

Autómatas y Lenguajes Formales 2016-1

Maestría en Ciencia e Ingeniería de la Computación UNAM
Tema 12: Homomorfismos, lenguajes de Dyck y el Teorema de Chomsky-Schuetzenberger

Dr. Favio Ezequiel Miranda Perea

favio@ciencias.unam.mx

Facultad de Ciencias UNAM¹

3 de febrero de 2020

¹Con el apoyo del proyecto PAPIME PE102117



Homomorfismos de lenguajes

- Sean Σ, Δ dos alfabetos. Una función $h : \Sigma^* \rightarrow \Delta^*$ es un homomorfismo si y sólo si



Homomorfismos de lenguajes

- Sean Σ, Δ dos alfabetos. Una función $h : \Sigma^* \rightarrow \Delta^*$ es un homomorfismo si y sólo si
 - ▶ $h(\varepsilon) = \varepsilon$



Homomorfismos de lenguajes

- Sean Σ, Δ dos alfabetos. Una función $h : \Sigma^* \rightarrow \Delta^*$ es un homomorfismo si y sólo si
 - ▶ $h(\varepsilon) = \varepsilon$
 - ▶ $h(xy) = h(x)h(y)$



Homomorfismos de lenguajes

- Sean Σ, Δ dos alfabetos. Una función $h : \Sigma^* \rightarrow \Delta^*$ es un homomorfismo si y sólo si
 - ▶ $h(\varepsilon) = \varepsilon$
 - ▶ $h(xy) = h(x)h(y)$
- Un homomorfismo h queda determinado de manera única por sus valores en Σ .



Homomorfismos de lenguajes

- Sean Σ, Δ dos alfabetos. Una función $h : \Sigma^* \rightarrow \Delta^*$ es un homomorfismo si y sólo si
 - ▶ $h(\varepsilon) = \varepsilon$
 - ▶ $h(xy) = h(x)h(y)$
- Un homomorfismo h queda determinado de manera única por sus valores en Σ .
- Más aún, cualquier función $f : \Sigma \rightarrow \Delta^*$ puede extenderse de manera única a un homomorfismo $h : \Sigma^* \rightarrow \Delta^*$ tal que si $x \in \Sigma$ entonces $h(x) = f(x)$. Este homomorfismo se define como:

$$h(s_1 s_2 \dots s_n) =_{\text{def}} f(s_1)f(s_2)\dots f(s_n)$$

para cada $s_i \in \Sigma$



Imagen e imagen inversa de una función

Dados una función $f : A \rightarrow B$, $X \subseteq A$ y $Y \subseteq B$, definimos:

- La imagen de X bajo f , denotada $f[X]$, como

$$f[X] = \{f(x) \mid x \in X\}$$

Imagen e imagen inversa de una función

Dados una función $f : A \rightarrow B$, $X \subseteq A$ y $Y \subseteq B$, definimos:

- La imagen de X bajo f , denotada $f[X]$, como

$$f[X] = \{f(x) \mid x \in X\}$$

- La imagen inversa de Y bajo f , denotada $f^{-1}[Y]$, como

$$f^{-1}[Y] = \{x \mid f(x) \in Y\}$$



Imagen e imagen inversa de una función

Dados una función $f : A \rightarrow B$, $X \subseteq A$ y $Y \subseteq B$, definimos:

- La imagen de X bajo f , denotada $f[X]$, como

$$f[X] = \{f(x) \mid x \in X\}$$

- La imagen inversa de Y bajo f , denotada $f^{-1}[Y]$, como

$$f^{-1}[Y] = \{x \mid f(x) \in Y\}$$

- Obsérvese que $f[X] \subseteq B$ y $f^{-1}[Y] \subseteq A$



Imagen e imagen inversa de una función

Dados una función $f : A \rightarrow B$, $X \subseteq A$ y $Y \subseteq B$, definimos:

- La imagen de X bajo f , denotada $f[X]$, como

$$f[X] = \{f(x) \mid x \in X\}$$

- La imagen inversa de Y bajo f , denotada $f^{-1}[Y]$, como

$$f^{-1}[Y] = \{x \mid f(x) \in Y\}$$

- Obsérvese que $f[X] \subseteq B$ y $f^{-1}[Y] \subseteq A$
- Propiedad relevante: $f[X \cup Z] = f[X] \cup f[Z]$



Propiedades de cerradura

Homomorfismos de lenguajes

Sean $h : \Sigma^* \rightarrow \Delta^*$, $L \subseteq \Sigma^*$, $S \subseteq \Delta^*$.

