Entrega 3

José Antonio Álvarez

22 de noviembre de 2017

Ejercicio 14. Dar gramáticas independientes del contexto que generen los siguientes lenguajes sobre el alfabeto $A = \{0, 1\}$:

1. L_1: conjunto de palabras tal que si la palabra empieza por 0 , entonces tiene el mismo número de 0s que de 1s:

$$S \to 0A_0|1D$$
$$D \to BD|\varepsilon$$
$$B \to 0|1$$
$$A_0 \to 0A_x|1A_0A_0$$
$$A_1 \to 1A_x|0A_1A_1$$
$$A_x \to 1A_0|0A_1|\varepsilon$$

2. L_2: conjunto de palabras tal que si la palabra termina por 1 , entonces tiene un número de 1s mayor o igual que el número de 0s:

$$S \to D0|C1$$
$$D \to BD|\varepsilon$$
$$B \to 0|1$$
$$C \to \varepsilon|0|C1C$$

3. $L_1 \cap L_2$:

Analicemos el problema antes de proceder a la gramática. Sean n_0 y n_1 el número de ceros y de unos de la palabra u respectivamente. En la siguiente tabla podemos apreciar las restricciones que tendrán que cumplir n_0 y n_1 dependiendo de si u empieza/acaba por 0/1 ($v, w \in \{0, 1\}^*$):

Tabla 1: Restricciones para u

	$u = w0$	$u = w1$
$u = 0v$	$n_0 = n_1$ (1)	$n_0 = n_1$ (2)
$u = 1v$	$u \in \{0, 1\}^*$ (3)	$n_0 <= n_1$ (4)

Distinción de casos:

$$S \to 0A_0|1E$$
$$E \to CC1|D0$$

Casos (1) y (2):

$$A_0 \to 0A_x|1A_0A_0$$

$$A_1 \to 1A_x|0A_1A_1$$

$$A_x \to 1A_0|0A_1|\varepsilon$$

Caso (3):

$$D \to BD|\varepsilon$$

$$B \to 0|1$$

Caso (4):

$$C \to \varepsilon|0|C1C$$

Ejercicio 16. Una gramática independiente del contexto generalizada es una gramática en el que las producciones son de la forma $A \to r$ donde r es una expresión regular de variables y símbolos terminales. Una gramática independiente del contexto generalizada representa una forma compacta de representar una gramática con todas las producciones $A \to \alpha$, donde α es una palabra del lenguaje asociado a la expresión regular r y $A \to r$ es una producción de la gramática generalizada. Observemos que esta gramática asociada puede tener infinitas producciones, ya que una expresión regular puede representar un lenguaje con infinitas palabras. El concepto de lenguaje generado por una gramática generalizada se define de forma análaga al de las gramáticas independientes del contexto, pero teniendo en cuenta que ahora puede haber infinitas producciones. Demostrar que un lenguaje es independiente del contexto si y solo si se puede generar por una gramática generalizada.

Llamaremos **producción clásica** a las producciones de la forma $A \to \alpha,\ \alpha \in (VUT)^*$ y **producción generalizada** a las producciones de la forma $A \to r$, donde r es una expresión regular.

Comencemos con la implicación hacia la derecha. Buscamos encontrar una gramática generaliza que genere L. Sean las producciones de G de la forma:

$$A_1 \to \alpha_{11} \mid \alpha_{12} \mid ... \mid \alpha_{1n_1}$$

$$A_2 \to \alpha_{21} \mid \alpha_{22} \mid ... \mid \alpha_{2n_2}$$

$$...$$

$$A_m \to \alpha_{m1} \mid \alpha_{m2} \mid ... \mid \alpha_{mn_m}$$

Con $A_i \in V, \alpha_{ij} \in (VUT)^*\ \forall i, j$.
Contruimos las siguientes expresiones regulares $r_i = \alpha_{i1} + \alpha_{i2} + ... + \alpha_{in_i}\ \forall i \in \{1,\ ...\ , m\}$.

Cada producción generalizada de la forma $A_i \to r_i$ representa todas las producciones clásicas de la gramática a partir de A_i. Por lo tanto, $\{A_i \to r_i : i \in \{1,\ ...\ , m\}\}$ es el conjunto de producciones de una gramática generaliza que genera L.

Resolvamos ahora la implicación hacia la izquierda. A partir de una gramática generalizada que genera L obtendremos una gramática independiente del contexto, mostrando así que L es independiente del contexto.

Las producciones de la gramática generalizada serán de la forma: $A_i \to r_i : i \in \{1,\ ...\ , n\}$. Veámos que podemos expresar cada una de estas reglas de producción generalizadas como un número finito de reglas de producción clásicas. Para ello utilizaremos el siguiente algoritmo. Sea P el conjunto de producciones, inicialmente $P = \{A_i \to r_i : i \in \{1, \ ... \ , n\}\}$, las producciones de la gramática.

