \to a$ para eliminarla, llegando así a w, por lo que $S \overset{*}{\Longrightarrow} w$ y $w \in \mathcal{L}(G)$.

$\supset$) Sea $w \in \mathcal{L}(G)$. Veamos la forma de w, para lo cual hay dos opciones:

- $S \to a$: En este caso, habremos finalizado la palabra con a, por lo que habremos añadido la subcadena a a la palabra al final.
- $S \to abAS$: En este caso, también hay dos opciones:
  - $S \to abAS \to baabS$: En este caso, habremos concatenado $baab$ con S, por lo que habremos añadido la subcadena $baab$ a la palabra.
  - $S \to abAS \to abbS$: En este caso, habremos concatenado abb con S, por lo que habremos añadido la subcadena abb a la palabra.

Por tanto, w es de la forma ua con u una concatenación de abb's y $baab$'s, es decir, $u \in \{abb, baab\}^*$. Por tanto, $w \in L$.

2. Tenga en cuenta que:

$$V = \{\langle\text{número}\rangle, \langle\text{dígito}\rangle\}$$
$$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$
$$S = \langle\text{número}\rangle$$
$$P = \left\{ \begin{array}{rcl} \langle\text{número}\rangle & \to & \langle\text{número}\rangle\langle\text{dígito}\rangle \\ \langle\text{número}\rangle & \to & \langle\text{dígito}\rangle \\ \langle\text{dígito}\rangle & \to & 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \end{array} \right\}$$

Tenemos que $\mathcal{L}(G)$ es el conjunto de los números naturales, permitiendo tantos ceros a la izquierda como se quiera. Es decir (usando la notación de potencia y concatenación vista para lenguajes):

$$L = \{0^i n \mid i \in \mathbb{N} \cup \{0\},\ n \in \mathbb{N} \cup \{0\}\}$$

Demostrémoslo mediante doble inclusión que $L = \mathcal{L}(G)$.

$\subset$) Sea $w \in L$. Entonces, $w = 0^i n$ con $i \in \mathbb{N} \cup \{0\}$ y $n \in \mathbb{N} \cup \{0\}$. Veamos que $\langle\text{número}\rangle \overset{*}{\Longrightarrow} w$:

- En primer lugar, aplicamos $|w| - 1$ veces la regla de producción $\langle\text{número}\rangle \to \langle\text{número}\rangle\langle\text{dígito}\rangle$ y la regla que lleva de $\langle\text{dígito}\rangle$ a uno de los símbolos terminales, consiguiendo así en cada etapa reemplazar la última variable presente en la cadena por un dígito.
- Finalmente, aplicamos la regla de producción $\langle\text{número}\rangle \to \langle\text{dígito}\rangle$ para reemplazar la última variable por un dígito, que será el primero del número formado.

Por tanto, $\langle\text{número}\rangle \overset{*}{\Longrightarrow} w$, teniendo que $w \in \mathcal{L}(G)$.

$\supset$) Sea $w \in \mathcal{L}(G)$. Como la única regla que aumenta la longitud es la regla de producción $\langle\text{número}\rangle \to \langle\text{número}\rangle\langle\text{dígito}\rangle$, tenemos que w tiene la forma:

$$\begin{aligned} \langle\text{número}\rangle &\Longrightarrow \langle\text{número}\rangle\langle\text{dígito}\rangle \overset{|w|-1 \text{ veces}}{\Longrightarrow} \\ &\Longrightarrow \langle\text{número}\rangle\langle\text{dígito}\rangle\langle\text{dígito}\rangle \overset{|w|-1 \text{ veces}}{\cdots} \langle\text{dígito}\rangle \Longrightarrow \\ &\Longrightarrow \langle\text{dígito}\rangle \overset{|w| \text{ veces}}{\cdots} \langle\text{dígito}\rangle \end{aligned}$$

Por tanto, tenemos que se trata una sucesión de $|w|$ dígitos, lo que nos lleva a que $w \in L$.

3. Tenga en cuenta que:

$$V = \{A, S\}$$
$$T = \{a, b\}$$
$$S = S$$
$$P = \left\{ \begin{array}{lcl} S & \to & aS \mid aA \\ A & \to & bA \mid b \end{array} \right\}$$

Sea $L = \{a^n b^m \in \{a, b\}^* \mid n, m \in \mathbb{N}\}$. Demostraremos mediante doble inclusión que $L = \mathcal{L}(G)$.

$\subset$) Sea $w \in L$. Entonces, $w = a^n b^m$ con $n, m \in \mathbb{N}$. Veamos que $S \overset{*}{\Longrightarrow} w$:

- En primer lugar, aplicamos $n-1$ veces la regla de producción $S \to aS$ para obtener $a^{n-1}S$,
$$S \overset{*}{\Longrightarrow} a^{n-1}S$$
- Para cambiar a la etapa de añadir b's, aplicamos la regla de producción $S \to aA$, obteniendo así $a^n A$,
- Después, aplicamos $m-1$ veces la regla de producción $A \to bA$ para obtener $a^n b^{m-1} A$.
- Para finalizar, aplicamos la regla de producción $A \to b$ para obtener $a^n b^m$.

Por tanto, $S \overset{*}{\Longrightarrow} w$, teniendo que $w \in \mathcal{L}(G)$.

$\supset$) Sea $w \in \mathcal{L}(G)$. Vemos que en la palabra siempre va a haber tan solo una variable (ya sea S o A). Se empezará con la S, y en cierto momento se cambiará a la A, sin poder entonces volver a la S.

- Cuando se está en la etapa en la que hay S, tan solo se pueden añadir a's, o bien cambiar a la A.
- Cuando se está en la etapa en la que hay A, tan solo se pueden añadir b's.

Por tanto, tenemos que w estará formada por una sucesión de a's seguida de una sucesión de b's, lo que nos lleva a que $w \in L$.

Ejercicio 1.1.3. Encontrar gramáticas de tipo 2 para los siguientes lenguajes sobre el alfabeto $\{a, b\}$. En cada caso determinar si los lenguajes generados son de tipo 3, estudiando si existe una gramática de tipo 3 que los genera.

1. Palabras en las que el número de b no es tres.

   Tenemos varias opciones:

   - Que no tenga b's.
   - Que tenga una b.
   - Que tenga dos b's.
   - Que tenga 4 o más b's.

Sea la gramática $G = (V, T, P, S)$ dada por:

$$
\begin{aligned}
V &= \{S, A, X\} \\
T &= \{a, b\} \\
S &= S \\
P &= \left\{ \begin{array}{rcl} S & \to & A \mid AbA \mid AbAbA \mid XbXbXbXbX \\ A & \to & aA \mid \varepsilon \\ X & \to & aX \mid bX \mid \varepsilon \end{array} \right\}
\end{aligned}
$$

Esta gramática no obstante es de tipo 2. Busquemos otra que sea de tipo 3. Sea la gramática $G' = (V', T', P', S')$ dada por:

$$
\begin{aligned}
V' &= \{S, X, Y, Z, W\} \\
T' &= \{a, b\} \\
S' &= S \\
P' &= \left\{ \begin{array}{rcl} S & \to & \varepsilon \mid aS \mid bX \\ X & \to & \varepsilon \mid aX \mid bY \\ Y & \to & \varepsilon \mid aY \mid bZ \\ Z & \to & aZ \mid bW \\ W & \to & \varepsilon \mid aW \mid bW \end{array} \right\}
\end{aligned}
$$

Esta sí es de tipo 3, y genera el lenguaje deseado.

2. Palabras que tienen 2 ó 3 b.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$
\begin{aligned}
V &= \{S, A, B\} \\
T &= \{a, b\} \\
S &= S \\
P &= \left\{ \begin{array}{rcl} S & \to & AbAbABA \\ A & \to & aA \mid \varepsilon \\ B & \to & b \mid \varepsilon \end{array} \right\}
\end{aligned}
$$

   Esta gramática no obstante es de tipo 2. Busquemos otra que sea de tipo 3. Sea la gramática $G' = (V', T', P', S')$ dada por:

$$
\begin{aligned}
V' &= \{S, X, Y, Z, W, V, T\} \\
T' &= \{a, b\} \\
S' &= S \\
P' &= \left\{ \begin{array}{rcl} S & \to & aS \mid X \\ X & \to & bY \\ Y & \to & aY \mid Z \\ Z & \to & bW \\ W & \to & aW \mid \varepsilon \mid V \\ V & \to & bT \\ T & \to & aT \mid \varepsilon \end{array} \right\}
\end{aligned}
$$

Esta gramática ya es de tipo 3, pero contiene un número elevado de variables. Veamos si podemos reducirlo: Sea la gramática $G'' = (V'', T'', P'', S'')$ dada por:

$$\begin{aligned} V'' &= \{S, X, Y, Z\} \\ T'' &= \{a, b\} \\ S'' &= S \\ P'' &= \left\{ \begin{array}{lcl} S & \to & aS \mid bX \\ X & \to & aX \mid bY \\ Y & \to & aY \mid \varepsilon \mid bZ \\ Z & \to & aZ \mid \varepsilon \end{array} \right\} \end{aligned}$$

Notemos que, en esta gramática de tipo 3, ya hemos conseguido el menor número de variables posibles, que representan las 4 etapas. Como la última es opcional, está la regla $Y \to \varepsilon$, para así no agregar la tercera b.

Ejercicio 1.1.4. Encontrar gramáticas de tipo 2 para los siguientes lenguajes sobre el alfabeto $\{a, b\}$. En cada caso determinar si los lenguajes generados son de tipo 3, estudiando si existe una gramática de tipo 3 que los genera.

1. Palabras que no contienen la subcadena ab.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, A\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{lcl} S & \to & aA \mid bS \mid \varepsilon \\ A & \to & aA \mid \varepsilon \end{array} \right\} \end{aligned}$$

   Notemos además que esta gramática es de tipo 3, y se tiene que:

$$\mathcal{L}(G) = \{b^i a^j \mid i, j \in \mathbb{N} \cup \{0\}\}$$

2. Palabras que no contienen la subcadena baa.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, B\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{lcl} S & \to & aS \mid bB \mid \varepsilon \\ B & \to & bB \mid abB \mid a \mid \varepsilon \end{array} \right\} \end{aligned}$$

   Notemos además que esta gramática es de tipo 3.

Ejercicio 1.1.5. Encontrar una gramática libre de contexto que genere el lenguaje sobre el alfabeto $\{a, b\}$ de las palabras que tienen más a que b (al menos una más).

Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, S'\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{rcl} S & \to & S'aS' \\ S' & \to & S'aS' \mid aS'bS' \mid bS'aS' \mid \varepsilon \end{array} \right\} \end{aligned}$$

Ejercicio 1.1.6. Encontrar, si es posible, una gramática regular (o, si no es posible, una gramática libre del contexto) que genere el lenguaje L supuesto que $L \subset \{a, b\}^*$ y verifica:

1. $u \in L$ si, y solamente si, verifica que u no contiene dos símbolos b consecutivos.

   Sea la gramática $G = (V, T, P, S)$ dada por:

   $$\begin{aligned} V &= \{S\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{rcl} S & \to & aS \mid baS \mid b \mid \varepsilon \end{array} \right\} \end{aligned}$$

2. $u \in L$ si, y solamente si, verifica que u contiene dos símbolos b consecutivos.

   Sea la gramática $G = (V, T, P, S)$ dada por:

   $$\begin{aligned} V &= \{S, B, F\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{rcl} S & \to & aS \mid bB \\ B & \to & bF \mid aS \\ F & \to & aF \mid bF \mid \varepsilon \end{array} \right\} \end{aligned}$$

   Notemos que, en este caso, tenemos tres estados:

   - S: No hemos encontrado dos b's consecutivas.
   - B: Hemos encontrado una b, y puede ser que nos encontremos la segunda b.
   - F: Hemos encontrado dos b's consecutivas; ya hay libertad.