- Si L es regular entonces $h[L]$ es regular. Es decir, la imagen de un lenguaje regular bajo un homomorfismo es regular.



Propiedades de cerradura

Homomorfismos de lenguajes

Sean $h : \Sigma^* \rightarrow \Delta^*$, $L \subseteq \Sigma^*$, $S \subseteq \Delta^*$.

- Si L es regular entonces $h[L]$ es regular. Es decir, la imagen de un lenguaje regular bajo un homomorfismo es regular.
- Si S es regular entonces $h^{-1}[S]$ es regular. Es decir, la imagen inversa de un lenguaje regular bajo un homomorfismo es regular.



Propiedades de cerradura

Homomorfismos de lenguajes

Sean $h : \Sigma^* \rightarrow \Delta^*$, $L \subseteq \Sigma^*$, $S \subseteq \Delta^*$.

- Si L es regular entonces $h[L]$ es regular. Es decir, la imagen de un lenguaje regular bajo un homomorfismo es regular.
- Si S es regular entonces $h^{-1}[S]$ es regular. Es decir, la imagen inversa de un lenguaje regular bajo un homomorfismo es regular.
- Si L es libre de contexto entonces $h[L]$ es libre de contexto. Es decir, la imagen de un lenguaje libre de contexto bajo un homomorfismo es libre de contexto.



Propiedades de cerradura

Homomorfismos de lenguajes

Sean $h : \Sigma^* \rightarrow \Delta^*$, $L \subseteq \Sigma^*$, $S \subseteq \Delta^*$.

- Si L es regular entonces $h[L]$ es regular. Es decir, la imagen de un lenguaje regular bajo un homomorfismo es regular.
- Si S es regular entonces $h^{-1}[S]$ es regular. Es decir, la imagen inversa de un lenguaje regular bajo un homomorfismo es regular.
- Si L es libre de contexto entonces $h[L]$ es libre de contexto. Es decir, la imagen de un lenguaje libre de contexto bajo un homomorfismo es libre de contexto.
- Propiedad de cerradura importante: Si R es regular y L es libre de contexto entonces $R \cap L$ es libre de contexto.



Homomorfismos de borrado

- Sean Σ, B dos alfabetos ajenos, es decir, $\Sigma \cap B = \emptyset$.



Homomorfismos de borrado

- Sean Σ, B dos alfabetos ajenos, es decir, $\Sigma \cap B = \emptyset$.
- Definimos $f : \Sigma \cup B \rightarrow \Sigma^*$ como

$$f(s) = \sigma \quad \text{si } s \in \Sigma$$

$$f(s) = \varepsilon \quad \text{si } s \in B$$



Homomorfismos de borrado

- Sean Σ, B dos alfabetos ajenos, es decir, $\Sigma \cap B = \emptyset$.
- Definimos $f : \Sigma \cup B \rightarrow \Sigma^*$ como

$$f(s) = \sigma \quad \text{si } s \in \Sigma$$

$$f(s) = \varepsilon \quad \text{si } s \in B$$

- f induce un único homomorfismo $h : (\Sigma \cup B)^* \rightarrow \Sigma^*$ de tal forma que $h(x) = y$ si y sólo si y se obtiene a partir de x al borrar todos los símbolos de B .



Homomorfismos de borrado

- Sean Σ, B dos alfabetos ajenos, es decir, $\Sigma \cap B = \emptyset$.
- Definimos $f : \Sigma \cup B \rightarrow \Sigma^*$ como

$$f(s) = \sigma \quad \text{si } s \in \Sigma$$

$$f(s) = \varepsilon \quad \text{si } s \in B$$

- f induce un único homomorfismo $h : (\Sigma \cup B)^* \rightarrow \Sigma^*$ de tal forma que $h(x) = y$ si y sólo si y se obtiene a partir de x al borrar todos los símbolos de B .
- A este homomorfismo lo denotamos con $erase_B$



