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 8.- Gramáticas de Contexto Libre

DIAN
DPTO. DE INFORMÁTICA
Y ANÁLISIS NUMÉRICO

Ejemplo de Gramática de Contexto Libre

$$\mathcal{G}_1 = (\mathcal{V}_\mathcal{N}, \mathcal{V}_\mathcal{T}, \mathcal{P}_1, \mathcal{S})$$

$$\mathcal{P}_1 = \{$$

$$(1)\ \mathcal{S} \longrightarrow \textit{identificador} = \mathcal{E}$$

$$(2)\ \mathcal{E} \longrightarrow \mathcal{E} + \mathcal{E}$$

$$(3)\ \mathcal{E} \longrightarrow \mathcal{E} * \mathcal{E}$$

$$(4)\ \mathcal{E} \longrightarrow (\ \mathcal{E}\ )$$

$$(5)\ \mathcal{E} \longrightarrow \textit{identificador}$$

$$(6)\ \mathcal{E} \longrightarrow \textit{número}$$

$$\}$$

Ejemplo de Gramática de Contexto Libre

$\mathcal{G}_2 = (\mathcal{V}_\mathcal{N}, \mathcal{V}_\mathcal{T}, \mathcal{P}_2, \mathcal{S})$

$\mathcal{P}_2 = \{$

$\mathcal{S} \rightarrow$ *mientras* $\mathcal{C}$ *hacer* $\mathcal{S}$ *fin mientras*

$\mathcal{S} \rightarrow$ *si* $\mathcal{C}$ *entonces* $\mathcal{S}$ *fin si*

$\mathcal{S} \rightarrow$ *si* $\mathcal{C}$ *entonces* $\mathcal{S}$ *si no* $\mathcal{S}$ *fin si*

$\mathcal{S} \rightarrow$ *Asignación*

$\mathcal{C} \rightarrow$ *Condición*

$\}$

Ejemplo de Derivación con la gramática G $_1$

$$S \Rightarrow_1 identificador = E$$

$$\Rightarrow_2 identificador = E + E$$

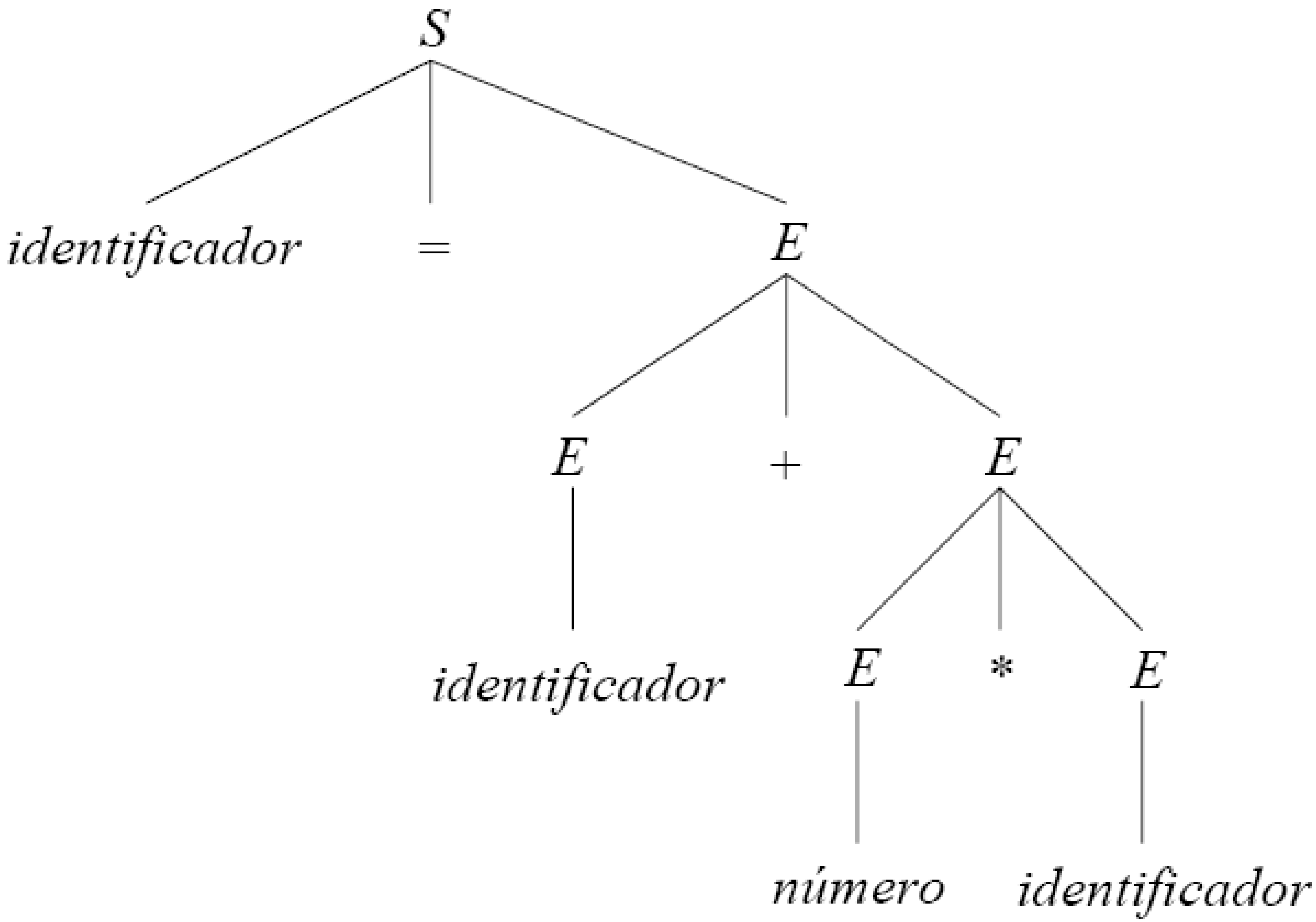
$$\Rightarrow_3 identificador = E + E * E$$

