



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INFORMÁTICA Y ANÁLISIS NUMÉRICO

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SEGUNDO CURSO, SEGUNDO CUATRIMESTRE



TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

Tema 10.- Lenguajes de Contexto Libre



Teorema previo al “lema de bombeo”

Si

- $G = (V_N, V_T, P, S)$ está en la **FNC**
- $z \in L(G)$
- T es el árbol sintáctico asociado a $S \Rightarrow^* z$
- i es la longitud del camino más largo de T

entonces

- $|z| \leq 2^{(i-1)}$

Teorema previo al “lema de bombeo”

Demostración por inducción sobre “ i ”

- Si ($i = 1$) entonces

- la derivación ha de ser de la forma $S \Rightarrow z = \mathbf{a} \in V_T$

- $|z| = |a| = 1 = 2^0 = 2^{(i-1)}$

Por tanto $|z| \leq 2^{(i-1)}$

Teorema previo al “lema de bombeo”

Demostración por inducción sobre “ i ”

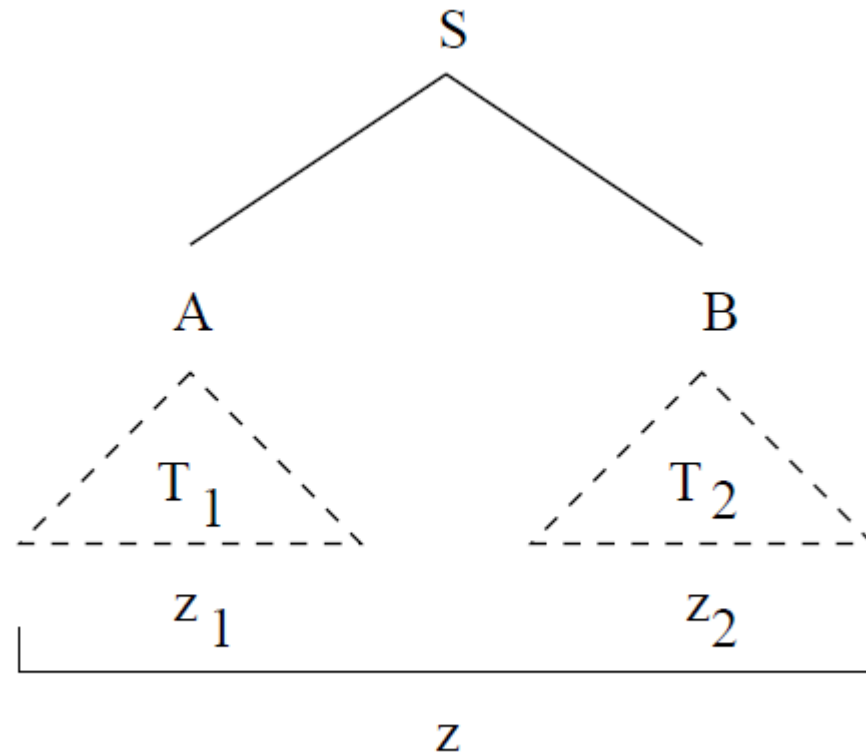
- Se verifica para $i = k \geq 1$
- ¿Se verifica para $i = k+1$?

Sea

$S \Rightarrow A B$

$\Rightarrow^{k_1} z_1 B \Rightarrow^{k_2} z_1 z_2 = z$

donde $k_1 + k_2 = k$



Estructura del árbol sintáctico

Teorema previo al “lema de bombeo”

- Como $k_1, k_2 < k$, por hipótesis de **inducción**, se verifica que

$$\circ |z_1| \leq 2^{(k_1-1)}$$

$$\circ |z_2| \leq 2^{(k_2-1)}$$

$$\begin{aligned} \circ |z| = |z_1 z_2| &= |z_1| + |z_2| \leq 2^{(k_1-1)} + 2^{(k_2-1)} \\ &< 2^{k_1} + 2^{k_2} = 2^{(k_1+k_2)} = 2^k = 2^{(k+1-1)} = 2^{(i-1)} \end{aligned}$$

Por tanto

$$|z| \leq 2^{(i-1)}$$

Corolario

Si

- $G = (V_N, V_T, P, S)$ está en la **FNC**
- cardinal $(V_N) = k$
- $n = 2^k$
- $z \in L(G)$ y $|z| \geq n = 2^k$

entonces

- *cualquier árbol sintáctico de una derivación que genera a **z** tiene, al menos, una **longitud mayor o igual** a $k + 1$*