   Sí es cierto que usamos tres variables. Para usar solo dos variables, podemos hacer lo siguiente. Sea la gramática $G' = (V', T', P', S')$ dada por:

   $$\begin{aligned} V' &= \{S, X\} \\ T' &= \{a, b\} \\ S' &= S \\ P' &= \left\{ \begin{array}{rcl} S & \to & aS \mid bS \mid bbX \\ X & \to & aX \mid bX \mid \varepsilon \end{array} \right\} \end{aligned}$$

Ejercicio 1.1.7. Encontrar, si es posible, una gramática regular (o, si no es posible, una gramática libre del contexto) que genere el lenguaje L supuesto que $L \subset \{a, b\}^*$ y verifica:

1. $u \in L$ si, y solamente si, verifica que contiene un número impar de símbolos a.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, X\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{lcl} S & \to & aX \mid bS \\ X & \to & aS \mid bX \mid \varepsilon \end{array} \right\} \end{aligned}$$

2. $u \in L$ si, y solamente si, verifica que no contiene el mismo número de símbolos a que de símbolos b.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, A, B, X\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{lcl} S & \to & AaA \mid BbB \\ A & \to & AaA \mid X \\ B & \to & BbB \mid X \\ X & \to & aXbX \mid bXaX \mid \varepsilon \end{array} \right\} \end{aligned}$$

Ejercicio 1.1.8. Dado el alfabeto $A = \{a, b\}$ determinar si es posible encontrar una gramática libre de contexto que:

1. Genere las palabras de longitud impar, y mayor o igual que 3, tales que la primera letra coincida con la letra central de la palabra.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, X, A, B, C, D\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{lcl} S & A \mid B & \\ A & \to & aCX \\ C & \to & a \mid XCX \\ B & \to & bDX \\ D & \to & b \mid XDX \\ X & \to & a \mid b \end{array} \right\} \end{aligned}$$

2. Genere las palabras de longitud par, y mayor o igual que 2, tales que las dos letras centrales coincidan.

Sea la gramática $G = (V, T, P, S)$ dada por:

$$V = \{S, X\}$$
$$T = \{a, b\}$$
$$S = S$$
$$P = \left\{ \begin{array}{rcl} S & \to & XSX \mid C \\ C & \to & aa \mid bb \\ X & \to & a \mid b \end{array} \right\}$$

Ejercicio 1.1.9. Sea la gramática $G = (V, T, P, S)$ dada por:

$$V = \{S, X\}$$
$$T = \{a, b\}$$
$$S = S$$
$$P = \left\{ \begin{array}{rcl} S & \to & SS \\ S & \to & XXX \\ X & \to & aX \mid Xa \mid b \end{array} \right\}$$

Determinar si el lenguaje generado por la gramática es regular. Justificar la respuesta.

Sea la siguiente gramática regular $G' = (V', T', P', S')$ dada por:

$$V' = \{S, X\}$$
$$T' = \{a, b\}$$
$$S' = S$$
$$P' = \left\{ \begin{array}{rcl} S & \to & aS \mid bX \\ X & \to & aX \mid bY \\ Y & \to & aY \mid bZ \\ Z & \to & aZ \mid bW \mid \varepsilon \\ W & \to & aW \mid bU \\ U & \to & aU \mid bV \\ V & \to & aV \mid \varepsilon \end{array} \right\}$$

Tenemos que $\mathcal{L}(G) = \mathcal{L}(G')$, y como G' es una gramática regular, tenemos que $\mathcal{L}(G)$ es regular. Sí es cierto que en el tema 2 aprendemos otras maneras de demostrarlo más sencillas, como buscar un autómata finito que lo genere.

Ejercicio 1.1.10. Dado un lenguaje L sobre un alfabeto A, ¿es L^* siempre numerable? ¿nunca lo es? ¿o puede serlo unas veces sí y otras, no? Pon ejemplos en este último caso.

L^* es siempre numerable, veámos por qué. L^* es un lenguaje sobre el alfabeto A, por lo que $L^* \subseteq A^*$ y A^* es numerable (visto en teoría), luego L^* también lo es.

Ejercicio 1.1.11. Dado un lenguaje L sobre un alfabeto A, caracterizar cuando $L^* = L$. Esto es, dar un conjunto de propiedades sobre L de manera que L cumpla esas propiedades si y sólo si $L^* = L$.

$$L = L^* \iff \begin{cases} \varepsilon \in L \\ \wedge \\ u, v \in L \implies uv \in L \end{cases}$$

Es decir, $L = L^*$ si y solo si la cadena vacía está en L y además es cerrado para concatenaciones.

Demostración. Demostramos mediante doble implicación.

$\Longleftarrow$) La inclusión $L \subseteq L^*$ es obvia, por lo que solo falta demostrar la otra inclusión.

Sea $v \in L^*$:

1. Si $v = \varepsilon \implies v \in L$ por hipótesis.
2. Si $v \neq \varepsilon$, $\exists n \in \mathbb{N}$ tal que
$$v = a_1 a_2 \dots a_n$$
con $a_i \in L \ \forall i \in \{1, \dots, n\}$, de donde tenemos que $v \in L$, por ser cerrado para concatenaciones. Luego $L^* \subseteq L$.

$\Longrightarrow$) Hemos de probar dos cosas:

1. $\varepsilon \in L^* = L$.
2. Sean $u, v \in L = L^* \implies uv \in L^* = L$.

□

Ejercicio 1.1.12. Dados dos homomorfismos $f : A^* \to B^*$, $g : A^* \to B^*$, se dice que son iguales si $f(x) = g(x)$, $\forall x \in A^*$. ¿Existe un procedimiento algorítmico para comprobar si dos homomorfismos son iguales?

Sí, basta probar que su imagen coincide sobre un conjunto finito de elementos, los de A:

$$f(x) = g(x) \quad \forall x \in A^* \iff f(a) = g(a) \quad \forall a \in A$$

Demostración.

$\Longleftarrow$) Sea $v \in A^*$, $\exists n \in \mathbb{N}$ tal que $v = a_1 a_2 \dots a_n$ con $a_i \in A \ \forall i \in \{1, \dots, n\}$

$$f(v) = f(a_1) f(a_2) \dots f(a_n) = g(a_1) g(a_2) \dots g(a_n) = g(v)$$

$\Longrightarrow$) Sea $a \in A \implies a \in A^* \implies f(a) = g(a)$.

□

Ejercicio 1.1.13. Sea $L \subseteq A^*$ un lenguaje arbitrario. Sea $C_0 = L$ y definamos los lenguajes S_i y C_i, para todo $i \geqslant 1$, por $S_i = C_{i-1}^+$ y $C_i = \overline{S_i}$.

1. ¿Es S_1 siempre, nunca o a veces igual a C_2? Justifica la respuesta.
2. Demostrar que $S_2 = C_3$, cualquiera que sea L.

Observación. Demuestra que C_2 es cerrado para la concatenación.

Ejercicio 1.1.14. Demuestra que, para todo alfabeto A, el conjunto de los lenguajes finitos sobre dicho alfabeto es numerable.

Sea $A = \{a_1, a_2, \ldots, a_n\}$, con $n \in \mathbb{N}$. Definimos el siguiente conjunto:

$$\Gamma = \{L \subseteq A^* \mid L \text{ es finito}\}$$

Dado un símbolo $z \notin A$, definimos el conjunto $B = \{z\} \cup A$. Sea B^* numerable, y buscamos una inyección de Γ en B^*. Dado un lenguaje $L \in \Gamma$, sea $L = \{l_1, l_2, \ldots, l_m\}$, con $m \in \mathbb{N}$ y $l_i \in A^*$ $\forall i \in \{1, \ldots, m\}$. Definimos la siguiente función:

$$\begin{array}{rccc} f: & \Gamma & \longrightarrow & B^* \\ & L & \longmapsto & zl_1zl_2 \ldots zl_mz \end{array}$$

Veamos que f es inyectiva. Sean $L_1, L_2 \in \Gamma$ tales que $f(L_1) = f(L_2)$. Entonces,

$$zl_1zl_2 \ldots zl_kz = zl'_1zl'_2 \ldots zl'_{k'}z$$

Por ser ambas palabras iguales, tenemos que $k = k'$ y $l_i = l'_i$ $\forall i \in \{1, \ldots, k\}$, de donde $L_1 = L_2$. Por tanto, f es inyectiva, por lo que Γ es inyectivo con un subconjunto de B^*, que es numerable. Por tanto, Γ es numerable.

1.1.1. Cálculo de gramáticas

Ejercicio 1.1.15 (Complejidad: Sencilla)**.** Calcula, de forma razonada, gramáticas que generen cada uno de los siguientes lenguajes:

1. $\{u \in \{0,1\}^* \mid |u| \leqslant 4\}$

   Sea la gramática $G = (V, T, P, S)$ dada por:

   $$\begin{aligned} V &= \{S, X\} \\ T &= \{0, 1\} \\ S &= S \\ P &= \left\{ \begin{array}{rcl} S & \to & XXXX \\ X & \to & 0 \mid 1 \mid \varepsilon \end{array} \right\} \end{aligned}$$

   No obstante, esta gramática es de tipo 2. Busquemos una de tipo 3. Sea la gramática $G' = (V', T', P', S')$ dada por:

   $$\begin{aligned} V' &= \{S, X, Y, Z\} \\ T' &= \{0, 1\} \\ S' &= S \\ P' &= \left\{ \begin{array}{rcl} S & \to & 0X \mid 1X \mid \varepsilon \\ X & \to & 0Y \mid 1Y \mid \varepsilon \\ Y & \to & 0Z \mid 1Z \mid \varepsilon \\ Z & \to & 0 \mid 1 \end{array} \right\} \end{aligned}$$

   Tenemos que $\mathcal{L}(G) = \mathcal{L}(G')$, y es igual al lenguaje deseado. Tenemos por tanto que es un lenguaje regular.

2. Palabras con 0's y 1's que no contengan dos 1's consecutivos y que empiecen por un 1 y que terminen por dos 0's.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$
\begin{aligned}
V &= \{S, X, Y\} \\
T &= \{0, 1\} \\
S &= S \\
P &= \left\{ \begin{array}{lcl} S & \to & 1X00 \\ X & \to & 0Y \mid \varepsilon \\ Y & \to & 0Y \mid 1X \mid \varepsilon \end{array} \right\}
\end{aligned}
$$

   Notemos que esta gramática es de tipo 2 debido a la primera regla de producción. Busquemos una de tipo 3. Sea la gramática $G' = (V', T', P', S')$ dada por:

$$
\begin{aligned}
V' &= \{S, X, Y\} \\
T' &= \{0, 1\} \\
S' &= S \\
P' &= \left\{ \begin{array}{lcl} S & \to & 1X \\ X & \to & 0Y \mid F \\ Y & \to & 0Y \mid 1X \mid F \\ F & \to & 00 \end{array} \right\}
\end{aligned}
$$

   Tenemos que $\mathcal{L}(G) = \mathcal{L}(G')$, y es igual al lenguaje deseado. Tenemos por tanto que es un lenguaje regular. En esta última gramática, tenemos los siguientes estados:

   - S: Es el estado inicial, empezamos con un 1.
   - X: Acabamos de escribir un 1, por lo que ahora tan solo podemos escribir 0's.
   - Y: Acabamos de escribir un 0, por lo que ahora podemos escribir tanto 0's como 1's.
   - F: Ya hemos terminado, y escribimos los dos 0's finales por la restricción impuesta.