Homomorfismos de borrado

- Sean Σ, B dos alfabetos ajenos, es decir, $\Sigma \cap B = \emptyset$.
- Definimos $f : \Sigma \cup B \rightarrow \Sigma^*$ como

$$f(s) = \sigma \quad \text{si } s \in \Sigma$$

$$f(s) = \varepsilon \quad \text{si } s \in B$$

- f induce un único homomorfismo $h : (\Sigma \cup B)^* \rightarrow \Sigma^*$ de tal forma que $h(x) = y$ si y sólo si y se obtiene a partir de x al borrar todos los símbolos de B .
- A este homomorfismo lo denotamos con $erase_B$
- Ejemplo: Si $\Sigma = \{0, 1\}$ y $B = \{(,)\}$ entonces $erase_B$ es el homomorfismo que borra paréntesis, por ejemplo:

$$erase_B(00(11)(0)(01)) = 0011001$$



Gramática de borrado a partir de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto y $R \subseteq T$



Gramática de borrado a partir de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto y $R \subseteq T$
- Definimos la gramática que borra a (símbolos de) R en G , denotada $Er_R(G)$, como

$$Er_R(G) = \langle V, T - R, S, \mathcal{P}' \rangle$$

Gramática de borrado a partir de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto y $R \subseteq T$
- Definimos la gramática que borra a (símbolos de) R en G , denotada $Er_R(G)$, como

$$Er_R(G) = \langle V, T - R, S, \mathcal{P}' \rangle$$

- Donde \mathcal{P}' se define como sigue:

Gramática de borrado a partir de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto y $R \subseteq T$
- Definimos la gramática que borra a (símbolos de) R en G , denotada $Er_R(G)$, como

$$Er_R(G) = \langle V, T - R, S, \mathcal{P}' \rangle$$

- Donde \mathcal{P}' se define como sigue:
 - ▶ Si $X \rightarrow v$ es una producción en \mathcal{P} , entonces $X \rightarrow \text{erase}_R(v)$ es una producción de \mathcal{P}' .



Gramática de borrado a partir de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto y $R \subseteq T$
- Definimos la gramática que borra a (símbolos de) R en G , denotada $Er_R(G)$, como

$$Er_R(G) = \langle V, T - R, S, \mathcal{P}' \rangle$$

- Donde \mathcal{P}' se define como sigue:
 - ▶ Si $X \rightarrow v$ es una producción en \mathcal{P} , entonces $X \rightarrow \text{erase}_R(v)$ es una producción de \mathcal{P}' .
- Si G es una gramática libre de contexto y $R \subseteq T$, entonces

$$L(Er_R(G)) = \text{erase}_R[L(G)]$$



Gramática de borrado a partir de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto y $R \subseteq T$
- Definimos la gramática que borra a (símbolos de) R en G , denotada $Er_R(G)$, como

$$Er_R(G) = \langle V, T - R, S, \mathcal{P}' \rangle$$

- Donde \mathcal{P}' se define como sigue:
 - ▶ Si $X \rightarrow v$ es una producción en \mathcal{P} , entonces $X \rightarrow \text{erase}_R(v)$ es una producción de \mathcal{P}' .
- Si G es una gramática libre de contexto y $R \subseteq T$, entonces

$$L(Er_R(G)) = \text{erase}_R[L(G)]$$

- Corolario: Si $L \subseteq \Sigma^*$ es libre de contexto y $R \subseteq \Sigma$ entonces $\text{erase}_R[L]$ es libre de contexto.



Gramática separadora de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto en forma normal de Chomsky.



Gramática separadora de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto en forma normal de Chomsky.
- Definimos la gramática separadora de G , denotada G_{sep} como

$$G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$$



Gramática separadora de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto en forma normal de Chomsky.
- Definimos la gramática separadora de G , denotada G_{sep} como

$$G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$$

- El conjunto de producciones \mathcal{P}' se define como sigue:



Gramática separadora de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto en forma normal de Chomsky.
- Definimos la gramática separadora de G , denotada G_{sep} como

$$G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$$

- El conjunto de producciones \mathcal{P}' se define como sigue:
 - ▶ Las producciones de \mathcal{P} de la forma $A \rightarrow a$, están en \mathcal{P}'