Corolario

Demostración

Si

- T es el árbol sintáctico asociado a una derivación de z
- y la longitud de su camino más largo es "i"

entonces

$$n = 2^k \leq |z| \leq 2^{(i-1)}$$

- y, por tanto, se verifica que **$i \geq k + 1$**

Lema de bombeo

Si L es un lenguaje de contexto libre entonces

- $\exists n$, que sólo depende de L
- $\forall z \in L \wedge |z| \geq n$

entonces

- $z = u v w x y$

donde

- a) $|v x| \geq 1$
- b) $|v w x| \leq n$
- c) $\forall i \geq 0, z_i = u v^i w x^i y \in L$

Lema de bombeo

Demostración

- $L - \{\varepsilon\} = L(G)$, donde $G = (V_N, V_T, P, S)$ está en la **FNC**
- Sea **$k = \text{cardinal}(V_N)$** y **$n = 2^k$**
- Si $z \in L(G) \wedge |z| \geq n = 2^k > 2^{(k-1)}$
entonces
- el **camino más largo** del árbol sintáctico asociado a la derivación de z tiene, al menos, una longitud igual o mayor a **$k+1$** (corolario)

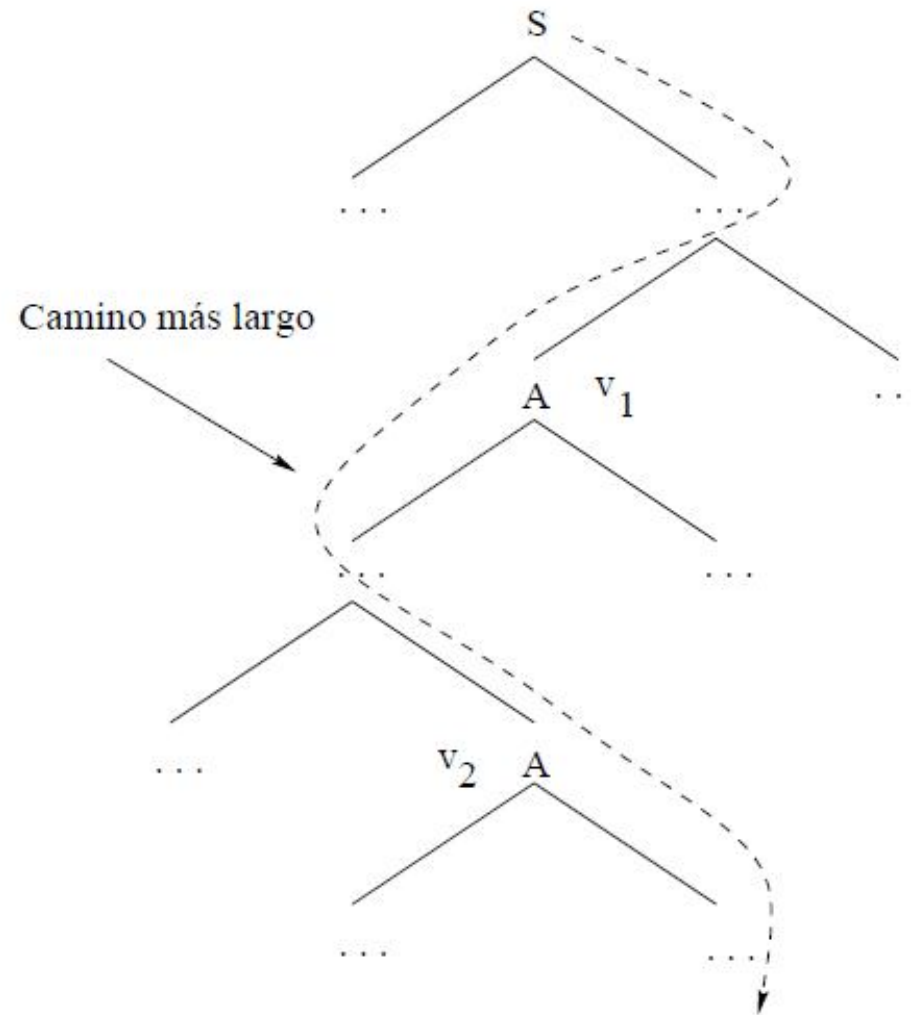
Lema de bombeo

Demostración

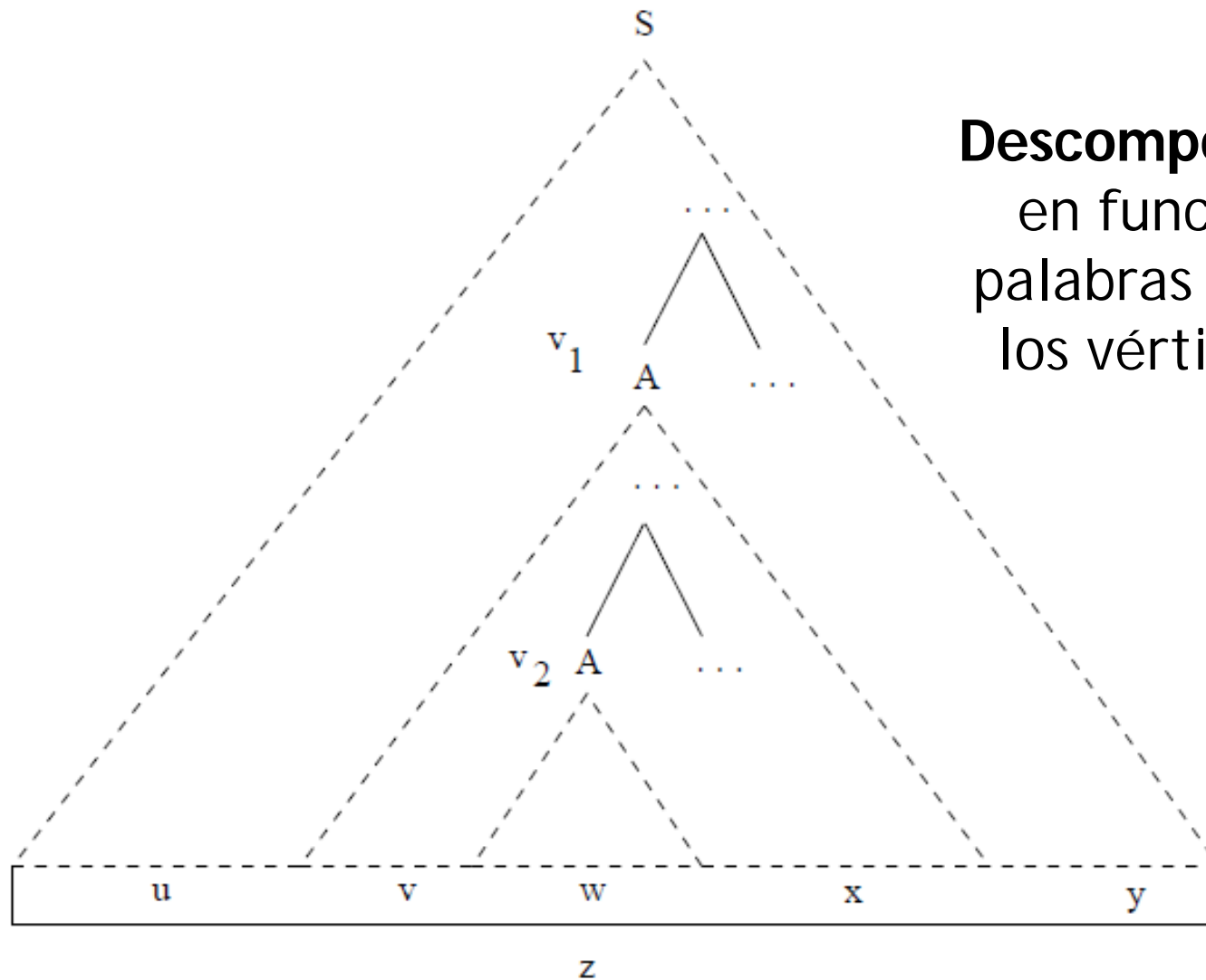
- ...