3. El conjunto vacío.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$
\begin{aligned}
V &= \{S\} \\
T &= \emptyset \\
S &= S \\
P &= \left\{ \begin{array}{lcl} S & \to & S \end{array} \right\}
\end{aligned}
$$

4. El lenguaje formado por los números naturales.

Sea la gramática $G = (V, T, P, S)$ dada por:

$$V = \{\langle\text{número no iniciado}\rangle, \langle\text{dígito no cero}\rangle, \langle\text{dígito}\rangle, \langle\text{número iniciado}\rangle\}$$
$$T = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$
$$S = \langle\text{número no iniciado}\rangle$$
$$P = \left\{ \begin{array}{rcl} \langle\text{número no iniciado}\rangle & \to & \langle\text{dígito no cero}\rangle \mid \langle\text{dígito no cero}\rangle\langle\text{número iniciado}\rangle \\ \langle\text{número iniciado}\rangle & \to & \langle\text{dígito}\rangle \mid \langle\text{dígito}\rangle\langle\text{número iniciado}\rangle \\ \langle\text{dígito no cero}\rangle & \to & 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\ \langle\text{dígito}\rangle & \to & 0 \mid \langle\text{dígito no cero}\rangle \end{array} \right\}$$

Notemos que esta gramática es similar a la descrita en el Ejercicio 1.1.2.2, pero adaptada para que los números naturales no puedan empezar por 0. No obstante, esta gramática es de tipo 2. Busquemos una de tipo 3. Sea la gramática $G' = (V', T', P', S')$ dada por:

$$V' = \{S, X, Y, Z\}$$
$$T' = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$
$$S' = S$$
$$P' = \left\{ \begin{array}{rcl} S & \to & 0 \mid 1N \mid 2N \mid 3N \mid 4N \mid 5N \mid 6N \mid 7N \mid 8N \mid 9N \\ N & \to & 0N \mid 1N \mid 2N \mid 3N \mid 4N \mid 5N \mid 6N \mid 7N \mid 8N \mid 9N \mid \varepsilon \end{array} \right\}$$

5. $\{a^n \in \{a, b\}^* \mid n \geqslant 0\} \cup \{a^n b^n \in \{a, b\}^* \mid n \geqslant 0\}$

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$V = \{S, X, Y\}$$
$$T = \{a, b\}$$
$$S = S$$
$$P = \left\{ \begin{array}{rcl} S & \to & X \mid Y \mid \varepsilon \\ X & \to & aX \mid \varepsilon \\ Y & \to & aYb \mid \varepsilon \end{array} \right\}$$

6. $\{a^n b^{2n} c^m \in \{a, b, c\}^* \mid n, m > 0\}$

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$V = \{S, X, Y, Z\}$$
$$T = \{a, b, c\}$$
$$S = S$$
$$P = \left\{ \begin{array}{rcl} S & \to & aXbbcY \\ X & \to & aXbb \mid \varepsilon \\ Y & \to & cY \mid \varepsilon \end{array} \right\}$$

7. $\{a^n b^m a^n \in \{a, b\}^* \mid m, n \geqslant 0\}$

Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, X\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{lcl} S & \to & aSa \mid bX \mid \varepsilon \\ X & \to & bX \mid \varepsilon \end{array} \right\} \end{aligned}$$

8. Palabras con 0's y 1's que contengan la subcadena 00 y 11.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, X\} \\ T &= \{0, 1\} \\ S &= S \\ P &= \left\{ \begin{array}{lcl} S & \to & X00X11X \mid X11X00X \\ X & \to & 0X \mid 1X \mid \varepsilon \end{array} \right\} \end{aligned}$$

   Notemos que esta gramática es de tipo 2. Busquemos una de tipo 3. Sea la gramática $G' = (V', T', P', S')$ dada por:

$$\begin{aligned} V' &= \{S, X, A, B, F\} \\ T' &= \{0, 1\} \\ S' &= S \\ P' &= \left\{ \begin{array}{lcl} S & \to & 0S \mid 1S \mid X \\ X & \to & 00A \mid 11B \\ A & \to & 0A \mid 1A \mid 11F \\ B & \to & 0B \mid 1B \mid 00F \\ F & \to & 0F \mid 1F \mid \varepsilon \end{array} \right\} \end{aligned}$$

   Notemos que:

   - S: No hemos encontrado ninguna subcadena.
   - X: Hemos encontrado una subcadena, y ahora buscamos la otra.
   - A: Hemos encontrado la subcadena 00, y ahora buscamos la subcadena 11.
   - B: Hemos encontrado la subcadena 11, y ahora buscamos la subcadena 00.
   - F: Hemos encontrado ambas subcadenas.

9. Palíndromos formados con las letras a y b.

   Sea la gramática $G = (V, T, P, S)$ dada por:

$$\begin{aligned} V &= \{S, X, Y\} \\ T &= \{a, b\} \\ S &= S \\ P &= \left\{ \begin{array}{lcl} S & \to & aSa \mid bSb \mid \varepsilon \mid a \mid b \end{array} \right\} \end{aligned}$$

Notemos que las reglas $S \to a \mid b$ se han añadido para añadir los palíndromos de longitud impar.

Ejercicio 1.1.16 (Complejidad: Media)**.** Calcula, de forma razonada, gramáticas que generen cada uno de los siguientes lenguajes:

1. $\{uv \in \{0,1\}^* \mid u^{-1}$ es un prefijo de $v\}$

   Sea la gramática $G = (V, T, P, S)$ dada por:

   $$V = \{S, X, Y\}$$
   $$T = \{0, 1\}$$
   $$S = S$$
   $$P = \left\{ \begin{array}{lcl} S & \to & XY \\ X & \to & 0X0 \mid 1X1 \mid \varepsilon \\ Y & \to & 0Y \mid 1Y \mid \varepsilon \end{array} \right\}$$

   Notemos que X deriva en el palíndromo, uu^{-1}, y Y en el resto de la palabra de v.

2. $\{ucv \in \{a,b,c\}^* \mid |u| = |v|\}$

   Sea la gramática $G = (V, T, P, S)$ dada por:

   $$V = \{S, X\}$$
   $$T = \{a, b, c\}$$
   $$S = S$$
   $$P = \left\{ \begin{array}{lcl} S & \to & XSX \mid c \\ X & \to & a \mid b \mid c \end{array} \right\}$$

3. $\{u1^n \in \{0,1\}^* \mid |u| = n\}$

   Sea la gramática $G = (V, T, P, S)$ dada por:

   $$V = \{S, X\}$$
   $$T = \{0, 1\}$$
   $$S = S$$
   $$P = \left\{ \begin{array}{lcl} S & \to & XS1 \mid \varepsilon \\ X & \to & 0 \mid 1 \end{array} \right\}$$

4. $\{a^n b^n a^{n+1} \in \{a,b\}^* \mid n \geqslant 0\}$ (observar transparencias de teoría)

   Sea la gramática $G = (V, T, P, S)$ dada por:

   $$V = \{S, X, Y\}$$
   $$T = \{a, b\}$$
   $$S = S$$
   $$P = \left\{ \begin{array}{lcl} S & \to & a \mid abaa \mid aXbaa \\ Xb & \to & bX \\ Xa & \to & Ybaa \\ bY & \to & Yb \\ aY & \to & aa \mid aaX \end{array} \right\}$$

Ejercicio 1.1.17 (Complejidad: Difícil)**.** Calcula, de forma razonada, gramáticas que generen cada uno de los siguientes lenguajes:

1. $\{a^n b^m c^k \in \{a,b,c\}^* \mid k = m + n\}$
   Sea la gramática $G = (V, T, P, S)$ dada por:
   $$\begin{aligned} V &= \{S, X\} \\ T &= \{a, b, c\} \\ S &= S \\ P &= \left\{ \begin{array}{rcl} S & \to & aSc \mid X \\ X & \to & bXc \mid \varepsilon \end{array} \right\} \end{aligned}$$
2. Palabras que son múltiplos de 7 en binario.

Ejercicio 1.1.18 (Complejidad: Extrema (no son libres de contexto))**.** Calcula, de forma razonada, gramáticas que generen cada uno de los siguientes lenguajes:

1. $\{ww \mid w \in \{0,1\}^*\}$
   Para este lenguaje, hemos construido la gramática $G = (V, T, P, S)$ dada por:
   $$\begin{aligned} V &= \{S, \alpha, \beta, \gamma, X, E, E_1, E_0, E', B\} \\ T &= \{0, 1\} \\ S &= S \end{aligned}$$
   P que contiene las reglas de producción que se mostrarán a continuación.

   La idea principal en la gramática es generar entre las variables α y β cualqueir palabra del lenguaje $\{0,1\}^*$. Posteriormente, iremos copiando dicha palabra a la derecha de β usando para ello las variables E y γ, de forma que con γ controlaremos la parte de la palabra de la izquierda que ya hayamos copiado a la derecha de β.

   Finalmente, usaremos B para eliminar cualquier rastro de las variable auxiliares. De esta forma, las reglas de P son:

   - Para generar cualquier palabra entre α y β:
     $$\begin{aligned} S &\to \alpha X \beta \\ X &\to 0X \mid 1X \mid E\gamma \end{aligned}$$
   - Para coger un 1 y copiarlo a la derecha:
     Hemos de estar al final de la parte de la palabra no copiada (luego ha de ser $xE\gamma$ siendo x 0 o 1). Posteriormente, avanzamos γ a la izquierda para indicar que dicho 1 ya está copiado y cambiamos a la variable que transporta el 1 a la derecha:
     $$1E\gamma \to \gamma 1 E_1$$
     Posteriormente, movemos dicha variable a la derecha:
     $$\begin{aligned} E_1 1 &\to 1 E_1 \\ E_1 0 &\to 0 E_1 \end{aligned}$$

Cuando lleguemos al final de la palabra de la izquierda, soltamos el 1 al inicio de la palabra de la derecha:

$$E_1\beta \to E'\beta 1$$

- Para coger un 0 y copiarlo a la derecha, es una situación análoga pero usamos otra variable:

$$0E\gamma \to \gamma 0E_0$$

$$E_0 1 \to 1E_0$$
$$E_0 0 \to 0E_0$$

$$E_0\beta \to E'\beta 0$$

- Ahora, explicamos E', cuya única funcionalidad es volver al final de la parte no copiada de la palabra de la izquierda:

$$1E' \to E'1$$
$$0E' \to E'0$$
$$\gamma E' \to E\gamma$$

- La copia de la palabra terminará cuando se de $\alpha E\gamma$ (ya que estará toda la palabra copiada a la derecha). En dicho caso, eliminamos todas las variables auxiliares restantes:

$$\alpha E\gamma \to B$$
$$B1 \to 1B$$
$$B0 \to 0B$$
$$B\beta \to \varepsilon$$

Puede demostrarse que el lenguaje generado por esta gramática es el solicitado. Por la complejidad de la gramática, nos limitamos a mostrar un ejemplo para ver de forma intuitiva el buen funcionamiento de la misma.

Trataremos de generar la cadena: 10111011 (es decir, $(1011)^2$):

$$\begin{aligned}
S &\to \alpha X\beta \to \alpha 1X\beta \to \alpha 10X\beta \to \alpha 101X\beta \to \alpha 1011X\beta \to \alpha 1011E\gamma\beta \to \\
&\to \alpha 101\gamma 1E_1\beta \to \alpha 101\gamma 1E'\beta 1 \to \alpha 101\gamma E'1\beta 1 \to \alpha 101E\gamma 1\beta 1 \to \\
&\to \alpha 10\gamma 1E_1 1\beta 1 \to \alpha 10\gamma 11E_1\beta 1 \to \alpha 10\gamma 11E'\beta 11 \to \alpha 10\gamma 1E'1\beta 11 \to \\
&\to \alpha 10\gamma E'11\beta 11 \to \alpha 10E\gamma 11\beta 11 \to \alpha 1\gamma 0E_0 11\beta 11 \to \alpha 1\gamma 01E_0 1\beta 11 \to \\
&\to \alpha 1\gamma 011E_0\beta 11 \to \alpha 1\gamma 011E'\beta 011 \to \alpha 1\gamma 01E'1\beta 011 \to \alpha 1\gamma 0E'11\beta 011 \to \\
&\to \alpha 1\gamma E'011\beta 011 \to \alpha 1E\gamma 011\beta 011 \to \alpha\gamma 1E_1 011\beta 011 \to \alpha\gamma 10E_1 11\beta 011 \to \\
&\to \alpha\gamma 101E_1 1\beta 011 \to \alpha\gamma 1011E_1\beta 011 \to \alpha\gamma 1011E'\beta 1011 \to \alpha\gamma 101E'1\beta 1011 \to \\
&\to \alpha\gamma 10E'11\beta 1011 \to \alpha\gamma 1E'011\beta 1011 \to \alpha\gamma E'1011\beta 1011 \to \alpha E\gamma 1011\beta 1011 \to \\
&\to B1011\beta 1011 \to 1B011\beta 1011 \to 10B11\beta 1011 \to 101B1\beta 1011 \to \\
&\to 1011B\beta 1011 \to 10111011
\end{aligned}$$

2. $\{a^{n^2} \in \{a\}^* \mid n \geqslant 0\}$
3. $\{a^p \in \{a\}^* \mid p \text{ es primo}\}$
4. $\{a^n b^m \in \{a, b\}^* \mid n \leqslant m^2\}$

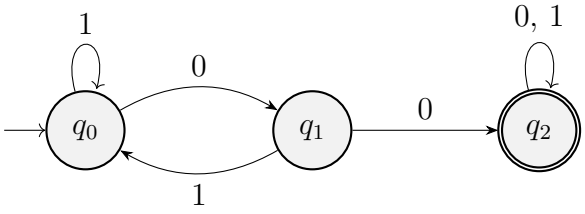


Figura 1.1: Autómata Finito Determinista del Ejercicio 1.2.1

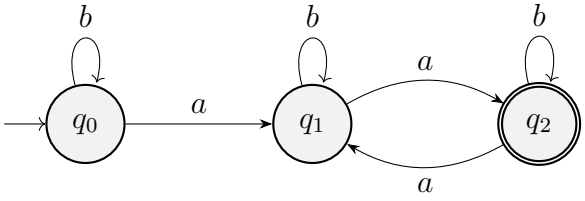


Figura 1.2: Autómata Finito Determinista del Ejercicio 1.2.2

1.2. Autómatas Finitos

Ejercicio 1.2.1. Considera el siguiente Autómata Finito Determinista (AFD) dado por $M = (Q, A, \delta, q_0, F)$, donde:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{0, 1\}$
- La función de transición viene dada por:

$$\begin{aligned} \delta(q_0, 0) &= q_1, & \delta(q_0, 1) &= q_0 \\ \delta(q_1, 0) &= q_2, & \delta(q_1, 1) &= q_0 \\ \delta(q_2, 0) &= q_2, & \delta(q_2, 1) &= q_2 \end{aligned}$$

- $F = \{q_2\}$

Describe informalmente el lenguaje aceptado.

