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:

$$\begin{aligned} V &= \{S, X, Y\} \\ T &= \{a, b\} \\ S &= S \end{aligned}$$

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

$$\begin{aligned} S &\to XYX \\ X &\to aX \mid bX \mid \varepsilon \\ Y &\to bbb \end{aligned}$$

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:

$$\begin{aligned} S &\to aX \\ X &\to aX \mid bX \mid \varepsilon \end{aligned}$$

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

$\subset$) Sea $z \in L^* = \bigcup\limits_{i \in \mathbb{N}} L^i$. 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 $z = b^{i_1} a b^{j_1} a b^{k_1} \cdot \ldots \cdot b^{i_n} a b^{j_n} a b^{k_n}$. 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 $u \in \{abb, baab\}^* = \bigcup\limits_{i \in \mathbb{N}} \{abb, baab\}^i$. 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:

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

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:

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

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:

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

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:

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

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:

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

   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:

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

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:

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

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:

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

   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:

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

   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).

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:

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

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:

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

   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.

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.

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

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.

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

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

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

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:

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

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

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 $L^* = L$.

$$L = L^* \Longleftrightarrow \varepsilon \in L \wedge a, b \in L \Longrightarrow ab \in L$$

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 $v = \varepsilon \Longrightarrow v \in L$ por hipótesis.
2. Si $v \neq \varepsilon$, $\exists n \in \mathbb{N}$ tal que
$$v = a_1 a_2 \ldots a_n$$
con $a_i \in L$ $\forall i \in \{1, \ldots, n\}$, 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 $a, b \in L = L^* \Longrightarrow ab \in L^* = L$.

□

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

Demostración.

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

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

□

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.

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:

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

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

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

   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:

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

   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:

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

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:

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

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.

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:

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

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:

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

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:

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

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

9. Palíndromos formados con las letras a y b.

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

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

   Notemos que las reglas $S \rightarrow a \mid b$ 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. $\{uv \in \{0, 1\}^* \mid u^{-1} \text{ es un prefijo de } v\}$

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

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

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

3. $\{u1^n \in \{0,1\}^* \mid |u| = n\}$
   Sea la gramática $G = (V, T, P, S)$ dada por:
$$V = \{S, X\}$$
$$T = \{0, 1\}$$
$$S = S$$
$$P = \left\{ \begin{array}{lll} S & \to & XS1 \mid \varepsilon \\ X & \to & 0 \mid 1 \end{array} \right\}$$

4. $\{a^n b^{n+1} \in \{a,b\}^* \mid n \geqslant 0\}$ (observar transparencias de teoría)
   Sea la gramática $G = (V, T, P, S)$ dada por:
$$V = \{S\}$$
$$T = \{a, b\}$$
$$S = S$$
$$P = \left\{ \begin{array}{lll} S & \to & aSb \mid b \end{array} \right\}$$

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:
$$V = \{S, X\}$$
$$T = \{a, b, c\}$$
$$S = S$$
$$P = \left\{ \begin{array}{lll} S & \to & aSc \mid X \\ X & \to & bXc \mid \varepsilon \end{array} \right\}$$

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. $\{ww \mid w \in \{0,1\}^*\}$
2. $\{a^{n^2} \in \{a\}^* \mid n \geqslant 0\}$
3. $\{a^p \in \{a\}^* \mid p \text{ es primo}\}$
4. $\{a^n b^m \in \{a,b\}^* \mid n \leqslant m^2\}$

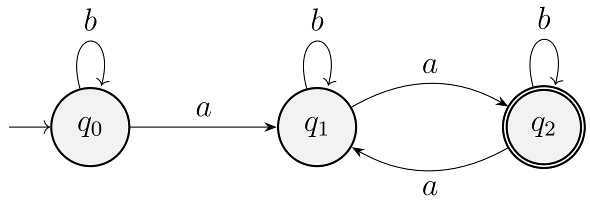

Figura 1.1: Autómata Finito Determinista del Ejercicio 1.2.2

1.2. Autómatas Finitos

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

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

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

- $F = \{q_2\}$

Describe informalmente el lenguaje aceptado.

Ejercicio 1.2.2. Dado el AFD de la Figura 1.1, describir el lenguaje aceptado por dicho autómata.

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

$$P = \{S \to 110B, B \to 1B, B \to 0B, B \to \varepsilon\},$$

un AFND que reconozca el lenguaje generado por esa gramática.

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.

Ejercicio 1.2.6. Construir un Autómata Finito No Determinista (AFND) que acepte las cadenas $u \in \{0, 1\}^*$ que contengan la subcadena 010. Construir un Autómata Finito No Determinista que acepte las cadenas $u \in \{0, 1\}^*$ que contengan la subcadena 110. Obtener un AFD que acepte las cadenas $u \in \{0, 1\}^*$ que contengan simultáneamente las subcadenas 010 y 110.

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

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

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.

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.

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.

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

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

Observación. Construir un AFD asociado.

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,
2. Palabras que no contienen la subcadena ab.
3. Palabras que no contienen la subcadena aba.

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.

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.

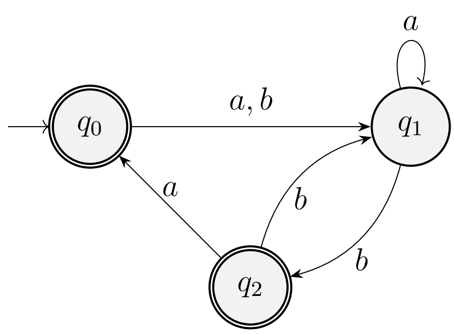

Figura 1.2: Autómata Finito Determinista del Ejercicio 1.2.18

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

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).

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

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

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

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

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

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

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

Ejercicio 1.2.18. Dar una expresión regular para el lenguaje aceptado por el autómata de la Figura 1.2.

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.

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.

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.

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

Ejercicio 1.2.23. Decimos que u es un prefijo de v si existe w 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

1. $NOPREFIJO(L) = \{u \in L \mid$ ningún prefijo propio de u pertenece a $L\}$,
2. $NOEXTENSION(L) = \{u \in L \mid u$ no es un prefijo propio de ninguna palabra de $L\}$.

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.
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 determinístico si y solo si el número de clases de equivalencia es finito.
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 $Per(u)$ 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 $Per(L) = \bigcup_{u \in L} Per(u)$. Dar expresiones regulares y autómatas minimales para $Per(L)$ en los siguientes casos:

1. $L = (00 + 1)^*$,
2. $L = (0 + 1)^*0$,
3. $L = (01)^*$.

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