- En el camino más largo hay “ $k+2$ ” vértices (símbolos)
 - 1 símbolo terminal
 - $k+1$ no terminales
- V_N sólo tiene k símbolos
- Por tanto, el camino tiene **vértices repetidos**

Lema de bombeo

Últimos vértices que repiten su símbolo en el camino más largo del árbol sintáctico



Lema de bombeo



Descomposición de z
 en función de las
 palabras asociadas a
 los vértices v_1 y v_2

Lema de bombeo

Demostración

Se verifica que

- $S \Rightarrow^* u A y$
- $A \Rightarrow^* v A x$
- $A \Rightarrow^+ w$

y, por tanto,

$$S \Rightarrow^* u A y \Rightarrow^* u v A x y \Rightarrow^+ u v w x y = z$$

Lema de bombeo

Demostración

$$z = u v w x y$$

a) $|v x| \geq 1$

- La primera regla de producción del **nodo** v_1 ha de ser de la forma $A \rightarrow B C$
- El nodo v_2 ha de estar contenido, por ejemplo, en el subárbol de B (o de C):
 - $A \Rightarrow B C \Rightarrow^* v A x_1 C \Rightarrow^+ v A x_1 x_2 = v A x$
 - $C \Rightarrow^+ x_2 \in V_T^+, |x_2| \geq 1$
- Por tanto, $|v x| \geq 1$