Su representación gráfica está en la Figura 1.1.

Tenemos que el lenguaje aceptado por el autómata es el conjunto de todas las palabras que contienen la cadena 00 como subcadena. Es decir,

$$L = \{u_1 00 u_2 \in \{0,1\}^* \mid u_1, u_2 \in \{0,1\}^*\}.$$

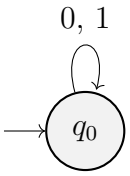
Ejercicio 1.2.2. Dado el AFD de la Figura 1.2, describir el lenguaje aceptado por dicho autómata.

El lenguaje aceptado por el autómata es el conjunto de todas las palabras que contienen un número par de a's. Es decir,

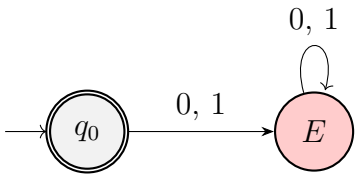
$$L = \{u \in \{a, b\}^* \mid n_a(u) \text{ es par}, \ n_a(u) > 0\},$$

Ejercicio 1.2.3. Dibujar AFDs que acepten los siguientes lenguajes con alfabeto $\{0, 1\}$:

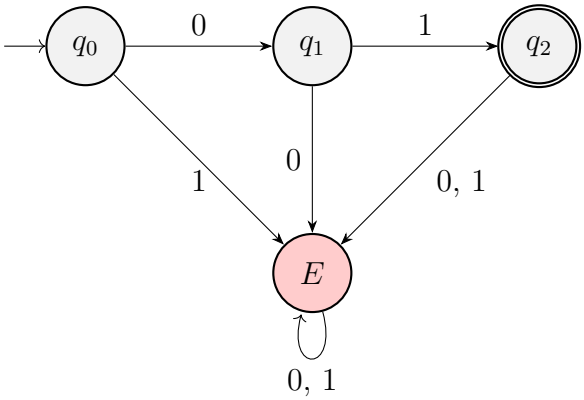
1. El lenguaje vacío,



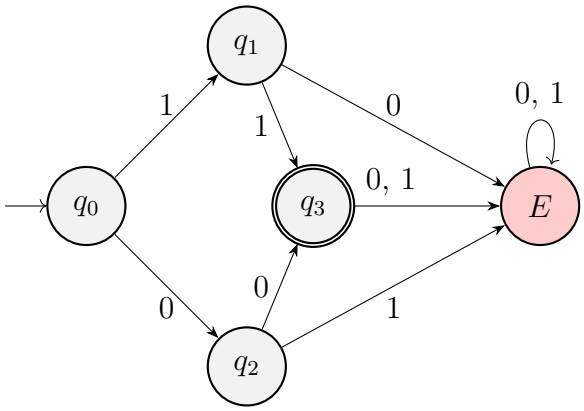
2. El lenguaje formado por la palabra vacía, es decir, $\{\varepsilon\}$,



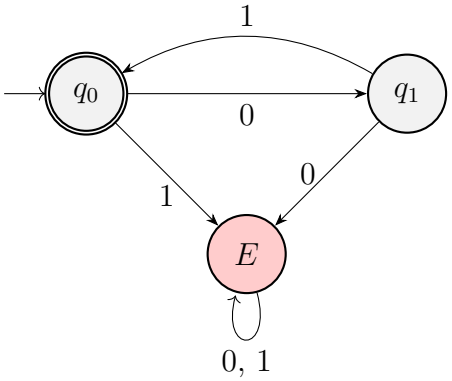
3. El lenguaje formado por la palabra 01, es decir, $\{01\}$,



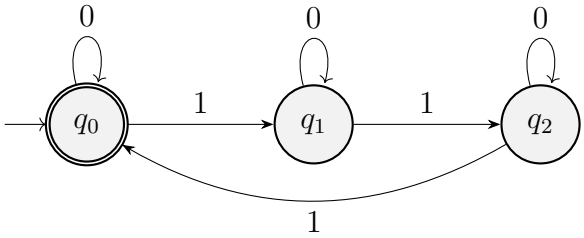
4. El lenguaje $\{11, 00\}$,



5. El lenguaje $\{(01)^i \mid i \geqslant 0\}$,



6. El lenguaje formado por las cadenas con 0's y 1's donde el número de unos es divisible por 3.



Ejercicio 1.2.4. Obtener a partir de la gramática regular $G = (\{S, B\}, \{1, 0\}, P, S)$, con

$$P = \begin{cases} S \to 110B \\ B \to 0B \mid 1B \mid \varepsilon \end{cases}$$

un AFND que reconozca el lenguaje generado por esa gramática.

El autómata obtenido es el de la Figura 1.3.

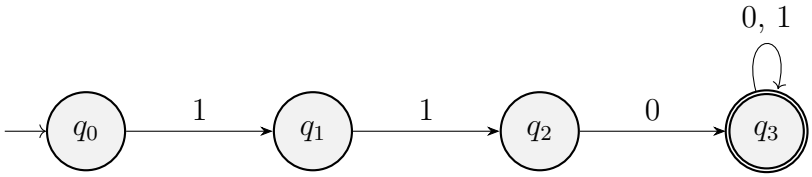


Figura 1.3: Autómata Finito No Determinista del Ejercicio 1.2.4

Ejercicio 1.2.5. Dada la gramática regular $G = (\{S\}, \{1, 0\}, P, S)$, con

$$P = \{S \to S10, S \to 0\},$$

obtener un AFD que reconozca el lenguaje generado por esa gramática.

El lenguaje es:

$$L = \{0(10)^n \mid n \in \mathbb{N} \cup \{0\}\}.$$

El autómata obtenido es el de la Figura 1.4.

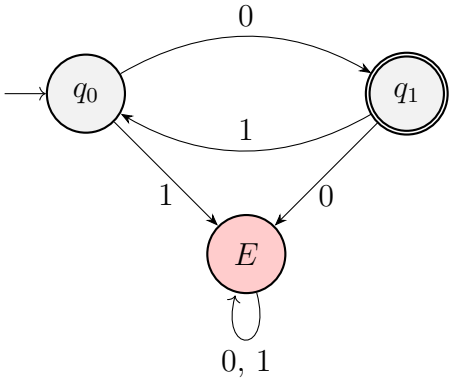


Figura 1.4: Autómata Finito Determinista del Ejercicio 1.2.5

Ejercicio 1.2.6. Construir un AFND o AFD (dependiendo del caso) que acepte las cadenas $u \in \{0,1\}^*$ que:

1. AFND. Contengan la subcadena 010.

   El autómata obtenido es el de la Figura 1.5.

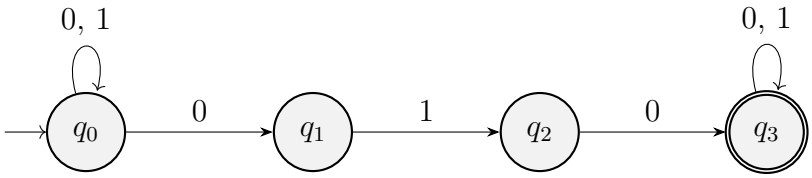


Figura 1.5: Autómata Finito No Determinista del Ejercicio 1.2.6 apartado 1.

2. AFND. Contengan la subcadena 110.

   El autómata obtenido es el de la Figura 1.6.

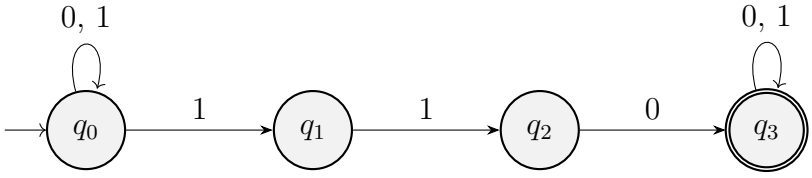


Figura 1.6: Autómata Finito No Determinista del Ejercicio 1.2.6 apartado 2.

3. AFD. Contengan simultáneamente las subcadenas 010 y 110.

   El estado q_0 representa que no se ha empezado ninguna de las subcadenas, y el estado q_F representa que se han encontrado ambas cadenas. Hay dos opciones:

   Opción 1 Primero se lee 010 y luego 110. Son los siguientes estados:

   - q_0: Estado inicial, no ha empezado la subcadena 010.
   - q_1: Se ha leído el 0 de la subcadena 010.
   - q_2: Se ha leído la subcadena 01 de la subcadena 010.
   - q_3: Se ha leído la subcadena 010. No ha empezado la subcadena 110.

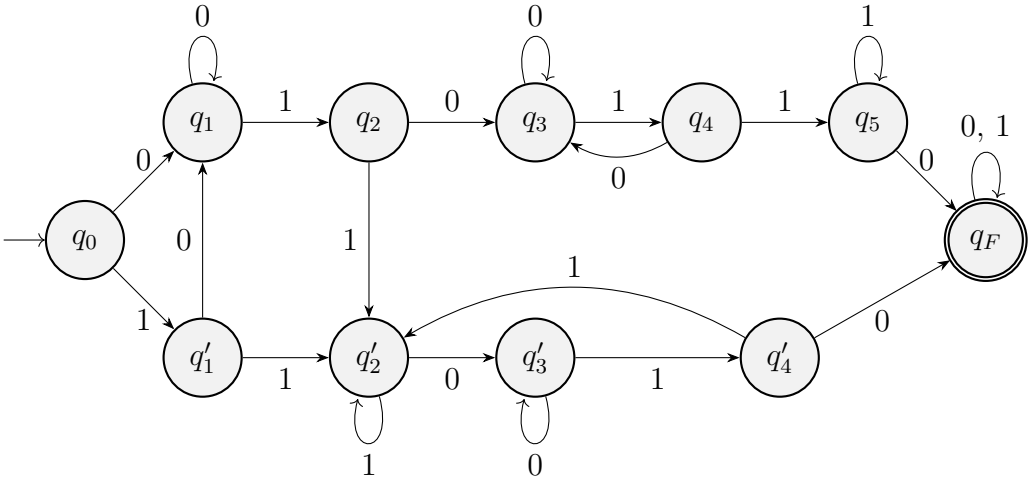


Figura 1.7: Autómata Finito Determinista del Ejercicio 1.2.6 apartado 3.

- q_4: Se ha leído el 1 de la subcadena 110.
- q_5: Se ha leído la subcadena 11 de la subcadena 110.
- q_F: Se ha leído la subcadena 110. Se han leído ambas subcadenas.

Opción 2 Primero se lee 110 y luego 010. Son los siguientes estados:

- q_0: Estado inicial, no ha empezado la subcadena 110.
- q_1': Se ha leído el 1 de la subcadena 110.
- q_2': Se ha leído la subcadena 11 de la subcadena 110.
- q_3': Se ha leído la subcadena 110. Se ha leído el 0 de la subcadena 010. Notemos que en este caso podemos agruparlo, puesto que el último carácter de la subcadena 110 es el mismo que el primero de la subcadena 010.
- q_4': Se ha leído la subcadena 01 de la subcadena 010.
- q_F: Se ha leído la subcadena 010. Se han leído ambas subcadenas.

El autómata obtenido es el de la Figura 1.7.

Ejercicio 1.2.7. Construir un AFD que acepte el lenguaje generado por la siguiente gramática:

$$S \to AB, \qquad A \to aA, \qquad A \to c, \qquad B \to bBb, \qquad B \to b.$$

El lenguaje generado por la gramática es:

$$L = \{a^n c b^{2m+1} \mid n, m \in \mathbb{N} \cup \{0\}\}.$$

El autómata obtenido es el de la Figura 1.8.

