

## Capítulo 12.

### *Propiedades de lenguajes independientes del contexto*

#### *12.1. Identificación de lenguajes independientes del contexto*

Lema de bombeo.

#### *12.2. Propiedades*

Cierre, Complemento de lenguajes, Sustitución, Algoritmos de decisión.

## 12.1. Identificación de Lenguajes independientes del contexto

---

### Lema de Bombeo

#### Teorema:

Sea  $L$  un lenguaje independiente del contexto. Entonces existe una constante  $n$  tal que, cualquier palabra  $x \in L$  con  $|x| \geq n$  puede escribirse de la forma  $z=y.z.u.v.w$ , con las siguientes condiciones:

1.  $|z.u.v| \leq n$  (la parte central no es demasiado larga)
2.  $|z.v| > 0$  (por lo menos uno de los trozos que se “bombean” no está vacío)
3. Para todo  $i \geq 0$ ,  $y.z^i.u.v^i.w \in L$ . Es decir, las cadenas  $z$  y  $v$  pueden ser “bombeadas” y la cadena resultante es una palabra de  $L$ .

#### Ejemplos:

$$L = \{a^n b^n \mid n \geq 1\}$$

$$L = \{xx^R \mid x \in \{a,b\}^*, x^R \text{ imagen inversa de } x\}$$

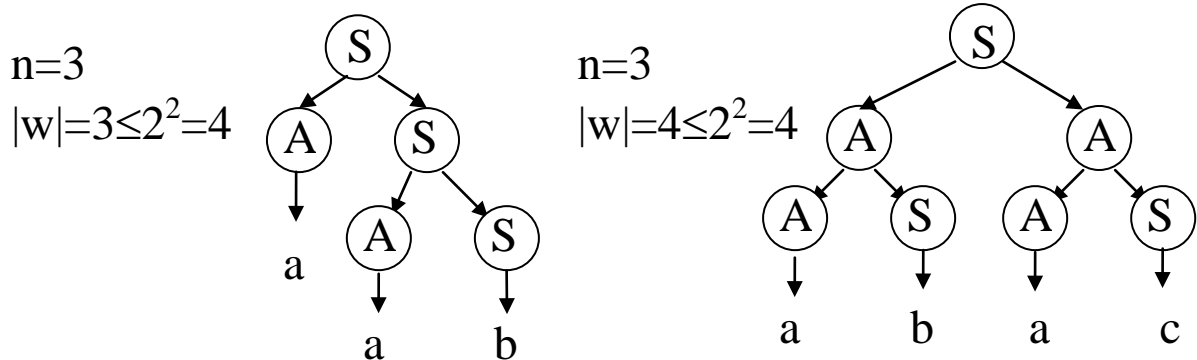
## Tamaño de los árboles de derivación

### Teorema:

Sea  $G=(\Sigma_T, \Sigma_N, S, P)$  una GIC en FNC. Sea  $w \in \Sigma_T^*$  una palabra de  $L(G)$  y  $A$  un árbol de derivación cuyo resultado es  $w$ .

Si la profundidad del árbol es  $n$ , entonces la longitud de  $w$  es menor o igual a  $2^{n-1}$ , es decir,  $|w| \leq 2^{n-1}$ .

Ejemplo: Árboles de GIC's en FNC son binarios.



Profundidad: la longitud del camino más largo del árbol

Camino: sucesión de arcos desde la raíz hasta un nodo hoja

Longitud de un camino: número de sus arcos

### **Demostración:** (por inducción)

*BASE:*  $n=1$ . Por tanto, el árbol solo tiene dos niveles. Por definición de los árboles de derivación y dado que  $G$  está en FNC, un árbol de dos niveles cuyo resultado es una palabra  $w \in \Sigma_T^*$  tiene solo dos nodos: la raíz ( $S$ ) y un nodo etiquetado con un símbolo terminal  $a \in \Sigma_T$ , y su resultado es  $a$ . Por lo tanto,  $w=a$  y  $|w|=1$ . Puesto que  $2^{n-1}=2^0=1$ , se cumple  $|w| \leq 2^{n-1}$ .