Lema de bombeo

Demostración

$$z = u v w x y$$

$$b) |v w x| \leq n$$

- El subárbol asociado al nodo v_1 tiene una longitud menor o igual a $k + 1$
- v_1 tiene asociada la derivación

$$A \Rightarrow^+ v A x \Rightarrow^* v w x$$

- Por tanto, aplicando el teorema,

$$|v w x| \leq 2^{(k+1)-1} = 2^k = n$$

Lema de bombeo

Demostración

$$z = u v w x y$$

$$c) \forall i \geq 0, z_i = u v^i w x^i y \in L$$

$$S \Rightarrow^* u A y$$

$$\Rightarrow^+ u v A x y$$

$$\Rightarrow^+ u v^2 A x^2 y$$

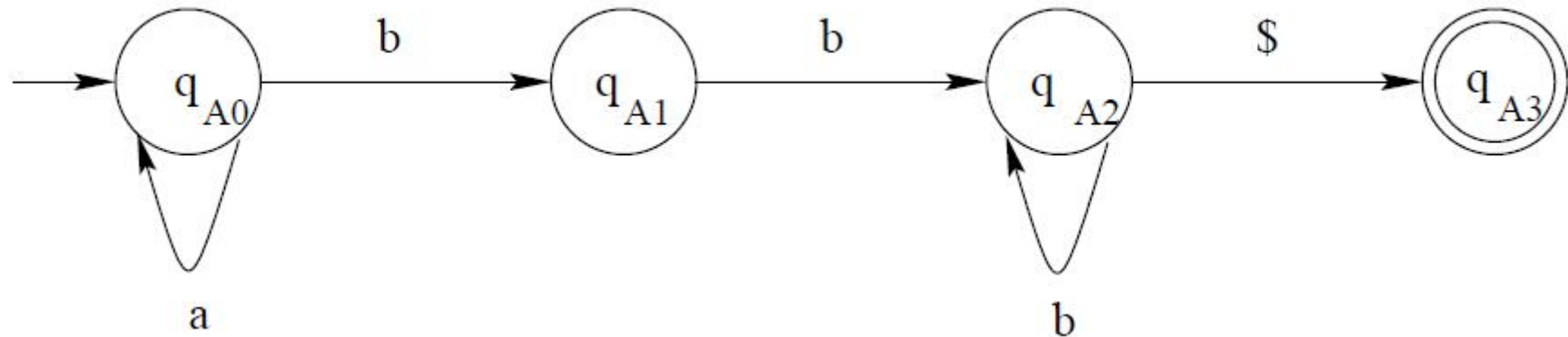
...