Ejercicio 1.2.8. Construir un AFD que acepte el lenguaje $L \subseteq \{a, b, c\}^*$ de todas las palabras con un número impar de ocurrencias de la subcadena abc.

El autómata tiene los siguientes estados:

- q_0: Llevo un número par de ocurrencias de abc, y no he empezado la siguiente.

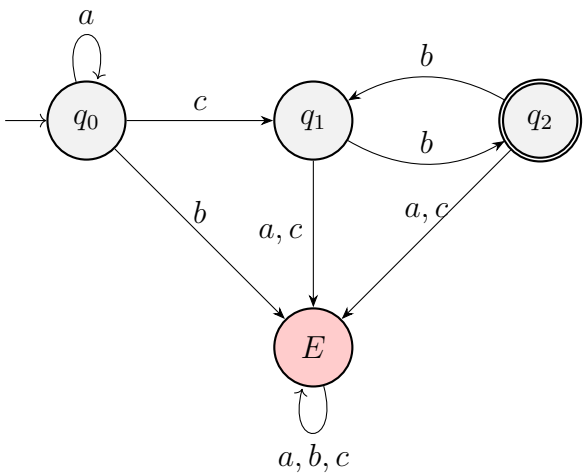


Figura 1.8: Autómata Finito Determinista del Ejercicio 1.2.7

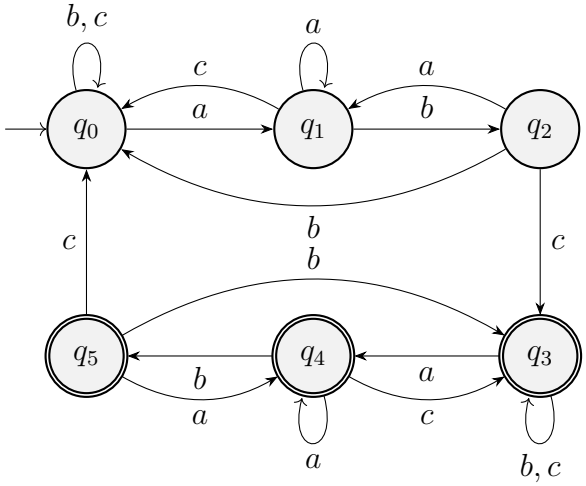


Figura 1.9: Autómata Finito Determinista del Ejercicio 1.2.8

- q_1: Acabo de empezar una ocurrencia impar de abc, llevo solo una a.
- q_2: Estoy en una ocurrencia impar de abc, llevo ab.
- q_3: Llevo un número impar de ocurrencias de abc, y no he empezado la siguiente.
- q_4: Acabo de empezar una ocurrencia par de abc, llevo solo una a.
- q_5: Estoy en una ocurrencia par de abc, llevo ab.

El autómata obtenido es el de la Figura 1.9.

Ejercicio 1.2.9. Sea L el lenguaje de todas las palabras sobre el alfabeto $\{0, 1\}$ que no contienen dos 1s que estén separados por un número impar de símbolos. Describir un AFD que acepte este lenguaje.

Sea $u \in L$. Veamos que, a lo sumo, puede tener dos 1's. Supongamos por reducción al absurdo que tiene tres 1's. Entonces, entre la primera y la segunda hay un

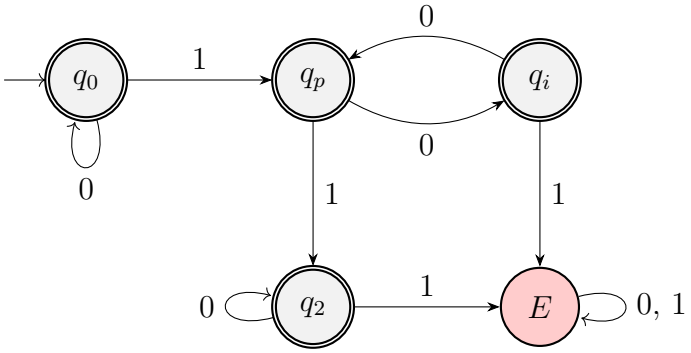


Figura 1.10: Autómata Finito Determinista del Ejercicio 1.2.9.

número impar de símbolos, y entre la segunda y la tercera hay un número impar de símbolos. Por lo tanto, entre el primer y el tercer 1 hay:

- Un número par de símbolos antes del segundo 1.
- El segundo 1.
- Un número par de símbolos entre el segundo y el tercer 1.

Por tanto, como el número de símbolos entre el primer y el tercer 1 es impar, entonces $u \notin L$. Por lo tanto, u tiene a lo sumo dos 1's.

Por tanto, los estados son:

- q_0: No se ha introducido ningún 1.
- q_p: Se ha introducido un 1, después de él y antes del siguiente 1 hay un número par de símbolos.
- q_i: Se ha introducido un 1, después de él y antes del siguiente 1 hay un número impar de símbolos.
- q_2: Se han introducido dos 1's, y no se ha introducido ningún otro.
- E: Estado de error.

El autómata obtenido es el de la Figura 1.10.

Ejercicio 1.2.10. Dada la expresión regular $(a+\varepsilon)b^*$, encontrar un AFND asociado y, a partir de este, calcular un AFD que acepte el lenguaje.

El AFND con transiciones nulas obtenido (siguiendo el algoritmo) es el de la Figura 1.11.

Podemos simplificar este autómata para que así la transición al AFD sea más sencilla. El autómata simplificado es el de la Figura 1.12.

A partir de este autómata simplificado, obtenemos el AFD de la Figura 1.13.

Ejercicio 1.2.11. Obtener una expresión regular para el lenguaje complementario al aceptado por la gramática

$$S \to abA \mid B \mid baB \mid \varepsilon, \qquad A \to bS \mid b, \qquad B \to aS.$$

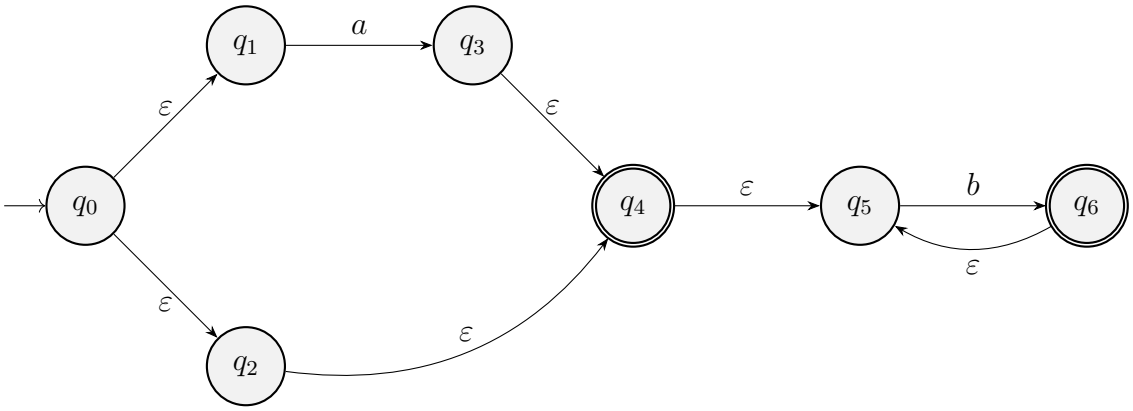


Figura 1.11: Autómata Finito No Determinista algorítmico del Ejercicio 1.2.10.

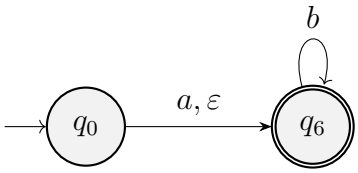


Figura 1.12: Autómata Finito No Determinista simplificado del Ejercicio 1.2.10.

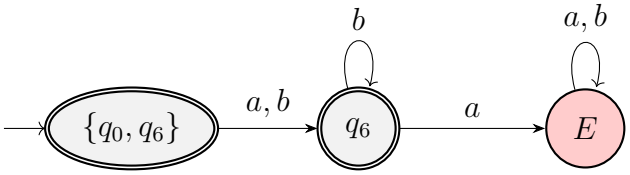


Figura 1.13: Autómata Finito Determinista del Ejercicio 1.2.10.

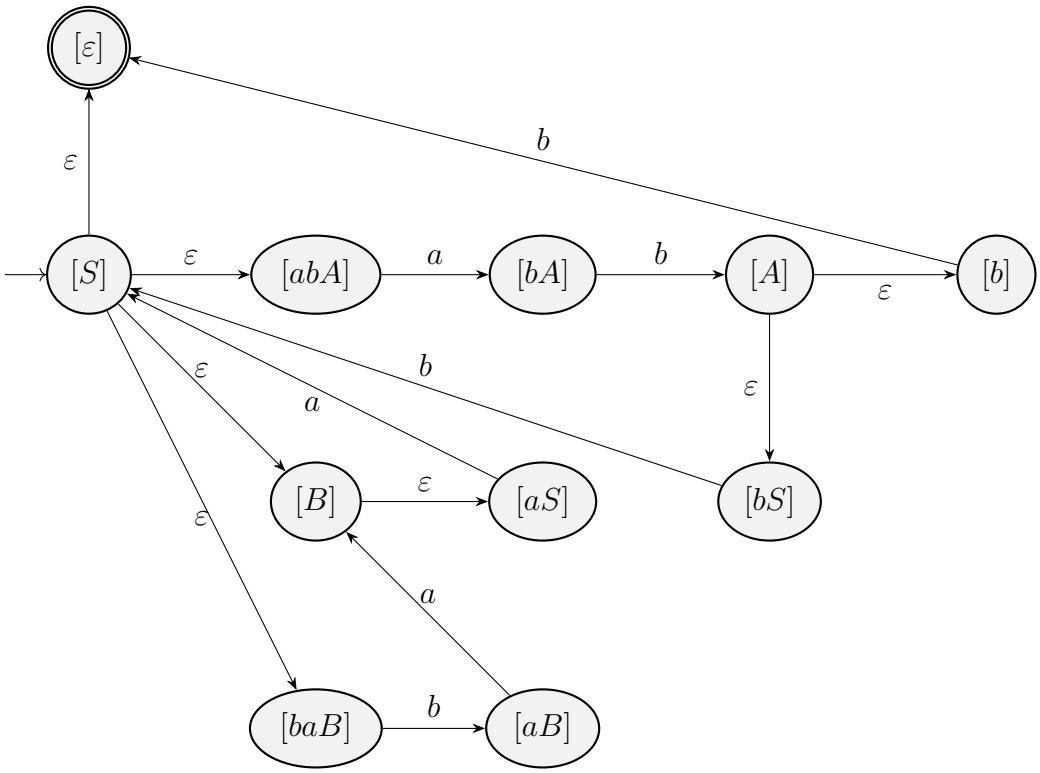


Figura 1.14: Autómata Finito Determinista del lenguaje $\mathcal{L}(S)$ del Ejercicio 1.2.11.

Observación. Construir un AFD asociado.

Esta gramática es lineal por la derecha. Algorítmicamente, obtenemos el autómata de la Figura 1.14 para el lenguaje generado por S, $\mathcal{L}(S)$.

Ahora, tendríamos que eliminar las transiciones nulas para poder así aplicar el algoritmo para hallar la expresión regular. Esto no es sencillo, por lo que vamos a intentar obtener de forma directa el AFD. Para ello, la gramática dada genera el mismo lenguaje que las siguientes reglas de producción, donde hemos eliminado la variable B:

$$S \to abA \mid aS \mid baaS \mid \varepsilon, \qquad A \to bS \mid b$$

Eliminamos ahora la variable A:

$$S \to abbS \mid abb \mid aS \mid baaS \mid \varepsilon$$

Veamos ahora que la regla $S \to abb$ no es relevante, ya que podemos obtenerla a partir de $S \to abbS$ y $S \to \varepsilon$. Por tanto, la gramática dada inicialmente genera el mismo lenguaje que si estas fuesen las reglas de producción:

$$S \to abbS \mid aS \mid baaS \mid \varepsilon$$

Por tanto, vemos que:

$$\mathcal{L}(G) = \{abb, a, baa\}^*.$$

En consecuencia, la expresión regular asociada a $\mathcal{L}(G)$ es:

$$(abb + a + baa)^*$$

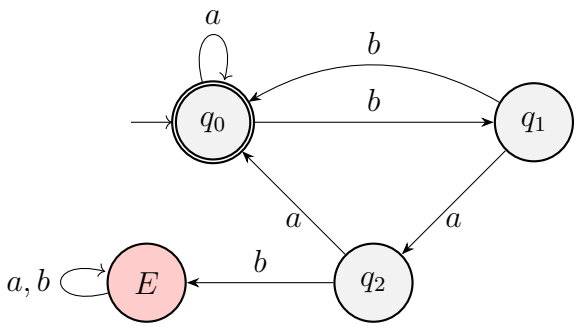


Figura 1.15: Autómata Finito Determinista del lenguaje $\mathcal{L}(G)$ del Ejercicio 1.2.11.