$$\Rightarrow_6 identificador = E + número * E$$

$$\Rightarrow_5 identificador = E + número * identificador$$

$$\Rightarrow_5 identificador = identificador + número * identificador$$

Árbol sintáctico de derivación



Lenguaje generado por una gramática de contexto libre

$$G_3 = (V_N, V_T, P_3, S)$$

$$P_3 = \{$$

$$(1)\ S \to a\,A\,a$$

$$(2)\ A \to a\,A\,a$$

$$(3)\ A \to b\,B\,b$$

$$(4)\ B \to b\,B\,b$$

$$(5)\ B \to c$$

$$\}$$

Lenguaje generado por una gramática de contexto libre

$\mathcal{L}(\mathcal{G}_3) = \{ a^i\ b^j\ c\ b^j\ a^i \mid i, j, > 0 \}$, *palíndromo impar*

$$S \Rightarrow_1 a\ \mathcal{A}\ a$$

$$\Rightarrow_2 a\ a\ \mathcal{A}\ a\ a$$

$$\Rightarrow_3 a\ a\ b\ \mathcal{B}\ b\ a\ a$$

$$\Rightarrow_4 a\ a\ b\ b\ \mathcal{B}\ b\ b\ a\ a$$

$$\Rightarrow_5 a\ a\ b\ b\ c\ b\ b\ a\ a$$

Ambigüedad

- Definición
- Gramática de las expresiones aritméticas
- Problema del "else danzante"
- Lenguajes "intrínsecamente" ambiguos

Ambigüedad

- **Definición**

G es una gramática ambigua si $\exists\ x \in V_T^*$ que posee

o **dos** derivaciones por la **izquierda** diferentes

o ***dos*** *derivaciones por la* ***derecha*** *diferentes*

o ***dos árboles sintácticos*** *diferentes*

Ambigüedad: expresiones aritméticas

Gramática ambigua

$$\mathcal{G}_1 = (\mathcal{V}_\mathcal{N}, \mathcal{V}_\mathcal{T}, \mathcal{P}_1, \mathcal{S})$$