### *PASO INDUCTIVO:*

Suponiendo que el teorema se cumple para árboles de profundidad  $n$ , se demuestra que se cumple también para árboles de profundidad  $n+1$ .

### *hipótesis inductiva:*

Dado un árbol de profundidad  $n$  con resultado  $w \in \Sigma_T^*$ , se cumple  $|w| \leq 2^{n-1}$ .

*paso de  $n$  a  $n+1$ :* (se demuestra que, para un árbol de profundidad  $n+1$  con resultado  $w$ , se cumple  $|w| \leq 2^n$ )

Sea un árbol de profundidad  $n+1$ , se demuestra que como  $n+1 > 1$ , la raíz del árbol tiene que usar una producción  $S \rightarrow AB$  (no se podría usar  $S \rightarrow a$  siendo  $a$  un terminal).

Por tanto la derivación de  $w$  tiene la forma  $S \rightarrow AB \rightarrow^* w$  y se puede descomponer en  $w=u.v$  tal que  $A \rightarrow^* u$  y  $B \rightarrow^* v$ .

Obviamente, la profundidad de los subárboles (para  $A \rightarrow^* u$  y  $B \rightarrow^* v$ ) no puede ser mayor que  $n$ , puesto que la profundidad del árbol entero es  $n+1$  y se excluye el primer nivel.

Por la hipótesis de inducción, se cumple que  $|u| \leq 2^{n-1}$  y  $|v| \leq 2^{n-1}$ . Dado que  $w=u.v$ , se sigue que  $|w| = |u| + |v| \leq 2^{n-1} + 2^{n-1} = 2^n$ .

**Corolario:**

Sea  $G=(\Sigma_T, \Sigma_N, S, P)$  una GIC en FNC. Sea  $w \in \Sigma_T^*$  cualquier palabra de  $L(G)$ . Sea  $m$  un número tal que  $|w| \geq 2^m$ .

La profundidad de cualquier árbol de derivación para la palabra  $w$  es mayor o igual a  $m+1$ .

**Demostración:**

Según el teorema de antes, un árbol de derivación de profundidad  $m$  solo puede tener un resultado  $w$  con  $|w| \leq 2^{m-1}$ .

Por tanto, para derivar una palabra  $w$  con  $|w| \geq 2^m$  se requiere un árbol de profundidad mayor o igual a  $m+1$ .

## **Demostración del Lema de bombeo:**

*CASOS ESPECIALES:*  $L=\emptyset$ ,  $L$  es finito

No se viola el lema.

Elegimos cualquier  $n>0$ , tal que,  $|x|<n$ , para cualquier  $x\in L$ .

El lema dice que cualquier palabra  $x$  con  $|x|\geq n$  cumple las condiciones, pero como no hay palabras con  $|x|\geq n$  no se pueden violar las condiciones.

*CASO NORMAL:*  $L$  es infinito

Sea  $G=(\Sigma_T, \Sigma_N, S, P)$  una GIC en FNC que genera este lenguaje. (dado que  $L$  es un lenguaje independiente del contexto, existe una GIC que genera  $L$ , y ya se ha demostrado también que toda GIC tiene una GIC en FNC equivalente).

Si  $L$  contiene la palabra vacía, se demuestra el lema para  $L-\{\lambda\}$  (se quita la regla  $S::=\lambda$  de  $G$ ).

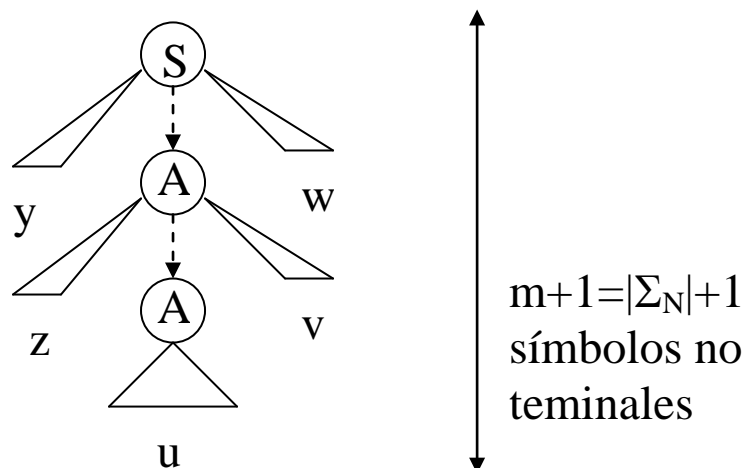
$\Rightarrow$  En general, como constante del lema se elige  $n>0$  por lo que la palabra  $\lambda$  con  $|\lambda|=0$  no tiene importancia.