Gramática separadora de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto en forma normal de Chomsky.
- Definimos la gramática separadora de G , denotada G_{sep} como

$$G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$$

- El conjunto de producciones \mathcal{P}' se define como sigue:
 - ▶ Las producciones de \mathcal{P} de la forma $A \rightarrow a$, están en \mathcal{P}'
 - ▶ Para cada producción en \mathcal{P} de la forma $A_i \rightarrow B_i C_i$, \mathcal{P}' tiene la siguiente producción

$$A_i \rightarrow p_i B_i \bar{p}_i C_i$$



Gramática separadora de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto en forma normal de Chomsky.
- Definimos la gramática separadora de G , denotada G_{sep} como

$$G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$$

- El conjunto de producciones \mathcal{P}' se define como sigue:
 - ▶ Las producciones de \mathcal{P} de la forma $A \rightarrow a$, están en \mathcal{P}'
 - ▶ Para cada producción en \mathcal{P} de la forma $A_i \rightarrow B_i C_i$, \mathcal{P}' tiene la siguiente producción

$$A_i \rightarrow p_i B_i \bar{p}_i C_i$$

- Donde $B = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ es un conjunto de símbolos nuevos, i.e., $B \cap T = \emptyset$



Gramática separadora de G

- Sean $G = \langle V, T, S, \mathcal{P} \rangle$ una gramática libre de contexto en forma normal de Chomsky.
- Definimos la gramática separadora de G , denotada G_{sep} como

$$G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$$

- El conjunto de producciones \mathcal{P}' se define como sigue:
 - ▶ Las producciones de \mathcal{P} de la forma $A \rightarrow a$, están en \mathcal{P}'
 - ▶ Para cada producción en \mathcal{P} de la forma $A_i \rightarrow B_i C_i$, \mathcal{P}' tiene la siguiente producción

$$A_i \rightarrow p_i B_i \bar{p}_i C_i$$

- Donde $B = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ es un conjunto de símbolos nuevos, i.e., $B \cap T = \emptyset$
- B es esencialmente un conjunto de n juegos distintos de paréntesis.



Gramática separadora

Ejemplo

- Sea G dada por

$$S \rightarrow XY \quad S \rightarrow YX \quad Y \rightarrow ZZ \quad X \rightarrow a \quad Z \rightarrow a$$



Gramática separadora

Ejemplo

- Sea G dada por

$$S \rightarrow XY \quad S \rightarrow YX \quad Y \rightarrow ZZ \quad X \rightarrow a \quad Z \rightarrow a$$

- G es ambigua:

$$S \rightarrow XY \rightarrow aY \rightarrow aZZ \rightarrow aaZ \rightarrow aaa$$

$$S \rightarrow YX \rightarrow ZXZ \rightarrow aZX \rightarrow aaX \rightarrow aaa$$

Gramática separadora

Ejemplo

- Sea G dada por

$$S \rightarrow XY \quad S \rightarrow YX \quad Y \rightarrow ZZ \quad X \rightarrow a \quad Z \rightarrow a$$

- G es ambigua:

$$S \rightarrow XY \rightarrow aY \rightarrow aZZ \rightarrow aaZ \rightarrow aaa$$

$$S \rightarrow YX \rightarrow ZXZ \rightarrow aZX \rightarrow aaX \rightarrow aaa$$

- G_{sep} está dada por

$$S \rightarrow (X)Y \quad S \rightarrow [Y]X \quad Y \rightarrow \{Z\}Z \quad X \rightarrow a \quad Z \rightarrow a$$



Gramática separadora

Ejemplo

- Sea G dada por

$$S \rightarrow XY \quad S \rightarrow YX \quad Y \rightarrow ZZ \quad X \rightarrow a \quad Z \rightarrow a$$

- G es ambigua:

$$S \rightarrow XY \rightarrow aY \rightarrow aZZ \rightarrow aaZ \rightarrow aaa$$

$$S \rightarrow YX \rightarrow ZXZ \rightarrow aZX \rightarrow aaX \rightarrow aaa$$

- G_{sep} está dada por

$$S \rightarrow (X)Y \quad S \rightarrow [Y]X \quad Y \rightarrow \{Z\}Z \quad X \rightarrow a \quad Z \rightarrow a$$

- G_{sep} separa y codifica las dos derivaciones en G :

$$S \rightarrow (X)Y \rightarrow (a)Y \rightarrow a\{Z\}Z \rightarrow (a)\{a\}Z \rightarrow (a)\{a\}a$$

$$S \rightarrow [Y]X \rightarrow [\{Z\}Z]X \rightarrow [\{a\}Z]X \rightarrow [\{a\}a]X \rightarrow [\{a\}a]a$$



Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.



Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo



Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
 - ▶ $B_1 = \{(,)\}$



Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
 - ▶ $B_1 = \{(,)\}$
 - ▶ $B_2 = \{(,), [,]\}$

Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
 - ▶ $B_1 = \{(,)\}$
 - ▶ $B_2 = \{(,), [,]\}$
 - ▶ $B_3 = \{(,), [,], \{, \}\}$



Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
 - ▶ $B_1 = \{(,)\}$
 - ▶ $B_2 = \{(,), [,]\}$
 - ▶ $B_3 = \{(,), [,], \{ , \}\}$
- Con $D_n(A)$ denotamos al lenguaje de cadenas de $A \cup B_n$ generadas por la siguiente gramática libre de contexto

Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
 - ▶ $B_1 = \{(,)\}$
 - ▶ $B_2 = \{(,), [,]\}$
 - ▶ $B_3 = \{(,), [,], \{, \}\}$
- Con $D_n(A)$ denotamos al lenguaje de cadenas de $A \cup B_n$ generadas por la siguiente gramática libre de contexto
 - ▶ Para toda $a \in A$, $S \rightarrow a$



Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
 - ▶ $B_1 = \{(,)\}$
 - ▶ $B_2 = \{(,), [,]\}$
 - ▶ $B_3 = \{(,), [,], \{, \}\}$
- Con $D_n(A)$ denotamos al lenguaje de cadenas de $A \cup B_n$ generadas por la siguiente gramática libre de contexto
 - ▶ Para toda $a \in A$, $S \rightarrow a$
 - ▶ Para toda $p_i \in P_n$, $S \rightarrow p_i S \bar{p}_i$



Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
 - ▶ $B_1 = \{(,)\}$
 - ▶ $B_2 = \{(,), [,]\}$
 - ▶ $B_3 = \{(,), [,], \{, \}\}$
- Con $D_n(A)$ denotamos al lenguaje de cadenas de $A \cup B_n$ generadas por la siguiente gramática libre de contexto
 - ▶ Para toda $a \in A$, $S \rightarrow a$
 - ▶ Para toda $p_i \in P_n$, $S \rightarrow p_i S \bar{p}_i$
 - ▶ $S \rightarrow SS$



Lenguajes de paréntesis o lenguajes de Dyck

- Sean A un conjunto finito y $B_n = \{p_1, \bar{p}_1, \dots, p_n, \bar{p}_n\}$ tal que $A \cap B_n = \emptyset$.
- B_n puede pensarse como un alfabeto con n juegos de paréntesis distintos. Por ejemplo
 - ▶ $B_1 = \{(,)\}$
 - ▶ $B_2 = \{(,), [,]\}$
 - ▶ $B_3 = \{(,), [,], \{, \}\}$
- Con $D_n(A)$ denotamos al lenguaje de cadenas de $A \cup B_n$ generadas por la siguiente gramática libre de contexto
 - ▶ Para toda $a \in A$, $S \rightarrow a$
 - ▶ Para toda $p_i \in P_n$, $S \rightarrow p_i S \bar{p}_i$
 - ▶ $S \rightarrow SS$
 - ▶ $S \rightarrow \varepsilon$



Lenguajes de Dyck

Propiedades

- $D_n(A)$ se llama el lenguaje (generalizado) de Dyck de orden n .



Lenguajes de Dyck

Propiedades

- $D_n(A)$ se llama el lenguaje (generalizado) de Dyck de orden n .
- $D_n(\emptyset)$ se llama el lenguaje (simple) de Dyck de orden n , denotado simplemente D_n .