$$\mathcal{P}_1 = \{$$

$$(1)\ \mathcal{S} \longrightarrow identificador = \mathcal{E}$$

$$(2)\ \mathcal{E} \longrightarrow \mathcal{E} + \mathcal{E}$$

$$(3)\ \mathcal{E} \longrightarrow \mathcal{E} * \mathcal{E}$$

$$(4)\ \mathcal{E} \longrightarrow (\mathcal{E})$$

$$(5)\ \mathcal{E} \longrightarrow identificador$$

$$(6)\ \mathcal{E} \longrightarrow número$$

$$\}$$

Ambigüedad: expresiones aritméticas

Primera derivación por la izquierda

$$S \Rightarrow_1 identificador = E$$

$$\Rightarrow_2 identificador = E + E$$

$$\Rightarrow_5 identificador = identificador + E$$

$$\Rightarrow_3 identificador = identificador + E * E$$

$$\Rightarrow_6 identificador = identificador + número * E$$

$$\Rightarrow_5 identificador = identificador + número * identificador$$

Ambigüedad: expresiones aritméticas

Segunda derivación por la izquierda

$$S \Rightarrow_1 identificador = E$$

$$\Rightarrow_2 identificador = E * E$$

$$\Rightarrow_3 identificador = E + E * E$$

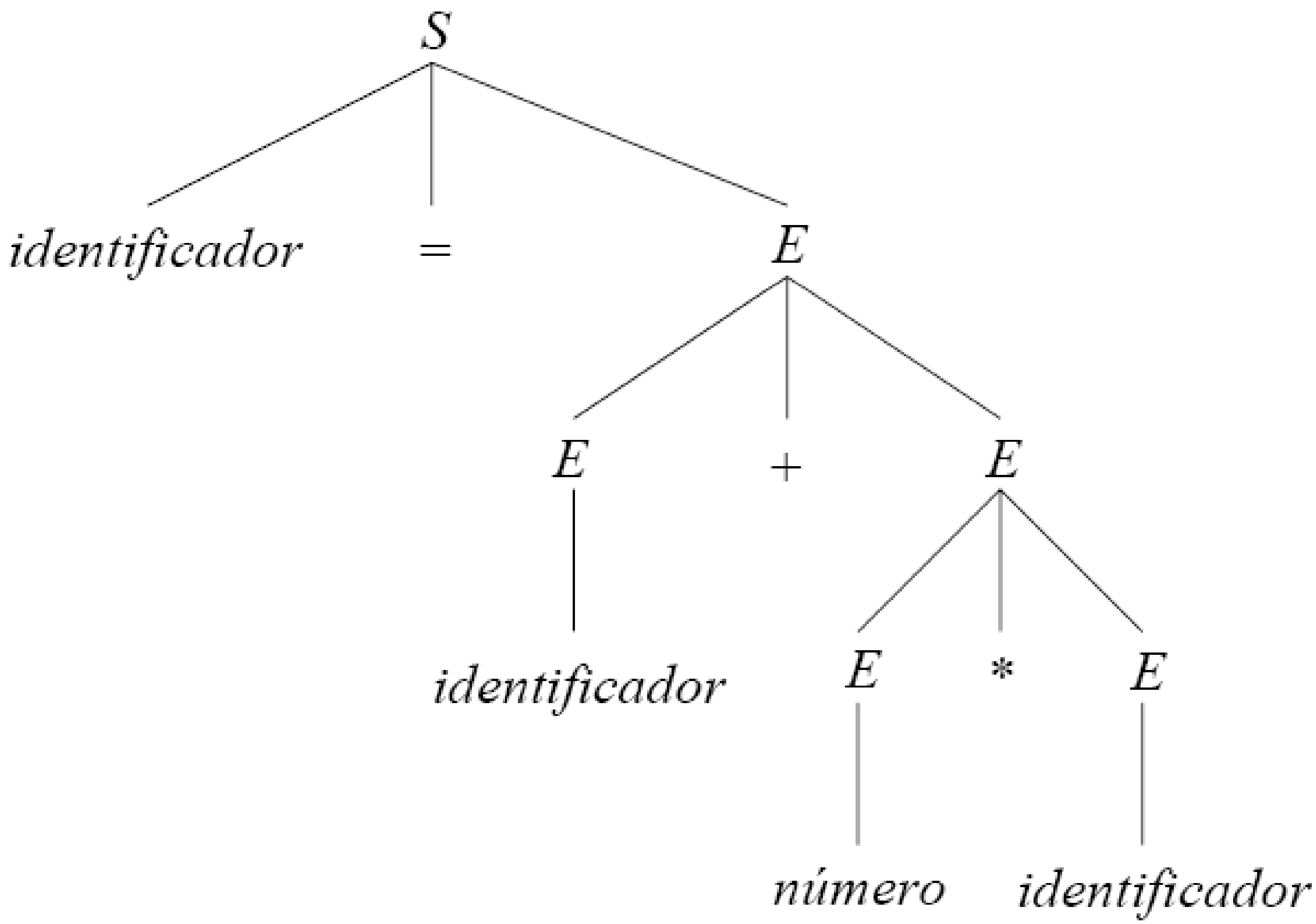
$$\Rightarrow_5 identificador = identificador + E * E$$