Supongamos que  $G$  tiene  $m$  variables ( $|\Sigma_N|=m$ ).

Elegimos  $n=2^m$  y suponemos una palabra genérica  $x \in L$  con  $|x| \geq n$  (es decir,  $|x| \geq 2^m$ ).

Según el corolario, para derivar  $x$ , se requiere un árbol de profundidad mayor o igual a  $m+1$ , es decir con al menos un camino que pasa por  $m+1=|\Sigma_N|+1$  nodos con símbolos no terminales. Como sólo hay  $|\Sigma_N|$  símbolos no terminales en  $G$ , por lo menos uno estará repetido en dicho camino.

Árbol de derivación de  $x$ :



*Primera condición:*

Sea A el primer símbolo no terminal que se repite (visto desde abajo) en el camino más largo (o uno de ellos).

Es decir, en el camino más largo del subárbol desde A no se repite ningún otro símbolo no terminal salvo A (y solo una vez).

La profundidad de este subárbol es menor o igual que

$m+1=|\Sigma_N|+1$  (como mucho aparecen todos los símbolos no terminales y uno dos veces) y el resultado del subárbol desde A es  $z.u.v$ .

De acuerdo con el teorema de antes, cualquier resultado de un árbol de profundidad menor o igual que  $m+1$ , tiene una longitud menor o igual a  $2^m$ . Por tanto, se verifica  $|z.u.v| \leq 2^m$ , es decir,  $|z.u.v| \leq n$ .

*Tercera condición:*

El árbol esquemático corresponde a una derivación:

$$S \rightarrow^* yAw \rightarrow^* yzAvw \rightarrow^* yzuvw$$

Por tanto, existen las derivaciones  $A \rightarrow^* zAv$  y también  $A \rightarrow^* u$ .

Usando solo la segunda, desde S también se puede derivar:

$$S \rightarrow^* yAw \rightarrow^* yuw$$

que equivale a la palabra  $yz^iuv^i w$  con  $i=0$ .

Además, se puede derivar:

$$S \rightarrow^* yAw \rightarrow^* yzAvw \rightarrow^* yzzAvvw \rightarrow^* \dots \rightarrow^* yz^iAv^i w \rightarrow^* yz^iuv^i w$$

es decir todas las palabras  $yz^iuv^i w$  para  $i=1,2,\dots$

*Segunda condición:*

Dado que G no contiene reglas unitarias ni reglas- $\lambda$  la derivación  $A \rightarrow^* zAv$  no puede tener la forma  $A \rightarrow^* A$ , eso es, por lo menos uno, z o v, debe ser distinto de  $\lambda$ . Por tanto se cumple  $|z.v| > 0$ .

Estos tres pasos demuestran el lema.



## Aplicación del lema de Bombeo

¿Para que sirve el lema de bombeo?

⇒ Para demostrar que ciertos lenguajes no son independientes del contexto.

No se puede demostrar que un lenguaje es independiente del contexto. Hay lenguajes que no son independientes del contexto pero sí cumplen el lema de bombeo.

PERO: no hay ningún lenguaje independiente del contexto que no cumpla el lema.

Para demostrar que un lenguaje no es independiente del contexto basta probar que no cumple el lema de bombeo. (A veces funciona a veces no.)

### Método:

Por reducción al absurdo.

Se supone que  $L$  es independiente del contexto (por tanto debe cumplir el lema de bombeo) y se demuestra que no lo cumple.