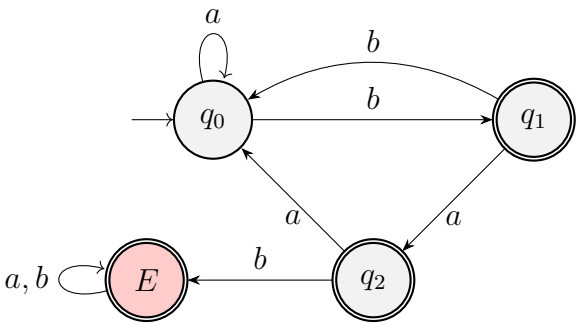


Figura 1.16: Autómata Finito Determinista del lenguaje $\overline{\mathcal{L}(G)}$ del Ejercicio 1.2.11.

El AFD asociado a esta expresión regular es el de la Figura 1.15.

Por tanto, el autómata finito determinista asociado al lenguaje complementario de $\mathcal{L}(G)$ es el de la Figura 1.16.

Buscamos ahora una expresión para $\overline{\mathcal{L}(G)}$. Resolvemos el siguiente sistema:

$$\begin{cases} q_0 & = aq_0 + bq_1 \\ q_1 & = bq_0 + aq_2 + \varepsilon \\ q_2 & = aq_0 + bq_E + \varepsilon \\ q_E & = aq_E + bq_E + \varepsilon \end{cases}$$

De la última ecuación, obtenemos que $q_E = (a+b)^*$. El sistema queda:

$$\begin{cases} q_0 & = aq_0 + bq_1 \\ q_1 & = bq_0 + aq_2 + \varepsilon \\ q_2 & = aq_0 + b(a+b)^* + \varepsilon \end{cases}$$

Sustituyendo q_2, obtenemos:

$$\begin{cases} q_0 & = aq_0 + bq_1 \\ q_1 & = bq_0 + a(aq_0 + b(a+b)^* + \varepsilon) + \varepsilon \end{cases}$$

Tenemos que:

$$q_1 = (b + aa)q_0 + ab(a+b)^* + a + \varepsilon$$

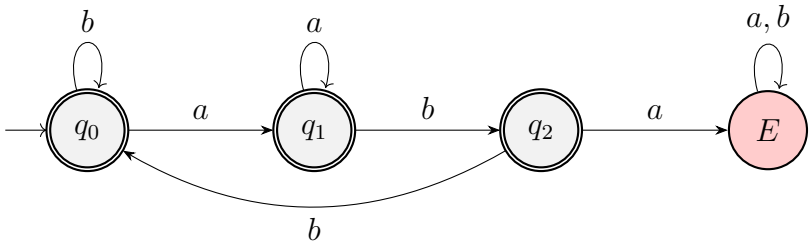


Figura 1.17: AFD del lenguaje del Ejercicio 1.2.12 apartado 3.

Sustituyendo, tenemos que:

$$\begin{aligned} q_0 &= aq_0 + b\left[(b+aa)q_0 + ab(a+b)^* + a + \varepsilon\right] = \\ &= (a + b(b+aa))q_0 + bab(a+b)^* + ba + b \overset{(*)}{=} \\ &\overset{(*)}{=} (a + b(b+aa))^*(bab(a+b)^* + ba + b) \end{aligned}$$

donde en $(*)$ hemos aplicado el Lema de Arden. Por tanto, la expresión regular asociada a $\overline{\mathcal{L}(G)}$ es:

$$(a + b(b+aa))^*(bab(a+b)^* + ba + b)$$

Ejercicio 1.2.12. Dar expresiones regulares para los lenguajes sobre el alfabeto $\{a, b\}$ dados por las siguientes condiciones:

1. Palabras que no contienen la subcadena a,

   $$b^*$$

2. Palabras que no contienen la subcadena ab.

   $$b^*a^*$$

3. Palabras que no contienen la subcadena aba.

   Este lenguaje viene descrito por el autómata de la Figura 1.17.

   Obtenemos la expresión regular asociada al lenguaje del autómata de la Figura 1.17.

   $$\begin{cases} q_0 &= bq_0 + aq_1 + \varepsilon \\ q_1 &= aq_1 + bq_2 + \varepsilon \\ q_2 &= bq_0 + aE + \varepsilon \\ E &= aE + bE \end{cases}$$

   Usando el Lema de Arden, obtenemos que $E = (a+b)^*$. Sustituyendo, obtenemos:

   $$\begin{cases} q_0 &= bq_0 + aq_1 + \varepsilon \\ q_1 &= aq_1 + bq_2 + \varepsilon \\ q_2 &= bq_0 + a(a+b)^* + \varepsilon \end{cases}$$

Sustituyendo q_2, obtenemos:

$$\begin{cases} q_0 &= bq_0 + aq_1 + \varepsilon \\ q_1 &= aq_1 + b(bq_0 + a(a+b)^* + \varepsilon) + \varepsilon \end{cases}$$

Usando el Lema de Arden, obtenemos que:

$$q_1 = a^*[b(bq_0 + a(a+b)^* + \varepsilon) + \varepsilon]$$

Sustituyendo en la primera ecuación, tenemos que:

$$\begin{aligned} q_0 &= bq_0 + aa^*[b(bq_0 + a(a+b)^* + \varepsilon) + \varepsilon] + \varepsilon = \\ &= bq_0 + aa^*[bbq_0 + ba(a+b)^* + b + \varepsilon] + \varepsilon = \\ &= (b + aa^*bb)q_0 + aa^*[ba(a+b)^* + b + \varepsilon] + \varepsilon \overset{(*)}{=} \\ &\overset{(*)}{=} (b + aa^*bb)^*[aa^*(ba(a+b)^* + b + \varepsilon) + \varepsilon] \end{aligned}$$

donde en $(*)$ hemos aplicado el Lema de Arden. Por tanto, la expresión regular asociada al lenguaje del autómata de la Figura 1.17 es:

$$(b + aa^*bb)^*[aa^*(ba(a+b)^* + b + \varepsilon) + \varepsilon]$$

Ejercicio 1.2.13. Determinar si el lenguaje generado por la siguiente gramática es regular:

$$S \to AabB, \qquad A \to aA \mid bA \mid \varepsilon, \qquad B \to Bab \mid Bb \mid ab \mid b.$$

En caso de que lo sea, encontrar una expresión regular asociada.

Es directo ver que el lenguaje generado por la gramática tiene como expresión regular asociada:

$$(a+b)^*ab(ab+b)^+.$$

Por tanto, el lenguaje es regular.

Ejercicio 1.2.14. Sobre el alfabeto $A = \{0, 1\}$ realizar las siguientes tareas:

1. Describir un autómata finito determinista que acepte todas las palabras que contengan a 011 o a 010 (o las dos) como subcadenas.

   Tenemos los siguientes estados:

   - $\underline{q_0}$: No se ha empezado ninguna subcadena.
   - $\underline{q_1}$: Se ha empezado una subcadena deseada. Tengo el carácter 0.
   - $\underline{q_2}$: Se continúa la subcadena deseada. Tengo los caracteres 01.
   - $\underline{q_3}$: Se ha encontrado la subcadena deseada. Tengo los caracteres 011 o 010.

   El autómata obtenido es el de la Figura 1.18.

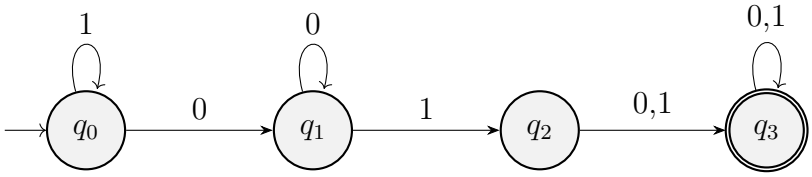


Figura 1.18: Autómata Finito Determinista del Ejercicio 1.2.14 apartado 1.

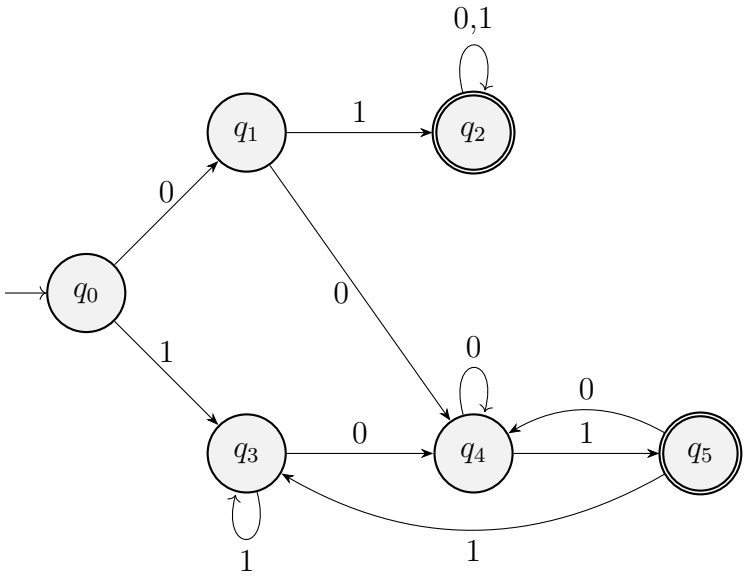


Figura 1.19: Autómata Finito Determinista del Ejercicio 1.2.14 apartado 2.

2. Describir un autómata finito determinista que acepte todas las palabras que empiecen o terminen (o ambas cosas) por 01.

   Tenemos los siguientes estados:

   - $\underline{q_0}$: No hemos leído nada.
   - $\underline{q_1}$: Hemos empezado con un 0, por lo que puede comenzar por 01 (o terminar por 01).
   - $\underline{q_2}$: Hemos empezado con 01, por lo que ya no hay más restricciones.
   - $\underline{q_3}$: No hemos empezado por 01, por lo que ha de terminar por 01.
   - $\underline{q_4}$: Ha de terminar por 01, y estamos en 0, por lo que si introduce un 1 puede terminar.
   - $\underline{q_5}$: Ha de terminar por 01, y acabamos de leer 01, por lo que podemos terminar.

   El autómata obtenido es el de la Figura 1.19.

3. Dar una expresión regular para el conjunto de las palabras en las que hay dos ceros separados por un número de símbolos que es múltiplo de 4 (los símbolos que separan los ceros pueden ser ceros y puede haber otros símbolos delante o detrás de estos dos ceros).

$$(0+1)^*0\,((0+1)(0+1)(0+1)(0+1))^*\,0(0+1)^*$$

Notemos que los dos 0's en cuestión están marcados en rojo para facilitar la comprensión.

4. Dar una expresión regular para las palabras en las que el número de ceros es divisible por 4.

   En un primer momento, podríamos pensar en:

   $$(1^*01^*01^*01^*01^*)^*$$

   No obstante, una palabra con 1's y sin 0's, que es aceptada por el lenguaje, no está contemplada en la expresión regular. La expresión regular correcta es:

   $$(1^*01^*01^*01^*0)^*1^*$$

Ejercicio 1.2.15. Construye una gramática regular que genere el siguiente lenguaje:

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

Tenemos los siguientes estados:

- $\underline{E_{01}}$: Tenemos un error en 0 y 1, 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 par. El número de 1's es impar.
- $\underline{E_1}$: Tenemos un error en 1, ya que el número de 1's es par. El número de 0's es impar.
- $\underline{X}$: No tenemos errores. El número de 0's y de 1's es impar.