$$\Rightarrow_5 identificador = identificador + número * E$$

$$\Rightarrow_5 identificador = identificador + número * identificador$$

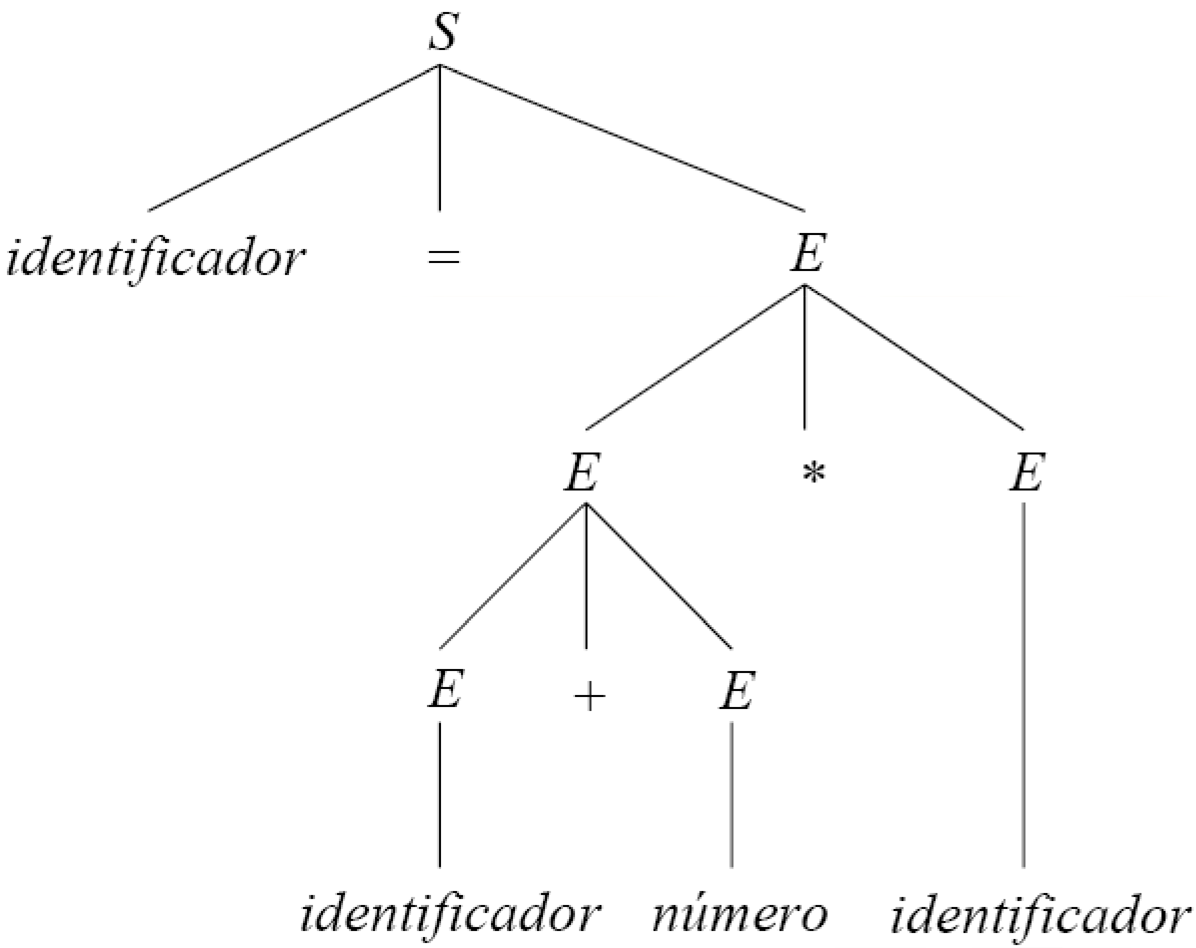
Ambigüedad: expresiones aritméticas

Árbol de la primera derivación por la izquierda



Ambigüedad: expresiones aritméticas

Árbol de la segunda derivación por la izquierda



Ambigüedad: expresiones aritméticas

Gramática que no es ambigua

$$\mathcal{G}_4 = (\mathcal{V}_\mathcal{N}, \mathcal{V}_\mathcal{T}, \mathcal{P}_4, \mathcal{S})$$

$$\mathcal{P}_4 = \{$$

$$(1)\ \mathcal{S} \longrightarrow identificador = \mathcal{E}$$

$$(2)\ \mathcal{E} \longrightarrow \mathcal{E} + \mathcal{T}$$

$$(3)\ \mathcal{E} \longrightarrow \mathcal{T}$$