$$\Rightarrow^+ u v^i A x^i y$$

$$\Rightarrow^+ u v^i w x^i y = z_i \in L$$

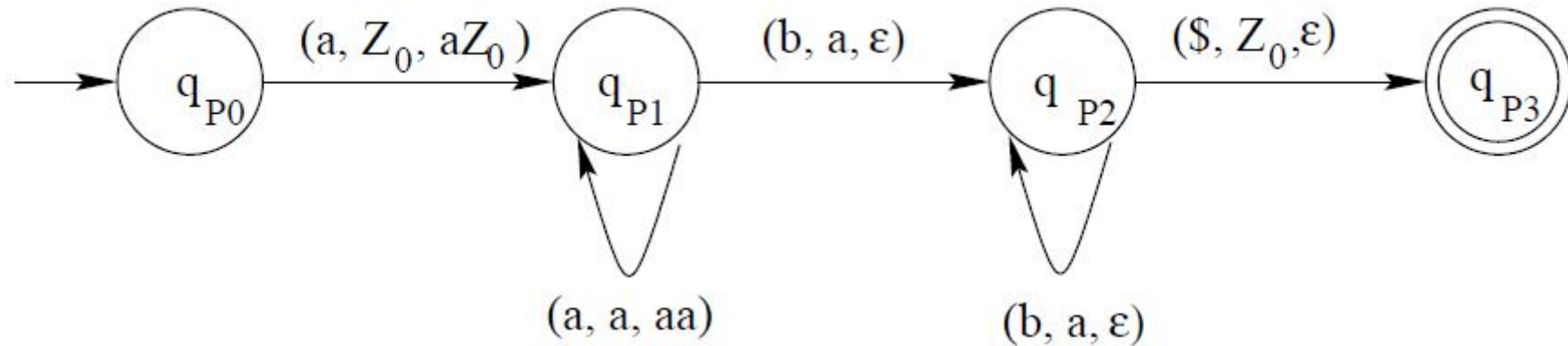
Autómata finito determinista **A**

$$L(A) = R$$



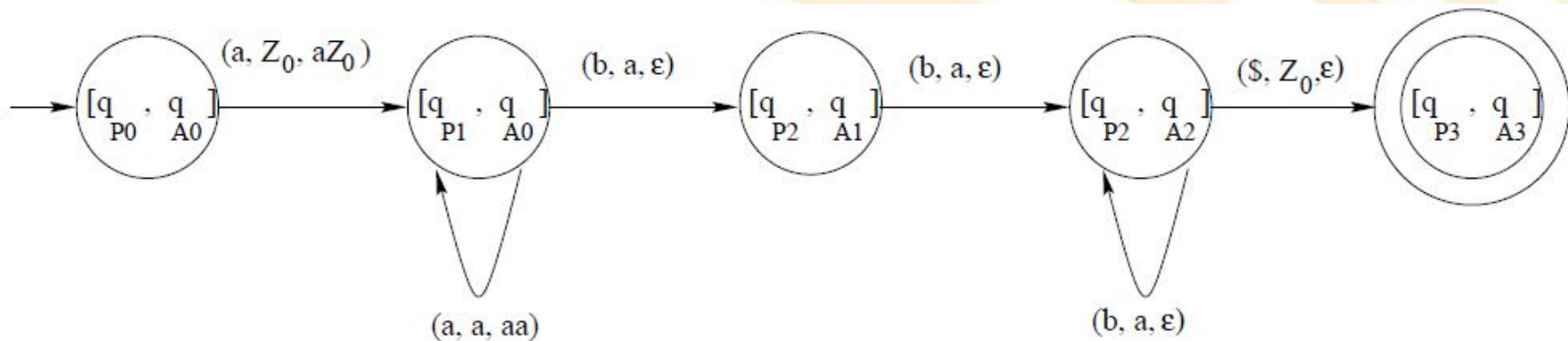
Autómata con pila P

$$F(P) = L$$



Autómata con pila P'

$$F(P') = F(P) \cap L(A) = L \cap R$$



Homomorfismo inverso

La clase de los lenguajes de contexto libre es **cerrada** respecto de los homomorfismos inversos.

Si

- **L** es un lenguaje de contexto libre y
- **h** es un homomorfismo

entonces

- **$h^{-1}(L) = \{ x \mid h(x) \in L \}$**

es un lenguaje de contexto libre

Homomorfismo inverso

Demostración

Sean

- L es un lenguaje de contexto libre
- $L = F(P)$, donde $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$
- $h : \Theta \rightarrow \Sigma^*$, donde $h(\theta) = w \in \Sigma^*$

Homomorfismo inverso

Demostración

Se va a construir $P' = (Q', \Theta, \Gamma, \delta', q'_0, Z_0, F')$

- $Q' = \{[q, x] \mid q \in Q$
 $\wedge \exists \theta \in \Theta (x \in \Sigma^* \text{ es un sufijo de } h(\theta))\}$

○ x se denomina buffer

- $q'_0 = [q_0, \varepsilon]$

- $F' = F \times \{\varepsilon\}$

verificándose que $F(P') = h^{-1}(L)$

Homomorfismo inverso

Demostración

- Función de transición δ

1. Se simulan las **transiciones** ε sin modificar el *buffer*

Si $(q', \alpha) \in \delta(q, \varepsilon, Z)$ entonces $([q', x], \alpha) \in \delta'([q, x] \varepsilon, Z)$

2. Se simulan los **movimientos sobre la cinta de lectura del autómata original P** con transiciones ε y eliminando símbolos del *buffer*

Si $(q', \alpha) \in \delta(q, \sigma, Z)$ entonces $([q', x], \alpha) \in \delta'([q, \sigma x] \varepsilon, Z)$

3. Se simulan los **movimientos sobre la cinta de lectura del nuevo autómata P'** cargando el *buffer*

$\forall q \in Q \forall \theta \in \Theta \forall \gamma \in \Gamma ([q, h(\theta)], \gamma) \in \delta'([q, \varepsilon], \theta, \gamma)$

Homomorfismo inverso

Ejemplo

- $\Theta = \{0, 1, \$\}$, $\Sigma = \{a, b, \$\}$
- $h : \Theta \rightarrow \Sigma^*$
 - $h(0) = a a$
 - $h(1) = b$
 - $h(\$) = \$$
- $L = \{a^n b^n \$ \mid n > 0\}$
- Se desea construir P' que reconozca a $h^{-1}(L)$

Homomorfismo inverso

Ejemplo

• $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, $F(P) = L$

○ $Q = \{q_0, q_1, q_2, q_3\}$

○ $\Sigma = \{a, b, \$\}$

○ $\Gamma = \{a, \$\}$

○ $F = \{q_3\}$

(1) $\delta(q_0, a, Z_0) = \{(q_1, aZ_0)\}$

(2) $\delta(q_1, a, a) = \{(q_1, aa)\}$

(3) $\delta(q_1, b, a) = \{(q_2, \varepsilon)\}$

(4) $\delta(q_2, b, a) = \{(q_2, \varepsilon)\}$

(5) $\delta(q_2, \$, Z_0) = \{(q_3, \varepsilon)\}$

Homomorfismo inverso

Ejemplo

- $P' = (Q', \Theta, \Gamma, \delta', q'_0, Z_0, F')$
 - $Q' = Q \times \{\varepsilon, a, a a, b, \$\}$
 - $F = \{[q_3, \varepsilon]\}$
 - $\Theta = \{0, 1, \$\}$
 - $\Gamma = \{a, Z_0\}$
 - ...