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

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

Ejercicio 1.2.16. Encuentra una expresión regular que represente el siguiente lenguaje:

$$L_2 = \{0^n 1^m \mid n \geqslant 1, m \geqslant 0, n \text{ múltiplo de 3 y } m \text{ es par}\}.$$

La expresión regular es:

$$(000)^*(11)^*$$

Ejercicio 1.2.17. Diseña un autómata finito determinista que reconozca el siguiente lenguaje:

$L_3 = \{u \in \{0,1\}^* \mid \text{el número de 1's no es múltiplo de 3 y el número de 0's es par}\}.$

Sean n_0 el número de 0's y n_1 el número de 1's.
Tenemos la siguiente disposición de estados:

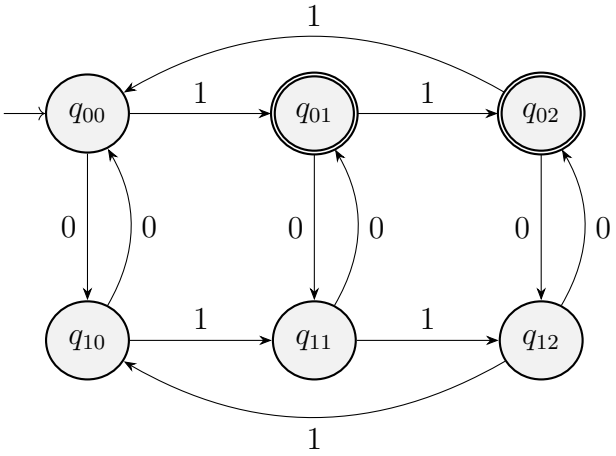


Figura 1.20: Autómata Finito Determinista del Ejercicio 1.2.17

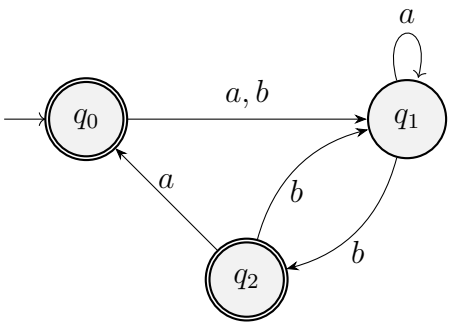


Figura 1.21: Autómata Finito Determinista del Ejercicio 1.2.18

- Los estados de arriba representan $n_0 \bmod 2 = 0$.
- Los estados de abajo representan $n_0 \bmod 2 = 1$.
- Los estados de la primera columna representan $n_1 \bmod 3 = 0$.
- Los estados de la segunda columna representan $n_1 \bmod 3 = 1$.
- Los estados de la tercera columna representan $n_1 \bmod 3 = 2$.

El estado q_{ij} representa $n_0 \bmod 2 = i$ y $n_1 \bmod 3 = j$.
El autómata obtenido es el de la Figura 1.20.

Ejercicio 1.2.18. Dar una expresión regular para el lenguaje aceptado por el autómata de la Figura 1.21.

Establecemos una ecuación por cada uno de los estados. El sistema inicial es:

$$\begin{cases} q_0 = \varepsilon + aq_1 + bq_1, \\ q_1 = aq_1 + bq_2, \\ q_2 = \varepsilon + aq_0 + bq_1. \end{cases}$$

Buscamos obtener la expresión regular asociada a q_1:

$$\begin{aligned} q_1 &= aq_1 + b + baq_0 + bbq_1 = \\ &= baq_0 + b + (a + bb)q_1 \overset{(*)}{=} \\ &\overset{(*)}{=} (a + bb)^*(baq_0 + b) \end{aligned}$$

donde en $(*)$ hemos aplicado el Lema de Arden. Sustituyendo en la ecuación de q_0 obtenemos:

$$\begin{aligned} q_0 &= \varepsilon + (a + b)q_1 = \\ &= \varepsilon + (a + b)(a + bb)^*(baq_0 + b) = \\ &= \varepsilon + (a + b)(a + bb)^*b + (a + b)(a + bb)^*baq_0 \overset{(*)}{=} \\ &\overset{(*)}{=} ((a + b)(a + bb)^*ba)^*(\varepsilon + (a + b)(a + bb)^*b) \end{aligned}$$

donde, de nuevo, en $(*)$ hemos aplicado el Lema de Arden. Por tanto, la expresión regular asociada al autómata es:

$$((a + b)(a + bb)^*ba)^*(\varepsilon + (a + b)(a + bb)^*b).$$

Ejercicio 1.2.19. Dado el lenguaje

$$L = \{u110 \mid u \in \{1, 0\}^*\},$$

encontrar la expresión regular, la gramática lineal por la derecha, la gramática lineal por la izquierda y el AFD asociado.

La expresión regular es:

$$(0 + 1)^*110.$$

La gramática lineal por la derecha es $G = (\{S, A\}, \{0, 1\}, P, S)$, donde P es:

$$\begin{aligned} S &\to 0S \mid 1S \mid A \\ A &\to 110. \end{aligned}$$

La gramática lineal por la izquierda es $G = (\{S, A\}, \{0, 1\}, P', S)$, donde P' es:

$$\begin{aligned} S &\to X110 \\ X &\to X0 \mid X1 \mid \varepsilon. \end{aligned}$$

El AFD asociado es el de la Figura 1.22. Sus estados son:

- $\underline{q_0}$: No estoy en la cadena 110 final.
- $\underline{q_1}$: He leído un 1 de la cadena final.
- $\underline{q_2}$: He leído un 11 de la cadena final.
- $\underline{q_3}$: He leído un 110 de la cadena final.

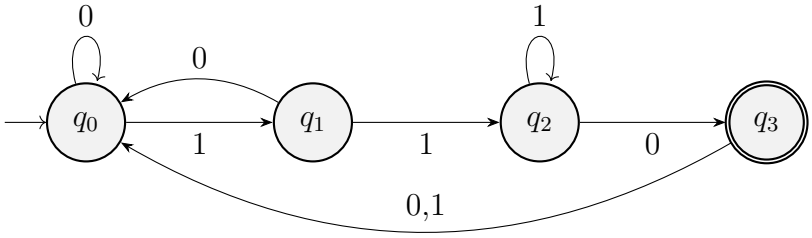


Figura 1.22: Autómata Finito Determinista del Ejercicio 1.2.19

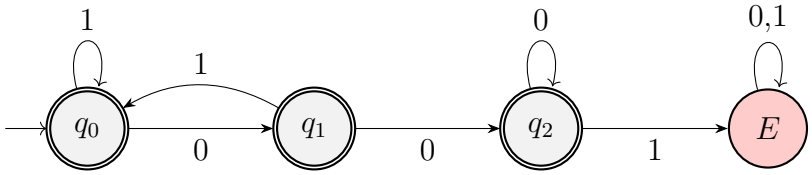


Figura 1.23: Autómata Finito Determinista del Ejercicio 1.2.21

Ejercicio 1.2.20. Dado un AFD, determinar el proceso que habría que seguir para construir una gramática lineal por la izquierda capaz de generar el Lenguaje aceptado por dicho autómata.

Sea $M = (Q, A, \delta, q_0, F)$ un AFD. Como Q es finito, podemos enumerar los estados como $Q = \{q_1, q_2, \ldots, q_n\}$. La Gramática Lineal por la Izquierda asociada es $G = (Q \cup \{S\}, A, P, S)$, donde hemos supuesto $S \notin Q$ debido a nuestra enumeración de los estados. Las reglas de producción son:

$$P = \begin{cases} S \to q_i & \forall q_i \in F \\ q_i \to q_j a & \forall q_i, q_j \in Q, a \in A \mid \delta(q_j, a) = q_i \\ q_0 \to \varepsilon. & \end{cases}$$

Notemos que lo que hacemos es invertir el autómata, obtener la gramática lineal por la derecha, y después invertir las reglas de producción de esta última.

Ejercicio 1.2.21. Construir un autómata finito determinista que acepte el lenguaje de todas las palabras sobre el alfabeto $\{0, 1\}$ que no contengan la subcadena 001. Construir una gramática regular por la izquierda a partir de dicho autómata.

Los estados son los siguientes:

- $\underline{q_0}$: No se ha empezado la subcadena 001
- $\underline{q_1}$: Se ha leído un 0 de la subcadena 001.
- $\underline{q_2}$: Se ha leído un 00 de la subcadena 001.
- $\underline{E}$: Se ha leído la subcadena 001, por lo que es el estado de error.

El autómata obtenido es el de la Figura 1.23.

Respecto a la gramática regular por la izquierda, usando el algoritmo descrito en el apartado anterior, tenemos que la gramática es $G = (Q \cup \{S\}, \{0,1\}, P, S)$, donde P es:

$$P = \begin{cases} S & \to & q_0 \mid q_1 \mid q_2, \\ q_0 & \to & q_0 1 \mid q_1 1 \mid \varepsilon \\ q_1 & \to & q_0 0, \\ q_2 & \to & q_1 0 \mid q_2 0, \\ E & \to & E0 \mid E1 \mid q_2 1, \end{cases}$$

Ejercicio 1.2.22. Sea $B_n = \{a^k \mid k \text{ es múltiplo de } n\}$. Demostrar que B_n es regular para todo n.

Fijado $n \in \mathbb{N}$, la expresión regular correspondiente es:

$$(a \overbrace{\cdots}^{n \text{ veces}} a)^* = (a^n)^*$$

Equivalentemente, usando la notación de las expresiones regulares de UNIX, la expresión regular sería:

$$(a\{n\})^*$$

Ejercicio 1.2.23. Sea A un alfabeto. Decimos que $u \in A^*$ es un prefijo de $v \in A^*$ si existe $w \in A^*$ tal que $uw = v$. Decimos que u es un prefijo propio de v si además $u \neq v$ y $u \neq \varepsilon$. Demostrar que si L es regular, también lo son los lenguajes siguientes:

1. NOPREFIJO$(L) = \{u \in L \mid$ ningún prefijo propio de u pertenece a $L\}$,

   Como L es regular, existe un AFD $M = (Q, A, \delta, q_0, F)$ tal que $L = \mathcal{L}(M)$. Construimos un AFD $M' = (Q \cup \{E\}, A, \delta', q_0, F)$, donde E es un estado de error ($E \notin Q$) y δ' es:

   $$\begin{cases} \delta'(q,a) = \delta(q,a) & \forall q \in Q \setminus F, a \in A \\ \delta'(q,a) = E & \forall q \in F, a \in A \\ \delta'(E,a) = E & \forall a \in A \end{cases}$$

   Demostramos mediante doble inclusión que NOPREFIJO$(L) = \mathcal{L}(M')$.

   $\subseteq$) Sea $u \in$ NOPREFIJO(L). Entonces, por definición de NOPREFIJO(L), $u \in L$. Por tanto, $\exists q \in F$ tal que $\delta^*(q_0, u) = q$. Para ver que $u \in \mathcal{L}(M')$, basta ver que $(\delta')^*(q_0, u) \in F$.

   Como u no tiene prefijos propios en L, entonces $\delta^*(q_0, u') \notin F$ para todo prefijo propio u' de u; es decir, en los pasos de cálculo desde q_0 hasta $\delta^*(q_0, u)$ no se pasa por ningún estado final. Por tanto, como en esos casos $\delta' = \delta$, entonces $\delta^*(q_0, u) = q \in F$, por lo que $u \in \mathcal{L}(M')$.

   $\supseteq$) Sea $u \in \mathcal{L}(M')$. En primer lugar, tenemos que $(\delta')^*(q_0, u) \in F$. Veamos ahora que los pasos de cálculo desde q_0 hasta $(\delta')^*(q_0, u)$ leyendo u no son ninguno finales.

Si alguno de ellos fuese final (si u tuviese algún prefijo propio $v \in L$), entonces desde él pasaríamos a E, y de este estado no final no saldríamos, llegando a contradicción. Por tanto, u no tiene prefijos propios pertenecientes a L. Además, como en estos casos $\delta' = \delta$, tenemos que $\delta^*(q_0, u) \in F$, luego $u \in L$. De esta forma, $u \in \text{NOPREFIJO}(L)$.

2. $\text{NOEXTENSION}(L) = \{u \in L \mid u \text{ no es un prefijo propio de ninguna palabra de } L\}$.

   Como L es regular, existe un AFD $M = (Q, A, \delta, q_0, F)$ tal que $L = \mathcal{L}(M)$. Construimos un AFD $M' = (Q, A, \delta, q_0, F')$, donde:

   $$F' = \{q \in F \mid \delta^*(q, u) \notin F, \forall u \in A^*\}$$

   Demostramos mediante doble inclusión que $\text{NOEXTENSION}(L) = \mathcal{L}(M')$.