$$(4)\ \mathcal{T} \longrightarrow \mathcal{T} * \mathcal{F}$$

$$(5)\ \mathcal{T} \longrightarrow \mathcal{F}$$

$$(6)\ \mathcal{F} \longrightarrow (\mathcal{E})$$

$$(7)\ \mathcal{F} \longrightarrow identificador$$

$$(8)\ \mathcal{F} \longrightarrow número$$

$$\}$$

Ambigüedad: expresiones aritméticas

Derivación por la izquierda con la gramática G_4

$$S \Rightarrow_1 identificador = E$$

$$\Rightarrow_2 identificador = E + T$$

$$\Rightarrow_3 identificador = T + T$$

$$\Rightarrow_5 identificador = F + T$$

$$\Rightarrow_7 identificador = identificador + T$$

$$\Rightarrow_4 identificador = identificador + T * F$$

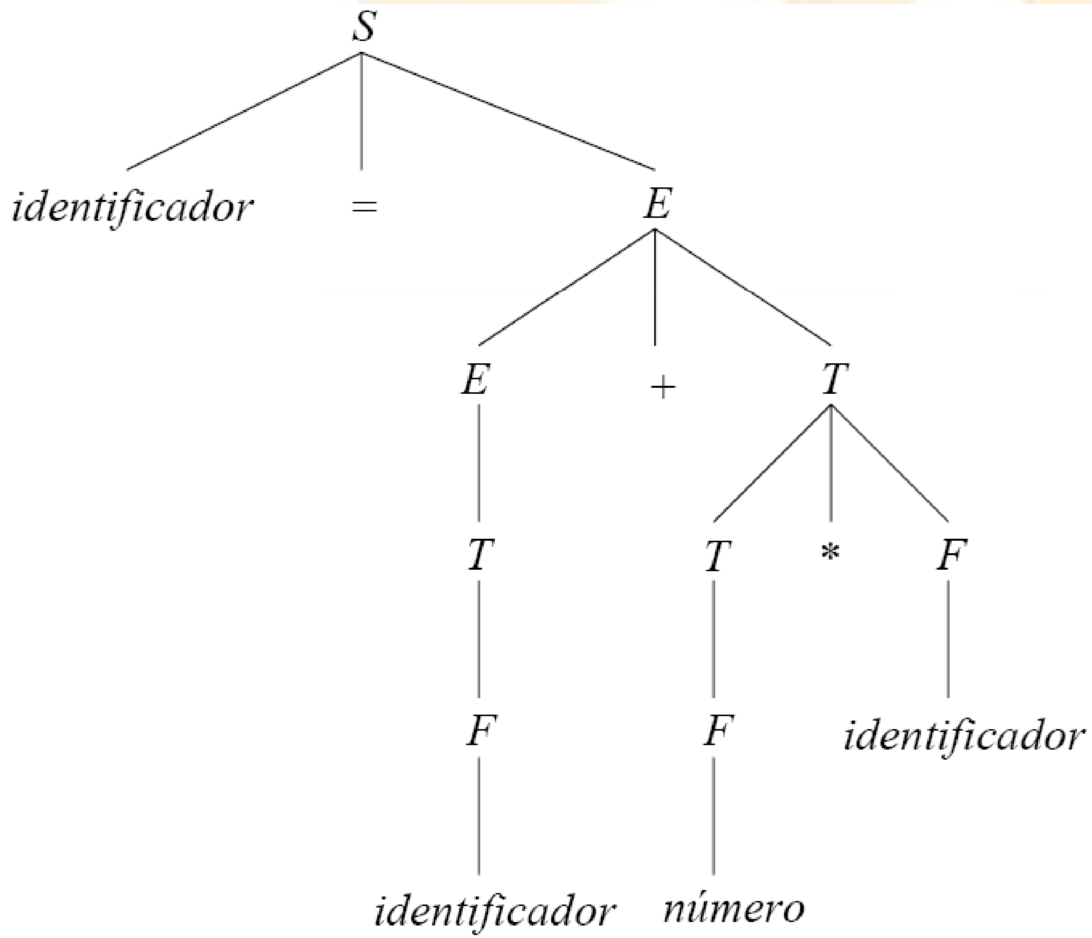
$$\Rightarrow_5 identificador = identificador + F * F$$