Homomorfismo inverso

Ejemplo

- Función de transición δ'

1. **No** se aplica la **regla número 1** porque no hay transiciones ε

2. Aplicación de la **regla número 2**

- La transición (1) $\delta(q_0, a, Z_0) = \{(q_1, aZ_0)\}$ genera

- $\delta'([q_0, a], \varepsilon, Z_0) = \{([q_1, \varepsilon], aZ_0)\}$

- $\delta'([q_0, aa], \varepsilon, Z_0) = \{([q_1, a], aZ_0)\}$

- La transición (2) $\delta(q_1, a, a) = \{(q_1, aa)\}$ genera

- $\delta'([q_1, a], \varepsilon, a) = \{([q_1, \varepsilon], aa)\}$

- $\delta'([q_1, aa], \varepsilon, a) = \{([q_1, a], aa)\}$

- ...

Homomorfismo inverso

Ejemplo

- Función de transición δ'
 2. Aplicación de **la regla número 2**
 - La transición (3) $\delta(q_1, b, a) = \{(q_2, \varepsilon)\}$ genera
 - $\delta'([q_1, b], \varepsilon, a) = \{([q_2, \varepsilon], \varepsilon)\}$
 - La transición (4) $\delta(q_2, b, a) = \{(q_2, \varepsilon)\}$ genera
 - $\delta'([q_2, b], \varepsilon, a) = \{([q_2, \varepsilon], \varepsilon)\}$
 - La transición (5) $\delta(q_2, \$, Z_0) = \{(q_3, \varepsilon)\}$ genera
 - $\delta'([q_2, \$], \varepsilon, Z_0) = \{([q_3, \varepsilon], \varepsilon)\}$

Homomorfismo inverso

Ejemplo

- Función de transición δ'
 3. Aplicación de **la regla número 3**
 - Transiciones para cargar el **buffer** desde $[q_0, \varepsilon]$
 - $\delta'([q_0, \varepsilon], 0, Z_0) = \{[q_0, aa], Z_0\}$
 - $\delta'([q_0, \varepsilon], 0, a) = \{[q_0, aa], a\}$
 - $\delta'([q_0, \varepsilon], 1, Z_0) = \{[q_0, b], Z_0\}$
 - $\delta'([q_0, \varepsilon], 1, a) = \{[q_0, b], a\}$
 - ...

Homomorfismo inverso

Ejemplo

- Función de transición δ'
 3. Aplicación de **la regla número 3**
 - Transiciones para cargar el **buffer** desde $[q_1, \varepsilon]$
 - $\delta'([q_1, \varepsilon], 0, Z_0) = \{[q_1, aa], Z_0\}$
 - $\delta'([q_1, \varepsilon], 0, a) = \{[q_1, aa], a\}$
 - $\delta'([q_1, \varepsilon], 1, Z_0) = \{[q_1, b], Z_0\}$
 - $\delta'([q_1, \varepsilon], 1, a) = \{[q_1, b], a\}$
 - ...

Homomorfismo inverso

Ejemplo

- Función de transición δ'
 3. Aplicación de **la regla número 3**
 - Transiciones para cargar el **buffer** desde $[q_2, \varepsilon]$
 - $\delta'([q_2, \varepsilon], 0, Z_0) = \{[q_2, aa], Z_0\}$
 - $\delta'([q_2, \varepsilon], 0, a) = \{[q_2, aa], a\}$
 - $\delta'([q_2, \varepsilon], 1, Z_0) = \{[q_2, b], Z_0\}$
 - $\delta'([q_2, \varepsilon], 1, a) = \{[q_2, b], a\}$
 - ...

Homomorfismo inverso

Ejemplo

- Función de transición δ'
 3. Aplicación de **la regla número 3**
 - Transiciones para cargar el **buffer** desde $[q_3, \varepsilon]$
 - $\delta'([q_3, \varepsilon], 0, Z_0) = \{[q_3, aa], Z_0\}$
 - $\delta'([q_3, \varepsilon], 0, a) = \{[q_3, aa], a\}$
 - $\delta'([q_3, \varepsilon], 1, Z_0) = \{[q_3, b], Z_0\}$
 - $\delta'([q_3, \varepsilon], 1, a) = \{[q_3, b], a\}$
 - ...

