## 4.13. Propiedades de clausura de los LIC

En la sección 3.2 se vio que los lenguajes regulares son cerrados bajo la concatenación, la estrella de Kleene y todas las operaciones booleanas. Los LIC poseen propiedades de clausura mucho más restringidas: son cerrados para las operaciones de unión, concatenación y estrella de Kleene Teorema 4.13.1 pero, en general, no son cerrados para intersección, complementos ni diferencias Teorema 4.13.2.

**4.13.1 Teorema.** *La colección de los lenguajes independientes del contexto es cerrada para las operaciones de unión, concatenación y estrella de Kleene. Es decir, dadas GIC* $G_1 = (V_1, \Sigma, S_1, P_1)$ *y* $G_2 = (V_2, \Sigma, S_2, P_2)$ *tales que* $L(G_1) = L_1$ *y* $L(G_2) = L_2$*, se pueden construir GIC que generen los lenguajes* $L_1 \cup L_2$*,* $L_1 L_2$ *y* $L_1^*$*, respectivamente.*

*Demostración*: Sin pérdida de generalidad, podemos suponer que $G_1$ y $G_2$ no tienen variables en común (en caso contrario, simplemente cambiamos los nombres de las variables). Para construir una GIC $G$ que genere $L_1 \cup L_2$ introducimos una variable *nueva* $S$, la variable inicial de $G$, junto con las producciones $S \to S_1$ y $S \to S_2$. Las producciones de $G_1$ y $G_2$ se mantienen. Concretamente,

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\})$$

Esquemáticamente, $G$ tiene el siguiente aspecto:

$$\begin{array}{l} S \to S_1 \mid S_2 \\ \left.\begin{array}{lll} S_1 & \to & \cdots \\ \vdots & & \vdots \end{array}\right\} \text{ producciones de } G_1 \\ \left.\begin{array}{lll} S_2 & \to & \cdots \\ \vdots & & \vdots \end{array}\right\} \text{ producciones de } G_2 \end{array}$$

Claramente, $L(G) = L_1 \cup L_2$.

Una GIC $G$ que genere $L_1 L_2$ se construye similarmente, añadiendo la producción $S \to S_1 S_2$. Es decir,

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, S, P_1 \cup P_2 \cup \{S \to S_1 S_2\})$$

Esquemáticamente, $G$ es la gramática:

$$\begin{array}{l} S \to S_1 S_2 \\ \left.\begin{array}{lcl} S_1 & \to & \cdots \\ \vdots & & \vdots \end{array}\right\} \text{producciones de } G_1 \\ \left.\begin{array}{lcl} S_2 & \to & \cdots \\ \vdots & & \vdots \end{array}\right\} \text{producciones de } G_2 \end{array}$$

Claramente, $L(G) = L_1 L_2$.

Para generar $L_1^*$ basta definir $G$ como

$$G = (V_1, \Sigma, S_1, P_1 \cup \{S_1 \to S_1 S_1,\ S_1 \to \lambda\})$$

Esquemáticamente, $G$ es la gramática:

$$\left.\begin{array}{lcl} S_1 & \to & S_1 S_1 \mid \lambda \mid \cdots \\ \vdots & & \vdots \end{array}\right\}$$

donde los puntos suspensivos representan las producciones originales de $G_1$. De esta forma, $L(G) = L_1^*$. □

***Ejemplo*** Utilizar las construcciones del Teorema 4.13.1 para encontrar una GIC que genere el lenguaje $L_1 L_2^*$ donde $L_1 = (ab \cup ba)^*$ y $L_2 = \{a^i b^i : i \geq 0\}$.

Solución: El lenguaje $L_1$ se puede generar con la gramática

$$S_1 \to S_1 S_1 \mid ab \mid ba \mid \lambda$$

y el lenguaje $L_2^*$ se puede generar con

$$S_2 \to S_2 S_2 \mid a S_2 b \mid \lambda.$$

Finalmente, el lenguaje $L_1 L_2^*$ se puede generar con

$$\begin{cases} S_3 \to S_1 S_2 \\ S_1 \to S_1 S_1 \mid ab \mid ba \mid \lambda \\ S_2 \to S_2 S_2 \mid a S_2 b \mid \lambda. \end{cases}$$

**4.13.2 Teorema.** *La colección de los lenguajes independientes del contexto no es cerrada (en general) para las siguientes operaciones:*

*(1) Intersección.*

*(2) Complemento.*

*(3) Diferencia.*

*Demostración*:

(1) La intersección de dos LIC puede ser un lenguaje que no es LIC. Considérense, como ejemplo, los lenguajes

$$L_1 = \{a^i b^i c^j : i, j \geq 0\},$$
$$L_2 = \{a^i b^j c^j : i, j \geq 0\}.$$

Tanto $L_1$ como $L_2$ son LIC porque son generados por las gramáticas $G_1$ y $G_2$, respectivamente:

$$G_1: \begin{cases} S \to AB \\ A \to aAb \mid \lambda \\ B \to cB \mid \lambda \end{cases} \qquad G_2: \begin{cases} S \to AB \\ A \to aA \mid \lambda \\ B \to bBc \mid \lambda \end{cases}$$

Pero $L_1 \cap L_2 = \{a^i b^i c^i : i \geq 0\}$ no es un LIC, según se mostró, usando el lema de bombeo, en la sección 4.12.