$$\Rightarrow_8 identificador = identificador + número * F$$

$$\Rightarrow_7 identificador = identificador + número * identificador$$

Ambigüedad: expresiones aritméticas

Árbol de derivación por la izquierda de G_4



Ambigüedad: problema del "else danzante"

$\mathcal{P} = \{ \ldots$

$(1) S \rightarrow if\ C\ S$

$(2) S \rightarrow if\ C\ S\ else\ S$

$(3) S \rightarrow I$

$\ldots \}$

1. **Primera** derivación por la izquierda

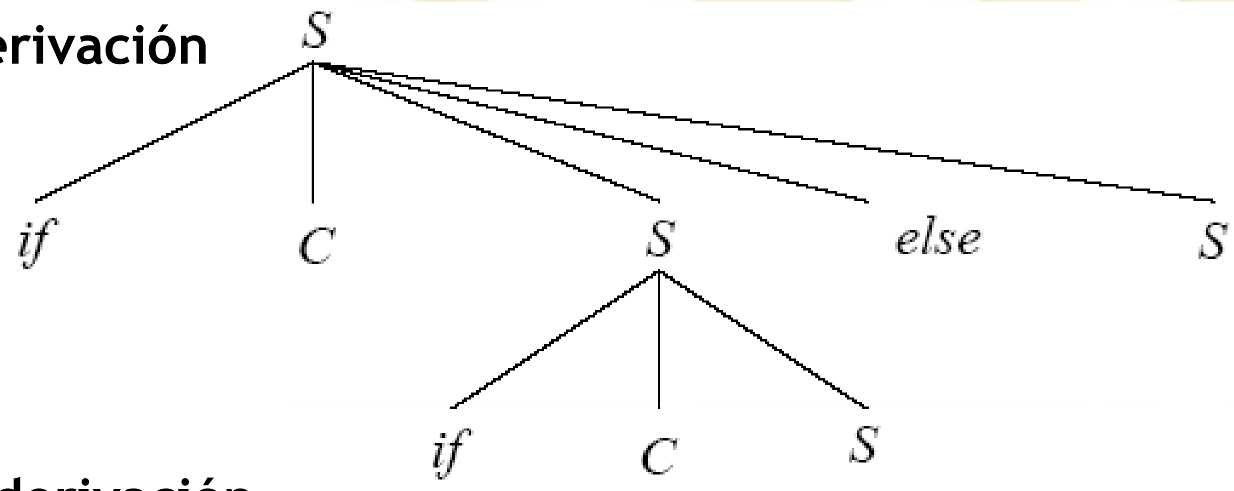
$$S \Rightarrow_1 if\ C\ S \Rightarrow_2 if\ C\ if\ C\ S\ else\ S$$