Lenguajes de Dyck

Propiedades

- $D_n(A)$ se llama el lenguaje (generalizado) de Dyck de orden n .
- $D_n(\emptyset)$ se llama el lenguaje (simple) de Dyck de orden n , denotado simplemente D_n .
- Los lenguajes de Dyck tienen las siguientes propiedades:

Lenguajes de Dyck

Propiedades

- $D_n(A)$ se llama el lenguaje (generalizado) de Dyck de orden n .
- $D_n(\emptyset)$ se llama el lenguaje (simple) de Dyck de orden n , denotado simplemente D_n .
- Los lenguajes de Dyck tienen las siguientes propiedades:
 - ▶ $A^* \subseteq D_n(A)$

Lenguajes de Dyck

Propiedades

- $D_n(A)$ se llama el lenguaje (generalizado) de Dyck de orden n .
- $D_n(\emptyset)$ se llama el lenguaje (simple) de Dyck de orden n , denotado simplemente D_n .
- Los lenguajes de Dyck tienen las siguientes propiedades:
 - ▶ $A^* \subseteq D_n(A)$
 - ▶ Si $x, y \in D_n(A)$ entonces $xy \in D_n(A)$

Lenguajes de Dyck

Propiedades

- $D_n(A)$ se llama el lenguaje (generalizado) de Dyck de orden n .
- $D_n(\emptyset)$ se llama el lenguaje (simple) de Dyck de orden n , denotado simplemente D_n .
- Los lenguajes de Dyck tienen las siguientes propiedades:
 - ▶ $A^* \subseteq D_n(A)$
 - ▶ Si $x, y \in D_n(A)$ entonces $xy \in D_n(A)$
 - ▶ Si $x \in D_n(A)$ entonces $p_i x \bar{p}_i \in D_n(A)$

Lenguajes de Dyck

Propiedades

- $D_n(A)$ se llama el lenguaje (generalizado) de Dyck de orden n .
- $D_n(\emptyset)$ se llama el lenguaje (simple) de Dyck de orden n , denotado simplemente D_n .
- Los lenguajes de Dyck tienen las siguientes propiedades:
 - ▶ $A^* \subseteq D_n(A)$
 - ▶ Si $x, y \in D_n(A)$ entonces $xy \in D_n(A)$
 - ▶ Si $x \in D_n(A)$ entonces $p_i x \bar{p}_i \in D_n(A)$
 - ▶ Si $x \in D_n(A)$ y $x \notin A^*$ entonces existen $u \in A^*$, $v, w \in D_n(A)$, $i \leq n$ tales que $x = up_i v \bar{p}_i w$



La gramática *J*

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática *J* como sigue:



La gramática J

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática J como sigue:
- $J = \langle V, T \cup B, S, \mathcal{P}'' \rangle$ donde \mathcal{P}'' consta de las siguientes producciones:



La gramática J

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática J como sigue:
- $J = \langle V, T \cup B, S, \mathcal{P}'' \rangle$ donde \mathcal{P}'' consta de las siguientes producciones:
 - ▶ Todas las producciones $V \rightarrow a$ de \mathcal{P}'

La gramática J

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática J como sigue:
- $J = \langle V, T \cup B, S, \mathcal{P}'' \rangle$ donde \mathcal{P}'' consta de las siguientes producciones:
 - ▶ Todas las producciones $V \rightarrow a$ de \mathcal{P}'
 - ▶ Si $A_i \rightarrow p_i B_i \bar{p}_i C_i$ está en \mathcal{P}' entonces \mathcal{P}'' tiene a:



La gramática J

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática J como sigue:
- $J = \langle V, T \cup B, S, \mathcal{P}'' \rangle$ donde \mathcal{P}'' consta de las siguientes producciones:
 - ▶ Todas las producciones $V \rightarrow a$ de \mathcal{P}'
 - ▶ Si $A_i \rightarrow p_i B_i \bar{p}_i C_i$ está en \mathcal{P}' entonces \mathcal{P}'' tiene a:
 - ★ $A_i \rightarrow p_i B_i$



La gramática J

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática J como sigue:
- $J = \langle V, T \cup B, S, \mathcal{P}'' \rangle$ donde \mathcal{P}'' consta de las siguientes producciones:
 - ▶ Todas las producciones $V \rightarrow a$ de \mathcal{P}'
 - ▶ Si $A_i \rightarrow p_i B_i \bar{p}_i C_i$ está en \mathcal{P}' entonces \mathcal{P}'' tiene a:
 - ★ $A_i \rightarrow p_i B_i$
 - ★ $V \rightarrow a \bar{p}_i C_i$, para cada producción $V \rightarrow a$ que esté en \mathcal{P}'