Hay que demostrar que para cualquier  $n$  existe un  $x \in L$  con  $|x| \geq n$  tal que para cualquier descomposición de  $x = y.z.u.v.w$  con  $|z.v| > 0$  y  $|z.u.v| \leq n$  existe algún  $y.z^i.u.v^i.w$  (con  $i=0,1,2,\dots$ ) que no pertenece a  $L$ .

**Ejemplo:**  $L = \{a^m b^m c^m \mid m > 0\}$

1. supongamos que  $L$  es independiente del contexto.
2. elegimos un  $n$  arbitrario como constante del lema
3. seleccionamos un  $x \in L$  con  $|x| \geq n$ :  $x = a^n b^n c^n$
4. Descomponemos  $x = a^n b^n c^n = y.z.u.v.w$  de todas formas posibles donde  $|z.v| > 0$  y  $|z.u.v| \leq n$ .

Dado la última condición ( $|z.u.v| \leq n$ ) **no** es posible que  $z.u.v$  contenga  $a$ 's y  $c$ 's a la vez. Además, dado que  $|z.v| > 0$  por lo menos uno de los dos tiene que ser distinto de  $\lambda$ .

Para que el lenguaje cumpla el lema de bombeo, debe ser posible bombear las palabras en  $z$  y  $v$  ( $y.z^i.u.v^i.w$ ), añadiendo el mismo número de  $a$ 's,  $b$ 's y  $c$ 's.

Sin embargo, por las dos condiciones de arriba, cualquiera cadenas que sean  $z$  y  $v$ , al bombearlos ( $z^i$  y  $v^i$ ) no se puede añadir nunca el mismo número de  $a$ 's,  $b$ 's y  $c$ 's.

Por lo tanto no se cumple el lema de bombeo.

$\Rightarrow$  La suposición de que  $L$  fuese independiente del contexto es errónea.

$\Rightarrow L$  no es independiente del contexto

## 12.2. Propiedades

---

### Propiedades de cierre

#### Clausura, concatenación y unión

##### **Teorema:**

El conjunto de lenguajes independientes del contexto ( $L_{ic\_Σ}$ ) es cerrado respecto a la concatenación, la unión y la clausura.

##### **Demostración:**

Sean las gramáticas  $G^1=(Σ^1_T,Σ^1_N,S^1,P^1)$  y  $G^2=(Σ^2_T,Σ^2_N,S^2,P^2)$  dos GIC que representan a dos lenguajes indep. del contexto  $L(G^1)$  y  $L(G^2)$ . Supondremos que  $Σ^1_N$  y  $Σ^2_N$  no tienen símbolos comunes.

### 1. Concatenación:

Construimos una gramática  $G=(\Sigma^1_T \cup \Sigma^2_T, \Sigma^1_N \cup \Sigma^2_N \cup \{S\}, S, P)$  donde:

$$P=P^1 \cup P^2 \cup \{S ::= S^1 S^2\} \text{ y } S \notin \Sigma^1_N \cup \Sigma^2_N$$

(Obviamente  $G$  es una gramática independiente del contexto.)

Vamos a demostrar que una palabra está en  $L(G)$  si y solo si está en  $L(G^1).L(G^2)$ . (Se demuestra que  $L(G)=L(G^1).L(G^2)$ ).

*SI:*  $(L(G) \supseteq L(G^1).L(G^2))$

Sea  $z.y$  con  $z \in L(G^1)$  e  $y \in L(G^2)$ , es decir,  $z.y \in L(G^1).L(G^2)$ .

Por tanto existen las derivaciones  $S^1 \rightarrow^* z$  y  $S^2 \rightarrow^* y$  en  $G^1$  y  $G^2$ , respectivamente. Por tanto, considerando  $G$ , existe la derivación  $S \rightarrow S^1 S^2 \rightarrow^* z S^2 \rightarrow^* zy$  y se sigue que  $z.y \in L(G)$ .

*SOLO SI:*  $(L(G) \subseteq L(G^1).L(G^2))$

Sea  $x \in L(G)$ . Con  $G$  existe la derivación  $S \rightarrow S^1 S^2 \rightarrow^* x$ . Para que eso sea así y dado que  $G$  es una gramática independiente del contexto, debe existir alguna descomposición  $x=z.y$  con  $S^1 \rightarrow^* z$  y  $S^2 \rightarrow^* y$ . Eso, a su vez, implica que  $z \in L(G^1)$  e  $y \in L(G^2)$  y, por tanto, se verifica que  $x=z.y$  pertenece a  $L(G^1).L(G^2)$ .