2. **Segunda** derivación por la izquierda

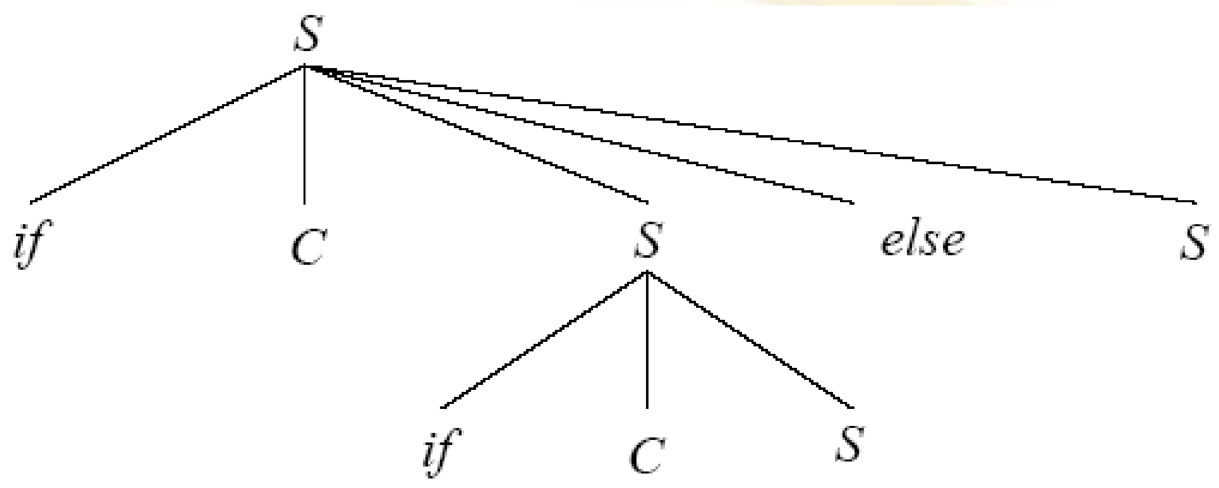
$$S \Rightarrow_2 if\ C\ S\ else\ S \Rightarrow_1 if\ C\ if\ C\ S\ else\ S$$

Ambigüedad: problema del "else danzante"

Árbol de la 1ª derivación



Árbol de la 2ª derivación



Ambigüedad: posible solución del problema del "else danzante"

$$\mathcal{P} = \{ \ldots$$

$$(1)\, S \rightarrow S1$$

$$(2)\, S \rightarrow S2$$

$$(3)\, S_1 \rightarrow \mathit{if}\; C\; S_1\; \mathit{else}\; S_1$$

$$(4)\, S_1 \rightarrow I$$

$$(5)\, S_2 \rightarrow \mathit{if}\; C\; S$$

$$(6)\, S_2 \rightarrow \mathit{if}\; C\; S_1\; \mathit{else}\; S_2$$

$$\ldots$$

$$\}$$

Ambigüedad: posible solución del problema del "else danzante"