La gramática J

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática J como sigue:
- $J = \langle V, T \cup B, S, \mathcal{P}'' \rangle$ donde \mathcal{P}'' consta de las siguientes producciones:
 - ▶ Todas las producciones $V \rightarrow a$ de \mathcal{P}'
 - ▶ Si $A_i \rightarrow p_i B_i \bar{p}_i C_i$ está en \mathcal{P}' entonces \mathcal{P}'' tiene a:
 - ★ $A_i \rightarrow p_i B_i$
 - ★ $V \rightarrow a \bar{p}_i C_i$, para cada producción $V \rightarrow a$ que esté en \mathcal{P}'
- Es claro que J es equivalente a una gramática lineal derecha.



La gramática J

- A partir de $G_{sep} = \langle V, T \cup B, S, \mathcal{P}' \rangle$ definimos ahora una gramática J como sigue:
- $J = \langle V, T \cup B, S, \mathcal{P}'' \rangle$ donde \mathcal{P}'' consta de las siguientes producciones:
 - ▶ Todas las producciones $V \rightarrow a$ de \mathcal{P}'
 - ▶ Si $A_i \rightarrow p_i B_i \bar{p}_i C_i$ está en \mathcal{P}' entonces \mathcal{P}'' tiene a:
 - ★ $A_i \rightarrow p_i B_i$
 - ★ $V \rightarrow a \bar{p}_i C_i$, para cada producción $V \rightarrow a$ que esté en \mathcal{P}'
- Es claro que J es equivalente a una gramática lineal derecha.
- Por lo tanto $L(J)$ es un lenguaje regular.



Gramática J

Ejemplo

- G_{sep} está dada por

$$S \rightarrow (X)Y \quad S \rightarrow [Y]X \quad Y \rightarrow \{Z\}Z \quad X \rightarrow a \quad Z \rightarrow a$$



Gramática J

Ejemplo

- G_{sep} está dada por

$$S \rightarrow (X)Y \quad S \rightarrow [Y]X \quad Y \rightarrow \{Z\}Z \quad X \rightarrow a \quad Z \rightarrow a$$

- J se construye como sigue:



Gramática J

Ejemplo

- G_{sep} está dada por

$$S \rightarrow (X)Y \quad S \rightarrow [Y]X \quad Y \rightarrow \{Z\}Z \quad X \rightarrow a \quad Z \rightarrow a$$

- J se construye como sigue:

- $S \rightarrow (X)Y$ genera a : $S \rightarrow (X \quad X \rightarrow a)Y \quad Z \rightarrow a)Y$

Gramática J

Ejemplo

- G_{sep} está dada por

$$S \rightarrow (X)Y \quad S \rightarrow [Y]X \quad Y \rightarrow \{Z\}Z \quad X \rightarrow a \quad Z \rightarrow a$$

- J se construye como sigue:

- ▶ $S \rightarrow (X)Y$ genera a : $S \rightarrow (X \quad X \rightarrow a)Y \quad Z \rightarrow a)Y$
- ▶ $S \rightarrow [Y]X$ genera a : $S \rightarrow [Y \quad X \rightarrow a]X \quad Z \rightarrow a]X$



Gramática J

Ejemplo

- G_{sep} está dada por

$$S \rightarrow (X)Y \quad S \rightarrow [Y]X \quad Y \rightarrow \{Z\}Z \quad X \rightarrow a \quad Z \rightarrow a$$

- J se construye como sigue:

- ▶ $S \rightarrow (X)Y$ genera a : $S \rightarrow (X \quad X \rightarrow a)Y \quad Z \rightarrow a)Y$
- ▶ $S \rightarrow [Y]X$ genera a : $S \rightarrow [Y \quad X \rightarrow a]X \quad Z \rightarrow a]X$
- ▶ $Y \rightarrow \{Z\}Z$ genera a : $Y \rightarrow \{Z \quad X \rightarrow a\}Z \quad Z \rightarrow a\}Z$