## 2. Unión:

Construimos una gramática  $G=(\Sigma^1_T \cup \Sigma^2_T, \Sigma^1_N \cup \Sigma^2_N \cup \{S\}, S, P)$  donde:

$$P=P^1 \cup P^2 \cup \{S ::= S^1 | S^2\} \text{ y } S \notin \Sigma^1_N \cup \Sigma^2_N$$

(Obviamente  $G$  es una gramática independiente del contexto.)

Vamos a demostrar que una palabra está en  $L(G)$  si y solo si está  $L(G^1) \cup L(G^2)$ . (Se demuestra que  $L(G) = L(G^1) \cup L(G^2)$ ).

*SI:*  $(L(G) \supseteq L(G^1) \cup L(G^2))$

*Parte 1:*  $(L(G) \supseteq L(G^1))$

Sea  $z \in L(G^1)$ . Por tanto existen las derivaciones  $S^1 \rightarrow^* z$  en  $G^1$ . Por tanto, considerando  $G$ , existe la derivación  $S \rightarrow S^1 \rightarrow^* z$  y se sigue que  $z \in L(G)$ .

*Parte 2:*  $(L(G) \supseteq L(G^2))$

Sea  $z \in L(G^2)$ . Por tanto existen las derivaciones  $S^2 \rightarrow^* z$  en  $G^2$ . Por tanto, considerando  $G$ , existe la derivación  $S \rightarrow S^2 \rightarrow^* z$  y se sigue que  $z \in L(G)$ .

*SOLO SI:*  $(L(G) \subseteq L(G^1) \cup L(G^2))$

Sea  $x \in L(G)$ . Por construcción de  $G$  existe una de las siguientes derivaciones  $S \rightarrow S^1 \rightarrow^* x$  o  $S \rightarrow S^2 \rightarrow^* x$ . Es decir, existe una derivación  $S^1 \rightarrow^* x$  o  $S^2 \rightarrow^* x$ . Eso, a su vez, implica que  $x \in L(G^1) \cup L(G^2)$ .

### 3. Clausura:

Construimos una gramática  $G=(\Sigma_T^1, \Sigma_N^1 \cup \{S\}, S, P)$  donde:

$$P=P^1 \cup \{S ::= S^1 S \mid \lambda\} \text{ y } S \notin \Sigma_N^1$$

(Obviamente  $G$  es una gramática independiente del contexto.)

Vamos a demostrar que una palabra está en  $L(G)$  si y solo si está  $L(G^1)^*$ . (Se demuestra que  $L(G) = L(G^1)^*$ ).

*SI:*  $(L(G) \supseteq L(G^1)^*)$

Sea  $z \in L(G^1)^*$ . Por definición de  $L(G^1)^*$ , se cumple que  $z = \lambda$  o  $z = z_1 z_2 \dots z_n$  con  $z_i \in L(G^1)$ .

*Caso 1:*  $z = \lambda$

En  $G$  existe la derivación  $S \rightarrow \lambda$ , por lo que  $z \in L(G)$ .

*Caso 2:*  $z = z_1 z_2 \dots z_n$  con  $z_i \in L(G^1)$

Obviamente, para cada  $z_i$  existe una derivación  $S^1 \rightarrow^* z_i$  en  $G^1$ . Considerando  $G$ , existe la derivación

$$S \rightarrow S^1 S \rightarrow^* z_1 S \rightarrow z_1 S^1 S \rightarrow^* \dots \rightarrow z_1 \dots z_n S \rightarrow z_1 \dots z_n$$

por lo que se sigue que  $z = z_1 \dots z_n \in L(G)$ .