Homomorfismo inverso

Ejemplo

- Reconocimiento de $\mathbf{h^{-1}(L)} = \{0^n 1^{2n} \$ \mid n > 0\}$

$$([q_0, \varepsilon], 011$, Z_0) \mid - ([q_0, aa], 11$, Z_0) = ([q_0, aa], \varepsilon 11$, Z_0)$$

$$\mid - ([q_1, a], \varepsilon 11$, aZ_0)$$

$$\mid - ([q_1, \varepsilon], 11$, aaZ_0)$$

$$\mid - ([q_1, b], 1$, aaZ_0) = ([q_1, b], \varepsilon 1$, aaZ_0)$$

$$\mid - ([q_2, \varepsilon], 1$, aZ_0)$$

$$\mid - ([q_2, b], $, aZ_0) = ([q_2, b], \varepsilon $, aZ_0)$$

$$\mid - ([q_2, \varepsilon], $, Z_0)$$

$$\mid - ([q_2, $], \varepsilon, Z_0)$$

$$\mid - ([q_3, \varepsilon], \varepsilon, \varepsilon)$$

Algoritmo CYK

- **Entrada**

- $G = (V_N, V_T, P, S)$ está en la **FNC**
- $x \in V_T^*$

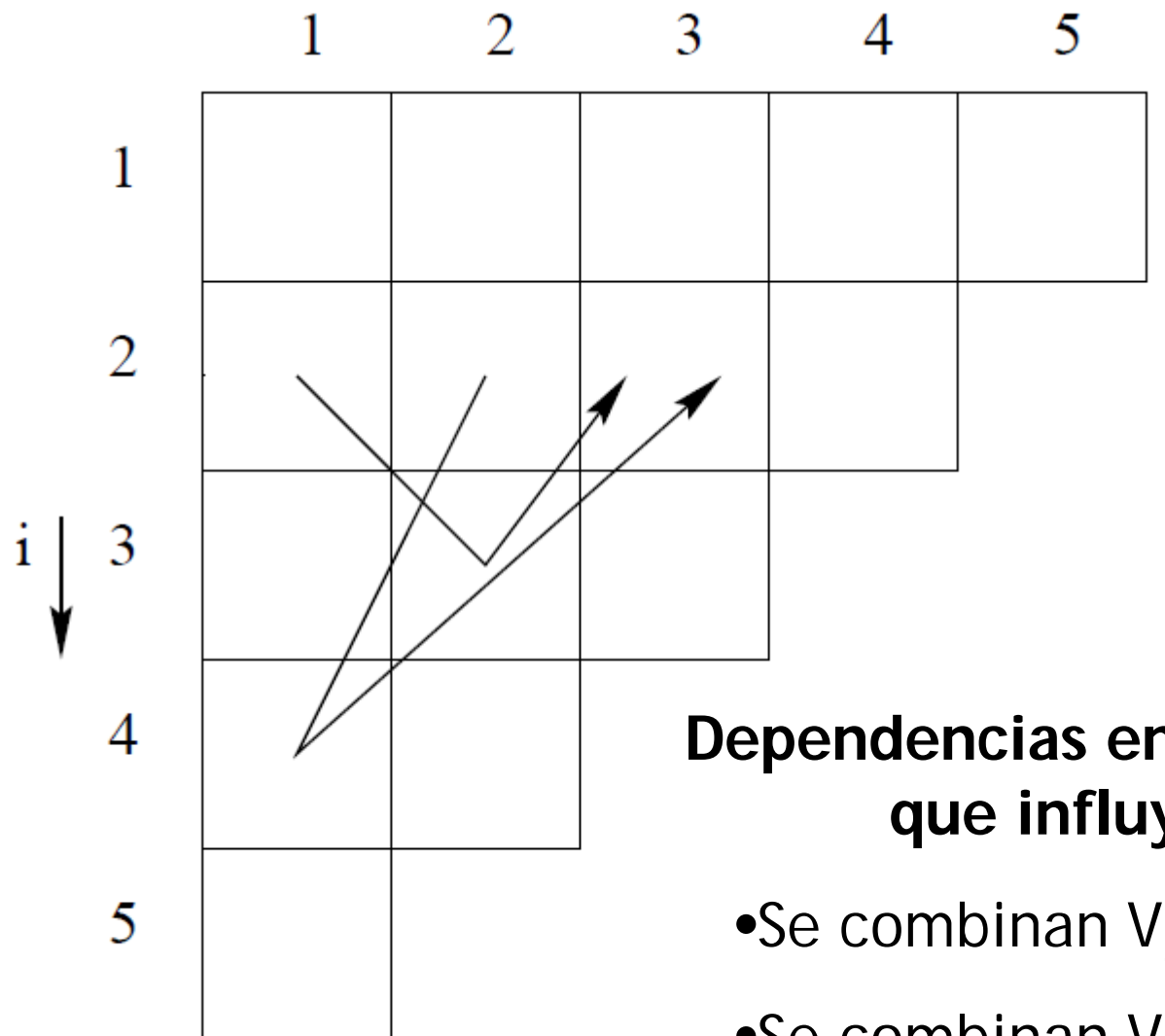
- **Salida**

- Verdadero si $x \in L(G)$
- Falso si $x \notin L(G)$

Algoritmo CYK

```
[1] inicio
[2]    $n \leftarrow |x|$ 
[3]   para  $i$  de 1 a  $n$  hacer
[4]      $V_{i,1} \leftarrow \{A \mid A \rightarrow \sigma \in P \wedge \sigma \text{ es el } i\text{-ésimo símbolo de } x\}$ 
[5]   fin para
[6]   para  $j$  de 2 a  $n$  hacer
[7]     para  $i$  de 1 a  $n - j + 1$  hacer
[8]        $V_{i,j} \leftarrow \emptyset$ 
[9]       para  $k$  de 1 a  $j - 1$  hacer
[10]         $V_{i,j} \leftarrow V_{i,j} \cup \{A \mid A \rightarrow BC \in P \wedge B \in V_{i,k} \wedge C \in V_{i+k, j-k}\}$ 
[11]      fin para
[12]    fin para
[13]  fin para
[14]  si  $S \in V_{1,n}$ 
[15]    entonces  $x \in L(G)$ 
[16]    si no  $x \notin L(G)$ 
[17]  fin si
[18] fin
```

Algoritmo CYK



**Dependencias entre los conjuntos
que influyen en $V_{2,3}$**

- Se combinan $V_{2,1}$ y $V_{3,2}$
- Se combinan $V_{2,2}$ y $V_{4,1}$

Algoritmo CYK

Ejemplo

$P = \{$
 $S \rightarrow A B$
 $S \rightarrow B C$
 $A \rightarrow B A$
 $A \rightarrow a$
 $B \rightarrow C C$
 $B \rightarrow b$
 $C \rightarrow A B$
 $C \rightarrow a$
 $\}$