(2) Razonamos por contradicción: si el complemento de todo LIC fuera un LIC se podría concluir que la intersección de dos LIC $L_1$ y $L_2$ sería un LIC ya que $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$. Esto estaría en contradicción con la parte (1) del presente teorema.

(3) Razonamos por contradicción: si la diferencia de dos LIC cualesquiera fuera un LIC se podría concluir que el complemento de un LIC $L$ sería también un LIC ya que $\overline{L} = \Sigma^* - L$. Esto estaría en contradicción con la parte (2) del presente teorema. □

El siguiente teorema afirma que los LIC también son cerrados bajo homomorfismos.

**4.13.3 Teorema.** *Sea $h : \Sigma^* \to \Gamma^*$ un homomorfismo. Si $L$ es un LIC sobre $\Sigma$, entonces $h(L)$ es un LIC sobre $\Gamma$.*

*Demostración*: La demostración consiste en transformar una gramática $G$ que genere el lenguaje $L$ en una gramática $G'$ que genere $h(L)$. Para ello basta mantener las mismas variables de $G$ y definir las producciones de $G'$, a partir de las de $G$, cambiando cada

terminal $a$ por $h(a)$. Es fácil ver que una derivación $S \overset{+}{\Longrightarrow} w$ en $G$, con $w \in \Sigma^*$, se puede transformar en una derivación $S \overset{+}{\Longrightarrow} h(w)$ en $G'$; esto muestra $h(L) \subseteq L(G')$.

Para establecer la otra contenencia, es decir, $L(G') \subseteq h(L)$, hay que demostrar que si $S \overset{+}{\Longrightarrow}_{G'} z$, con $z \in \Gamma^*$, entonces $z$ es de la forma $z = h(w)$ para algún $w \in L$. Esto puede hacerse considerando el árbol de la derivación $S \overset{+}{\Longrightarrow}_{G'} z$. Dicho árbol se puede transformar en un árbol de una derivación en $G$, modificando adecuadamente las hojas: las hojas del nuevo árbol forman una cadena $w \in \Sigma^*$ y las hojas del árbol inicial forman la cadena $h(w)$. □

***Ejemplo*** Utilizar homomorfismos para concluir que el lenguaje $L = \{0^i 1^i 2^i 3^i : i \geq 0\}$, sobre el alfabeto $\{0, 1, 2, 3\}$ no es LIC.

Solución: La idea es "convertir" $L$ en el lenguaje $\{a^i b^i c^i : i \geq 0\}$, que no es LIC, según se mostró en el primer ejemplo de la sección 4.12. Razonamos de la siguiente manera: si $L$ fuera un LIC, lo sería también $h(L)$, donde $h$ es el homomorfismo $h : \{0, 1, 2, 3\}^* \to \{a, b, c\}^*$ definido por $h(0) = a$, $h(1) = b$, $h(2) = c$ y $h(3) = \lambda$. Pero

$$h(L) = \{h(0)^i h(1)^i h(2)^i h(3)^i : i \geq 0\} = \{a^i b^i c^i : i \geq 0\}.$$

Por consiguiente, $L$ no es un LIC.

## Ejercicios de la sección 4.13

1. Utilizar las construcciones del Teorema 4.13.1 para encontrar GIC que generen los siguientes lenguajes:

   (i) $a^+(a \cup bab)^*(b^* \cup a^*b)$.

   (ii) $(L_1 \cup L_2)L_3^*$, donde $L_1 = ab^*a$, $L_2 = a \cup b^+$ y $L_3 = aa \cup bb \cup aba$.

   (iii) $L_1 \cup L_2 L_3^*$, donde $L_1 = ab^*a$, $L_2 = b^+$ y $L_3 = \{a^i b a^i : i \geq 0\}$.

2. (i) Mostrar que los dos lenguajes siguientes, sobre $\Sigma = \{a, b, c, d\}$, son LIC:

   $$L_1 = \{a^i b^i c^j d^j : i, j \geq 1\},$$
   $$L_2 = \{a^i b^j c^j d^k : i, j, k \geq 1\}.$$

   (ii) Demostrar que $L_1 \cap L_2$ no es un LIC.

3. Utilizar homomorfismos para concluir que los siguientes lenguajes sobre el alfabeto $\{0, 1\}$ no son LIC:

(i) $L = \{u : |u| \text{ es un número primo}\}$.

(ii) $L = \{u : |u| \text{ es un cuadrado perfecto}\}$.

4. Demostrar que los LIC son cerrados para la operación de reflexión. Concretamente, demostrar que si $L$ es un LIC, también lo es el lenguaje $L^R = \{w^R : w \in L\}$.