Gramática J

Ejemplo

- G_{sep} está dada por

$$S \rightarrow (X)Y \quad S \rightarrow [Y]X \quad Y \rightarrow \{Z\}Z \quad X \rightarrow a \quad Z \rightarrow a$$

- J se construye como sigue:

- ▶ $S \rightarrow (X)Y$ genera a : $S \rightarrow (X \quad X \rightarrow a)Y \quad Z \rightarrow a)Y$
- ▶ $S \rightarrow [Y]X$ genera a : $S \rightarrow [Y \quad X \rightarrow a]X \quad Z \rightarrow a]X$
- ▶ $Y \rightarrow \{Z\}Z$ genera a : $Y \rightarrow \{Z \quad X \rightarrow a\}Z \quad Z \rightarrow a\}Z$

- Así J es:

$$S \rightarrow (X \quad X \rightarrow a)Y \quad Z \rightarrow a)Y$$

$$S \rightarrow [Y \quad X \rightarrow a]X \quad Z \rightarrow a]X$$

$$Y \rightarrow \{Z \quad X \rightarrow a\}Z \quad Z \rightarrow a\}Z$$

$$X \rightarrow a \quad Z \rightarrow a$$

Propiedades de G_{sep}

P1 $\text{erase}_B[L(G_{sep})] = L(G)$



Propiedades de G_{sep}

P1 $\text{erase}_B[L(G_{sep})] = L(G)$

P2 $L(G_{sep}) \subseteq D_n(T)$

Propiedades de G_{sep}

P1 $\text{erase}_B[L(G_{sep})] = L(G)$

P2 $L(G_{sep}) \subseteq D_n(T)$

P3 $L(G_{sep}) \subseteq L(J)$

Propiedades de G_{sep}

P1 $\text{erase}_B[L(G_{sep})] = L(G)$

P2 $L(G_{sep}) \subseteq D_n(T)$

P3 $L(G_{sep}) \subseteq L(J)$

P4 $L(J) \cap D_n(T) \subseteq L(G_{sep})$

Propiedades de G_{sep}

P1 $\text{erase}_B[L(G_{sep})] = L(G)$

P2 $L(G_{sep}) \subseteq D_n(T)$

P3 $L(G_{sep}) \subseteq L(J)$

P4 $L(J) \cap D_n(T) \subseteq L(G_{sep})$

P5 Si G está en FNC entonces existe R regular tal que

$$L(G_{sep}) = R \cap D_n(T)$$

Teorema de Chomsky-Schuetzenberger

Sea $L \subseteq T^*$. Las siguientes condiciones son equivalentes:

- I) L es un lenguaje libre de contexto

Es decir, todo lenguaje libre de contexto es imagen homomórfica de la intersección de un lenguaje regular y un lenguaje de Dyck.



Teorema de Chomsky-Schuetzenberger

Sea $L \subseteq T^*$. Las siguientes condiciones son equivalentes:

- I) L es un lenguaje libre de contexto
- II) Existen un lenguaje regular R , un homomorfismo h y un número natural $n \in \mathbb{N}$ tales que

$$L = h[R \cap D_n(T)]$$

Es decir, todo lenguaje libre de contexto es imagen homomórfica de la intersección de un lenguaje regular y un lenguaje de Dyck.



Teorema de Chomsky-Schuetzenberger

Versión usual

- El teorema de Chomsky-Schuetzenberger suele enunciarse y demostrarse utilizando lenguajes de Dyck simples como sigue:



Teorema de Chomsky-Schuetzenberger

Versión usual

- El teorema de Chomsky-Schuetzenberger suele enunciarse y demostrarse utilizando lenguajes de Dyck simples como sigue:

Si L es un lenguaje libre de contexto entonces existen un lenguaje regular R , un homomorfismo h y un número natural $n > 0$ tales que:

$$L = h[R \cap D_n]$$



Teorema de Chomsky-Schuetzenberger

Versión usual

- El teorema de Chomsky-Schuetzenberger suele enunciarse y demostrarse utilizando lenguajes de Dyck simples como sigue:

Si L es un lenguaje libre de contexto entonces existen un lenguaje regular R , un homomorfismo h y un número natural $n > 0$ tales que:

$$L = h[R \cap D_n]$$

- Este teorema resulta equivalente a la versión aquí presentada.