   $\subseteq$) Sea $u \in \text{NOEXTENSION}(L)$. Entonces, por definición de $\text{NOEXTENSION}(L)$, $u \in L$. Por tanto, $\exists q \in F$ tal que $\delta^*(q_0, u) = q$. Para ver que $u \in \mathcal{L}(M')$, basta ver que $q \in F'$.

   Supongamos por reducción al absurdo $q \notin F'$. Entonces, $\exists v \in A^*$ tal que $\delta^*(q, v) \in F$. Pero entonces, $\delta^*(q_0, uv) = \delta^*(\delta^*(q_0, u), v) = \delta^*(q, v) \in F$, por lo que $uv \in L$ y, por tanto, u es prefijo propio de uv, lo cual es una contradicción. Por tanto, $q \in F'$ y, por tanto, $u \in \mathcal{L}(M')$.

   $\supseteq$) Sea $u \in \mathcal{L}(M')$. En primer lugar, tenemos que $\delta^*(q_0, u) = q \in F' \subset F$, luego $u \in L$. Veamos ahora que u no es prefijo propio de ninguna palabra de L.

   Supongamos por reducción al absurdo que u es prefijo propio de alguna palabra de L. Entonces, $\exists v \in A^* \setminus \{\varepsilon\}$ tal que $uv \in L$. Por tanto, $\delta^*(q_0, uv) \in F$. Pero entonces, $\delta^*(q_0, uv) = \delta^*(\delta^*(q_0, u), v) = \delta^*(q, v) \in F$. No obstante, hemos demostrado entonces que $q \notin F'$, lo cual es una contradicción. Por tanto, u no es prefijo propio de ninguna palabra de L y, por tanto, $u \in \text{NOEXTENSION}(L)$.

Ejercicio 1.2.24. Si $L \subseteq A^*$, define la relación $\equiv$ en A^* como sigue: si $u, v \in A^*$, entonces $u \equiv v$ si y solo si para toda $z \in A^*$, tenemos que $(uz \in L \Leftrightarrow vz \in L)$.

1. Demostrar que $\equiv$ es una relación de equivalencia.

   Veamos las tres propiedades de las relaciones de equivalencia:

   - Reflexiva: Sea $u \in A^*$. Entonces, para todo $z \in A^*$, tenemos trivialmente que $(uz \in L \Leftrightarrow uz \in L)$. Por tanto, $u \equiv u$.
   - Simétrica: Sean $u, v \in A^*$ tales que $u \equiv v$. Entonces, para todo $z \in A^*$, tenemos que $(uz \in L \Leftrightarrow vz \in L)$. Por tanto, para todo $z \in A^*$, tenemos que $(vz \in L \Leftrightarrow uz \in L)$, lo cual implica que $v \equiv u$.
   - Transitiva: Sean $u, v, w \in A^*$ tales que $u \equiv v$ y $v \equiv w$. Entonces, para todo $z \in A^*$, tenemos que $(uz \in L \Leftrightarrow vz \in L)$ y $(vz \in L \Leftrightarrow wz \in L)$. Por tanto, para todo $z \in A^*$, tenemos que $(uz \in L \Leftrightarrow wz \in L)$, lo cual implica que $u \equiv w$.

   Tenemos por tanto que $\equiv$ es una relación de equivalencia.

2. Calcular las clases de equivalencia de $L = \{a^i b^i \mid i \geqslant 0\}$.

3. Calcular las clases de equivalencia de $L = \{a^i b^j \mid i, j \geqslant 0\}$.

4. Demostrar que L es aceptado por un autómata finito determinista si y solo si el número de clases de equivalencia es finito.

   Demostramos mediante doble inclusión.

   $\Longrightarrow$) Supongamos que L es aceptado por un autómata finito determinista. Sea $M = (Q, A, \delta, q_0, F)$ su AFD minimal que acepta L. Supongamos $u, v \in A^*$ tales que $u \equiv v$. Sean:

   $$q_u := \delta^*(q_0, u),$$
   $$q_v := \delta^*(q_0, v).$$

   Veamos ahora que q_u, q_v son indistingibles, es decir, $q_u = q_v$.

   - Para todo $z \in A^*$, como $u \equiv v$, se tiene que:

     $$uz \in L \Leftrightarrow vz \in L.$$

     Equivalentemente, tenemos que:

     $$\delta^*(\delta^*(q_0, u), z) \in F \Longleftrightarrow \delta^*(\delta^*(q_0, v), z) \in F$$

     Es decir:

     $$\delta^*(q_u, z) \in F \Longleftrightarrow \delta^*(q_v, z) \in F$$

     Por tanto, q_u y q_v son indistinguibles; y como el autómata es minimal, $q_u = q_v$.

   Por tanto, hay una clase de equivalencia por cada estado de Q. Como Q es finito, el número de clases de equivalencia es finito.

   $\Longleftarrow$) Supongamos que el número de clases de equivalencia es finito. Sea el autómata $M = (Q, A, \delta, q_0, F)$, donde:

   - Q es el conjunto de clases de equivalencia de L,
   - $q_0 = [\varepsilon]$,
   - $\delta([u], a) = [ua]$. Veamos que está bien definida.
     Sea $u, v \in A^*$ tales que $u \equiv v$, y veamos que, para todo $a \in A$, $ua \equiv va$. Como $u \equiv v$, para todo $z \in A^*$, tenemos que:

     $$uz \in L \Leftrightarrow vz \in L.$$

     Por tanto, tomando $z = az'$, con $z' \in L$, tenemos que:

     $$uaz' \in L \Leftrightarrow vaz' \in L.$$

     Es decir, $ua \equiv va$. Por tanto, δ está bien definida.
   - $F = \{[u] \in Q \mid u \in L\}$.
     Para ver que F está bien definida, veamos que, si $u \equiv v$, entonces $u \in L \Longleftrightarrow v \in L$. Esto es directo tomando $z = \varepsilon$.

Veamos ahora que $L = \mathcal{L}(M)$.

$u \in \mathcal{L}(M) \iff \delta^*(q_0, u) \in F \iff \delta^*([\varepsilon], u) \in F \iff \delta^*([u], \varepsilon) = [v], \text{con } v \in L$

Como $\delta^*([u], \varepsilon) = [u]$, tenemos que:

$$u \in \mathcal{L}(M) \iff u \in L$$

5. ¿Qué relación existe entre el número de clases de equivalencia y el autómata finito minimal que acepta L?

Ejercicio 1.2.25. Dada una palabra $u = a_1 \cdots a_n \in A^*$, se llama $\text{Per}(u)$ al conjunto

$$\{a_{\sigma(1)}, \ldots, a_{\sigma(n)} \mid \sigma \text{ es una permutación de } \{1, \ldots, n\}\}.$$

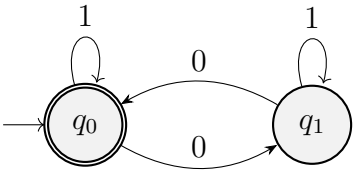
Dado un lenguaje L, se llama $\text{Per}(L) = \bigcup_{u \in L} \text{Per}(u)$. Dar expresiones regulares y autómatas minimales para $\text{Per}(L)$ en los siguientes casos:

1. $L = (00 + 1)^*$,

   Tenemos que:

   $$\text{Per}(L) = \{u \in A^* \mid \ n_0(u) \text{ es par}\}$$

   Su autómata finito minimal es:

2. $L = (0 + 1)^*0$,
3. $L = (01)^*$.

¿Es posible que, siendo L regular, $Per(L)$ no lo sea?

1.3. Propiedades de los Lenguajes Regulares

Ejercicio 1.3.1. Determinar si los siguientes lenguajes son regulares o libres de contexto. Justificar las respuestas.

1. $\{0^i b^j \mid i = 2j \text{ ó } 2i = j\}$
2. $\{uu^{-1} \mid u \in \{0,1\}^*, |u| \leqslant 1000\}$
3. $\{uu^{-1} \mid u \in \{0,1\}^*, |u| \geqslant 1000\}$
4. $\{0^i 1^j 2^k \mid i = j \text{ ó } j = k\}$

Ejercicio 1.3.2. Determinar qué lenguajes son regulares o libres de contexto de los siguientes:

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

b) Números en binerio que sean múltiplos de 4

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

Ejercicio 1.3.3. Determinar qué lenguajes son regulares y qué lenguajes son libres de contexto entre los siguientes:

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

b) Conjunto $\{0^i 1^2 j 0^{i+j} \mid i, j \geqslant 0\}$

c) Conjunto $\{0^i 1^j 0^{i*j} \mid i, j \geqslant 0\}$

Ejercicio 1.3.4. Determina si los siguientes lenguajes son regulares. Encuentra una gramática que los genere o un reconocedor que los acepte.

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

b) $L_2 = \{001^i 0^j \mid i, j \geqslant 1\}$.

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

Ejercicio 1.3.5. Sea el alfabeto $A = \{0, 1, +, =\}$, demostrar que el lenguaje

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

no es regular.

Ejercicio 1.3.6. Determinar si los siguientes lenguajes son regulares o no:

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

b) 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.

c) L es el lenguaje sobre el alfebeto $\{0,1\}$ formado por las palabres en las que el tercer símbolo empezando por el final es un 1.

Ejercicio 1.3.7. 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.

Ejercicio 1.3.8. Encontrar un AFD minimal para el lenguaje

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

Ejercicio 1.3.9. 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^+
2. $a(a+b)^*b$

Ejercicio 1.3.10. Determinar autómatas minimales para los lenguajes $L(M_1) \cup L(M_2)$ y $L(M_1) \cap \overline{L(M_2)}$ donde,

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

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

Ejercicio 1.3.11. Dado el conjunto regular representado por la expresión regular $a^*b^* + b^*a^*$, construir un autómata finido determinístico minimal que lo acepte.

Ejercicio 1.3.12. 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.

Ejercicio 1.3.13. Dados los alfabetos $A = \{0, 1, 2, 3\}$ y $B = \{0, 1\}$ y el homomorfismo f de A^* en B^* dado por:

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

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 $f^{-1}(L)$.

Ejercicio 1.3.14. Determinar si las expresiones regulares siguientes representan el mismo lenguaje:

a) $(b + (c + a)a^*(b + c))^*(c + a)a^*$

b) $b^*(c + a)((b + c)b^*(c + a))^*a^*$

c) $b^*(c + a)(a^*(b + c)b^*(c + a))^*a^*$

Justificar la respuesta.

Ejercicio 1.3.15. Construir un autómata finito determinista minila que acepte el conjunto de palabras sobre el alfabeto $A = \{0, 1\}$ que representen números no divisibles por dos ni por tres.

Ejercicio 1.3.16.

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

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

2. Construye un autómata que reconozca el siguiente lenguaje:

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

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

Ejercicio 1.3.17. Construir autómatas finitos para los siguientes lenguajes sobre el alfabeto $\{a, b, c\}$:

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

b) L_2: palabras en las que nunca hay una 'a' posterior a una 'c'.

c) $(L_1 \setminus L_2) \cup (L_2 \setminus L_1)$

¿Qué podemos concluir sobre L_1 y L_2?

Ejercicio 1.3.18. Si $f : \{0, 1\}^* \to \{a, b, c\}^*$ es un homomorfismo dao por

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

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.

Ejercicio 1.3.19. Si L_1 es el lenguaje asociadoa la expresión regular $01(01 + 1)^*$ y L_2 el lenguaje asociado a la expresión $(1 + 10)^*01$, encontrar un autómata minimal que acepte el lenguaje $L_1 \setminus L_2$.

Ejercicio 1.3.20. 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 para el lenguaje $f^{-1}(L)$ donde f es el homomorfismos entre A_1^* y A_2^* dado por

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

Ejercicio 1.3.21. Dado el lenguaje L asociado a la expresión regular $(01 + 011)^*$ y el homomorfismo $f : \{0, 1\}^* \to \{0, 1\}^*$ dado por $f(0) = 01$, $f(1) = 1$, construir una expresión regular para el lenguaje $f^{-1}(L)$.

Ejercicio 1.3.22. Dar expresiones regulares para los siguientes lenguajes sobre el alfabeto $A_1 = \{0, 1, 2\}$:

a) 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.

b) 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 $f(L)$.

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

Ejercicio 1.3.23. Dados los lenguajes

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

y

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

encuentra:

a) Una gramática regular que genere el lenguaje L_1.

b) Una expresión regular que represente al lenguaje L_2.

c) Un automata finito determinista que acepte las cadenas de la concatenación de los lenguajes L_1L_2. Aplica el algoritmos para minimizar este autómata.