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1					
a	2					
a	3					
b	4					
a	5					

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}				
a	2	{A,C}				
a	3	{A,C}				
b	4	{B}				
a	5	{A,C}				

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}	{A}			
a	2	{A,C}				
a	3	{A,C}				
b	4	{B}				
a	5	{A,C}				

$A \rightarrow BA \in P$

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}	{A, S}			
a	2	{A, C}				
a	3	{A, C}				
b	4	{B}				
a	5	{A, C}				

$$S \rightarrow BC \in P$$

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}	{A, S}			
a	2	{A, C}	{B}			
a	3	{A, C}	{S, C}			
b	4	{B}	{A, S}			
a	5	{A, C}				

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}	{A, S}	\emptyset		
a	2	{A, C}	{B}	{B}		
a	3	{A, C}	{S, C}	{B}		
b	4	{B}	{A, S}			
a	5	{A, C}				

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}	{A, S}	\emptyset		
a	2	{A, C}	{B}	{B}		
a	3	{A, C}	{S, C}	{B}		
b	4	{B}	{A, S}			
a	5	{A, C}				

$B \rightarrow C C \in P$

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}	{A, S}	\emptyset	\emptyset	
a	2	{A, C}	{B}	{B}	{S, C, A}	
a	3	{A, C}	{S, C}	{B}		
b	4	{B}	{A, S}			
a	5	{A, C}				

$S \rightarrow AB \in P$

$C \rightarrow AB \in P$

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}	{A, S}	\emptyset	\emptyset	
a	2	{A, C}	{B}	{B}	{S, C, A}	
a	3	{A, C}	{S, C}	{B}		
b	4	{B}	{A, S}			
a	5	{A, C}				

$A \rightarrow BA \in P$

Algoritmo CYK

Ejemplo

- Análisis de $X = \mathbf{b a a b a}$

	i, j	1	2	3	4	5
b	1	{B}	{A, S}	\emptyset	\emptyset	{S, C, A}
a	2	{A, C}	{B}	{B}	{S, C, A}	
a	3	{A, C}	{S, C}	{B}		
b	4	{B}	{A, S}			
a	5	{A, C}				

$$S \in V_{1,5}$$



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR
DEPARTAMENTO DE INFORMÁTICA Y ANÁLISIS NUMÉRICO

INGENIERÍA TÉCNICA EN INFORMÁTICA DE SISTEMAS

SEGUNDO CURSO, SEGUNDO CUATRIMESTRE



TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

Tema 10.- Lenguajes de Contexto Libre