*SOLO SI:*  $(L(G) \subseteq L(G^1)^*)$  :

Sea  $x \in L(G)$ . Por construcción de  $G$  se deriva  $x$  de una de las siguientes derivaciones:  $S \rightarrow \lambda$  o

$$S \rightarrow S^1 S \rightarrow^* S^1 S^1 \dots S^1 S \rightarrow S^1 S^1 \dots S^1 \rightarrow^* x.$$

*Caso 1:*  $S \rightarrow \lambda$

Por definición  $\lambda$  también pertenece a  $L(G^1)^*$ .

*Caso 2:*  $S \rightarrow S^1 S \rightarrow^* S^1 S^1 \dots S^1 S \rightarrow S^1 S^1 \dots S^1 \rightarrow^* x$

Dado que  $G$  es una GIC, se sigue que  $x = x_1 x_2 \dots x_n$  tal que para cada  $x_i$  existe la derivación  $S^1 \rightarrow^* x_i$ . Por la construcción de  $G$  se sigue que todos los  $x_i$  pertenecen a  $L(G)$ . Dada la definición de  $L(G^1)^* =$ , se verifica que  $x = x_1 x_2 \dots x_n \in L(G^1)^*$ .

## Intersección y Complemento

### Teorema:

El conjunto de lenguajes independientes del contexto no es cerrado respecto a la intersección y la complementación.

### Demostración:

1. *Intersección* (Basta encontrar un contraejemplo.):

Sean  $G^1$  y  $G^2$  dos GICs

$S ::= AB$

$C ::= DE$

$A ::= aAb \mid ab$

$E ::= bEc \mid bc$

$B ::= cB \mid c$

$D ::= aD \mid a$

representando  $L(G^1) = \{a^n b^n c^m \mid m, n > 0\}$  y  $L(G^2) = \{a^m b^n c^n \mid m, n > 0\}$ .

Obviamente,  $L(G^1)$  y  $L(G^2)$  son lenguajes independientes del contexto y su intersección  $L(G^1) \cap L(G^2)$  es el lenguaje  $L = \{a^n b^n c^n \mid n > 0\}$ .

Ya hemos demostrado que  $L$  no es un lenguaje independiente del contexto (lema de bombeo).

2. *Complementación* (por reducción al absurdo):

Suponiendo que  $\forall L \in L_{ic\_Σ}: c(L) \in L_{ic\_Σ}$ .

Sean  $L_1, L_2 \in L_{ic\_Σ}$ . Por la suposición se sigue que  $c(L_1)$  y  $c(L_2) \in L_{ic\_Σ}$ . Por tanto, como ya hemos visto (unión), se verifica que también  $c(L_1) \cup c(L_2) \in L_{ic\_Σ}$ . Y con la suposición de arriba se verifica que  $c(c(L_1) \cup c(L_2)) \in L_{ic\_Σ}$ .

De acuerdo con las leyes de De Morgan:  $c(c(L_1) \cup c(L_2)) = L_1 \cap L_2$ .

Se verifica  $\forall L_1, L_2 \in L_{ic\_Σ}: L_1 \cap L_2 \in L_{ic\_Σ}$ . Ya hemos visto que eso es falso, por lo que se concluye que la suposición inicial es incorrecta, es decir, el conjunto de lenguajes independientes del contexto **no** está cerrado respecto a la complementación.

## Más propiedades:

### Teorema:

Sea  $L_1$  y  $L_2$  dos lenguajes independientes del contexto y  $L_3$  un lenguaje regular. Se verifica lo siguiente:

1.  $L_1 \cap L_3$  es un lenguaje independiente del contexto.
2.  $L_1 - L_3$  es un lenguaje independiente del contexto.
3.  $L_1 - L_2$  no necesariamente es un lenguaje independiente del contexto.

(nos ahorramos las demostraciones)



## Sustituciones

Sean  $\Sigma$  y  $\Theta$  dos alfabetos. Se define una *sustitución*  $s$  sobre  $\Sigma$  en  $\Theta$  como la aplicación  $s: \Sigma \rightarrow P(\Theta^*)$ , que asigna a cada símbolo  $a$  de  $\Sigma$  un lenguaje  $L_a \subseteq \Theta^*$ .