Tomaremos una producción de P, $A \to r$, y haremos lo siguiente:

1. Si la producción ya es una producción clásica, no la modificamos y tomamos otra producción distinta de P.
2. Si la producción es de la forma $A \to r^*$, la eliminamos de P y añadimos $A \to rA \mid \varepsilon$.
3. Si la producción es de la forma $A \to r^+$, la eliminamos y añadimos $A \to rB$ y $B \to rB \mid \varepsilon$.
4. Si la producción es de la forma $A \to r_1 + r_2$, la eliminamos y añadimos $A \to r_1 \mid r_2$.
5. Si la producción es de la forma $A \to \alpha r$, donde $\alpha \in (VUT)^*$ la eliminamos y añadimos $A \to aB$ y $B \to r$.

De esta forma en cada paso o bien eliminamos un símbolo propio de expresión regular (r^* , r^+ , $r_1 + r_2$), o bien r decrece en uno o más elementos si aplicamos el último paso. Como todas las expresiones regulares son finitas llega un momento en el que todas las producciones de P son clásicas.

Las producciones de P componen una gramática libre del contexto que genera L, como buscábamos.

Ejercicio 17. Demostrar que los siguientes lenguajes son independientes del contexto:

1. $L_1 = \{u\#w \mid u^{-1} \text{ es una subcadena de } w, \ u, w \in \{0,1\}^*\}$
2. $L_2 = \{u_1\#u_2\#...\#u_k \mid k \geq 1, \text{ cada } u_i \in \{0,1\}^*, \text{ y para algún i y j }, u_i = u_j^{-1}\}$

Para el primer lenguaje la gramática resulta ser relativamente sencilla. Recordemos que la cadena vacía es subcadena de todas las cadenas del lenguaje, de ahí la producción $A \to \#B$:

$$S \to SX \mid A$$

$$X \to 0|1$$

$$A \to 0A0 \mid 1A1 \mid \#B$$

$$B \to XB$$

Para el segundo lenguaje vale la pena remarcar un par de situaciones:

- Puede darse $u_i = \varepsilon$ para cualquier i pues $u_i \in \{0,1\}^*$. Esto está reflejado en la producción $X \to \varepsilon$.
- Puede darse $u_i = u_j^{-1}$ para $i = j$ si u_i es un palíndromo. De ahí la producción $A \to \varepsilon$.
- Entre u_i y u_j puede no haber palabras ($A \to \#$) o haber un número indeterminado de palabras ($A \to \#C\#$).

Por último, la idea general del algoritmo es parecida al anterior: S genera todas las u_k antes de la palabra u_i; X genera cada u_k con $k \neq i, j$; A se encarga de generar u_i y u_j; C genera las posibles palabras u_k entre u_i y u_j; y finalmente B genera las palabras u_k que aparecen después de u_j.

$$S \to X\#S \mid AB$$

$$X \to 0X|1X|\varepsilon$$
$$A \to 0A0 \mid 1A1 \mid \varepsilon \mid \# \mid \#C\#$$
$$C \to X\#C \mid X$$
$$B \to \#XB \mid \varepsilon$$

Ejercicio 19. Sea el lenguaje $L = \{u\#v \ : \ u, v \in \{0,1\}^*, u \neq v\}$, demostrar que es independiente del contexto.

Este leguanje **no** es independiente del contexto. Intuitivamente, necesitaríamos crear distintos elementos en dos puntos distintos de la palabra que no mantienen relación para saber si esta pertenece al lenguaje. Para mostrar que no lo es usaremos el lema de bombeo para lenguajes independientes del contexto. Enunciémoslo:

Sea L un lenguaje independiente del contexto. Entonces $\exists n \in N$ tal que $\forall x \in L, |x| \geq n$ existe una descomposición $x = yzuvw$ cumpliendo:

1. $|zuv| \leq n$
2. $|zv| > n$
3. $\forall i \geq 0,\ yz^iuv^iw \in L$

Negémoslo. Sea L un lenguaje, si $\forall n \in N, \exists x \in L, |x| \geq n$ con una descomposición $x = yzuvw$ cumpliendo:

1. $|zuv| \leq n$
2. $|zv| > n$
3. Y sin embargo $\exists i \geq 0$ tal que $yz^iuv^iw \notin L$

Entonces L no es independiente del contexto.

En nuestro caso, sea $x = 0^n\#0^n$, $y = w = 0^n$, $z = \#$ y $u = v = \varepsilon$. Obviamente $x = yzuvw$. Además, $|zuv| = 1 \leq n \ \forall n$, y $|zv| = 1 > 0$.

Tomando $i = 2$, $yz^iuv^iw = 0^n\#\#0^n \notin L$. Por tanto L no es independiente del contexto.

Ejercicio 21. Dar gramáticas independientes del contexto no ambiguas para los siguientes lenguajes sobre el alfabeto $\{0, 1\}$:

- El conjunto de palabras w tal que en todo prefijo de w el número de 0s es mayor o igual que el número de 1s.

$$S \to 0A \mid \varepsilon$$
$$A \to 1S \mid 0AA \mid \varepsilon$$

- El conjunto de palabras w en las que el número de 0s es mayor o igual que el número de 1s.

$$S \to 0A \mid A0 \mid \varepsilon \mid A0S0A$$
$$A \to 1 \mid A0A \mid \varepsilon$$

En este caso las gramáticas son sencillas: cada A te permite colocar como máximo un 1. El único detalle a destacar reside en la producción $A \rightarrow A0S0A$, ya que sin esta producción todas las palabras o bien comenzarían por 0, o bien terminarían por 0.