La sustitución se extiende a palabras  $x = x_1 x_2 \dots x_n$  de la siguiente forma:

$$s(x_1 x_2 \dots x_n) = s(x_1).s(x_2) \dots s(x_n) \text{ (concatenación de lenguajes)}$$

Se extiende  $s$  a lenguajes:

$$s(L) = \bigcup_{x \in L} s(x)$$

### Ejemplo:

Lenguaje  $L$  definido por  $E ::= (E) \mid E + E \mid E * E \mid i$

Sustitución:

$$s(( ) = \{ (, s() = \{ ) \}, s(+) = \{ + \}, s(*) = \{ * \},$$

$$s(i) = \text{lenguaje definido por } 0 + ((1 + \dots + 9)(0 + \dots + 9)^*)$$

$$L = \{ i, (i), i+i, i*i, (i+i)*i, \dots \}$$

$$s(L) = \{ 2, (45), 67+3, 9*0, (9+33)*99, \dots \}$$

### Teorema:

Sea  $L$  un lenguaje independiente del contexto sobre el alfabeto  $\Sigma$  y  $s$  una sustitución sobre  $\Sigma$  en  $\Theta$ . Si para cada  $a \in \Sigma$ ,  $s(a)$  es un lenguaje independiente del contexto, entonces  $s(L)$  es un lenguaje independiente del contexto.

### Idea de la demostración:

Por construcción de una gramática para  $s(L)$  a partir de las GIC:

$$G_L = (\Sigma, \Sigma_{NL}, S, P) \text{ y } G_a = (\Theta, \Sigma_{Na}, S_a, P_a) \text{ para todo } a \in \Sigma.$$

## Algunos algoritmos de decisión para lenguajes generados por una GIC

### Lema

Existe un algoritmo para reconocer si el lenguaje generado por una GIC es vacío.

### Demostración:

Sea la GIC  $G = (\Sigma_T, \Sigma_N, S, P)$ . Obviamente, si  $L(G) = \emptyset$ , no existe ninguna derivación de la forma  $S \rightarrow^* x$ , con  $x \in \Sigma_T^*$ , es decir,  $S$  es un símbolo superfluo.

Existe un algoritmo para eliminar los símbolos superfluos de cualquier GIC.

Algoritmo (para comprobar si  $L(G) = \emptyset$ ):

1. Aplicar el algoritmo de eliminación de símbolos superfluos a  $G$ , obteniendo  $G' = (\Sigma'_T, \Sigma'_N, S', P')$ .
2.  $L(G)$  es vacío si y solo si en este algoritmo se elimina el símbolo  $S$  como símbolo superfluo.

Ejemplo:

$G = ( \{a,b\} \{A,B,S\}, S, \{ S ::= bA \mid baS \mid B, A ::= Aa, B ::= Bb \} )$

## Lema

“Existe un algoritmo para reconocer si el lenguaje generado por una GIC es infinito”

## Demostración

Sea la GIC  $G = (\Sigma_T, \Sigma_N, S, P)$ .

Se define el algoritmo:

Primero, se usa el algoritmo anterior para ver si  $L(G)$  es vacío.

Si lo es, entonces  $L$  es finito (vacío). Si no lo es, entonces seguimos:

Hay dos pasos:

1. Primero se convierte  $G$  en una gramática bien formada  $G' = (\Sigma'_T, \Sigma'_N, S', P')$ .
2. Construimos un grafo cuyos nodos son los símbolos de  $\Sigma'_N$ . Para cada par de nodos  $(A, B)$ , se añade un arco de  $A$  a  $B$  si existe una regla  $A ::= \alpha B \beta$ , con  $\alpha, \beta \in \Sigma'^*$  y  $B \in \Sigma'_N$  ( $A$  y  $B$  pueden ser iguales)
3. Si existen ciclos en este grafo, entonces  $L(G)$  es infinito. En caso contrario  $L(G)$  es finito.

(Los ciclos corresponden a derivaciones de la forma  $A \rightarrow^* \alpha A \beta$ , con  $|\alpha| + |\beta| > 0$ ).

Ejemplo:

$G = (\{a, b\} \cup \{A, B, S\}, S, \{ S ::= bA \mid baS \mid B, A ::= Aa, B ::= bb \mid b \})$