Derivación

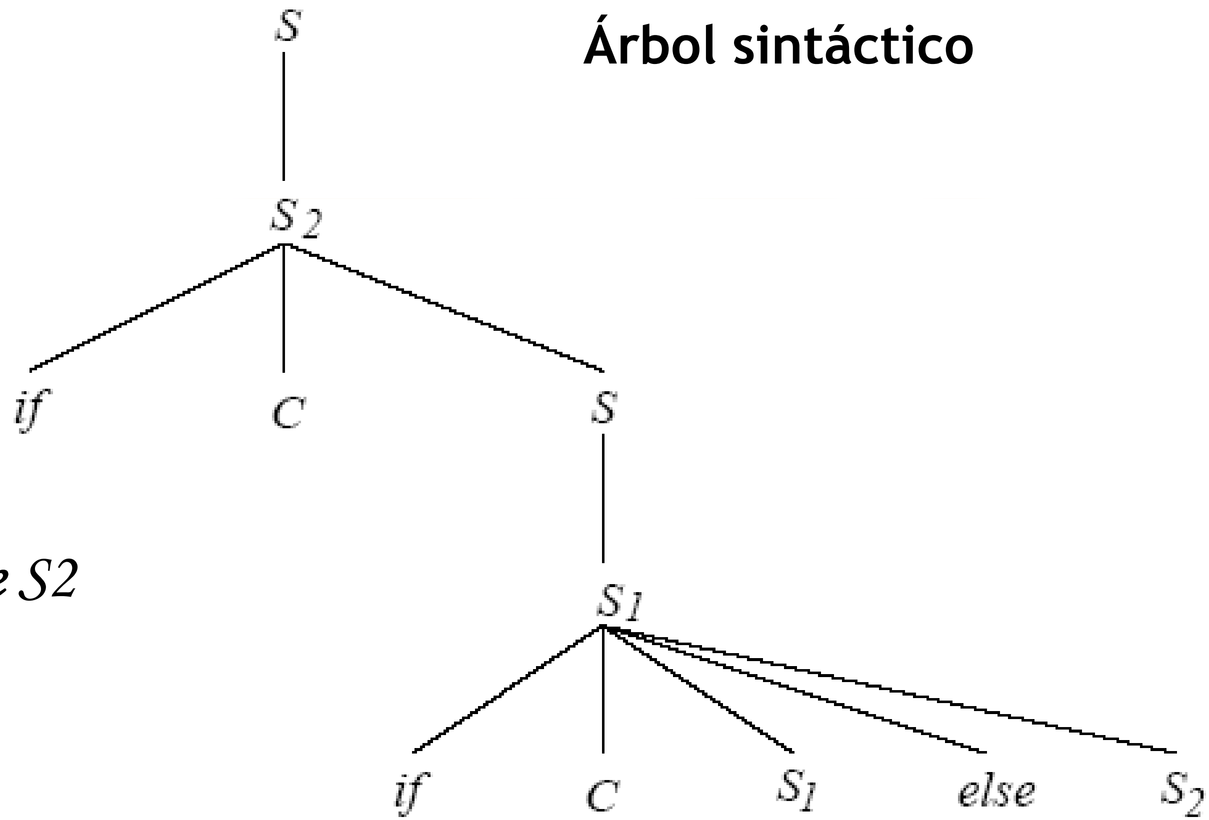
$$S \Rightarrow_2 S2$$

$$\Rightarrow_5 if\ C\ S$$

$$\Rightarrow_1 if\ C\ S1$$

$$\Rightarrow_3 if\ C\ if\ C\ S1\ else\ S2$$

Árbol sintáctico



Ambigüedad

Lenguaje intrínsecamente ambiguo

$$\mathcal{L} = \{a^i b^i c^j \mid i, j \geq 1\} \cup \{a^i b^j c^j \mid i, j \geq 1\}$$

$\mathcal{P}_5 = \{$

(1) $S \rightarrow AC$

(2) $S \rightarrow BD$

(3) $A \rightarrow a\,A\,b$

(4) $A \rightarrow a\,b$

(5) $C \rightarrow c\,C$

(6) $C \rightarrow c$

(7) $B \rightarrow a\,B$

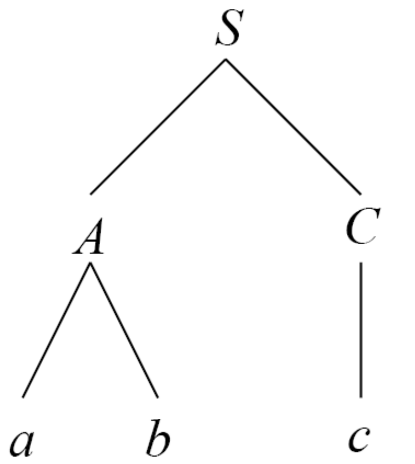
(8) $B \rightarrow a$

(9) $D \rightarrow b\,D\,c$

(10) $D \rightarrow b\,c$

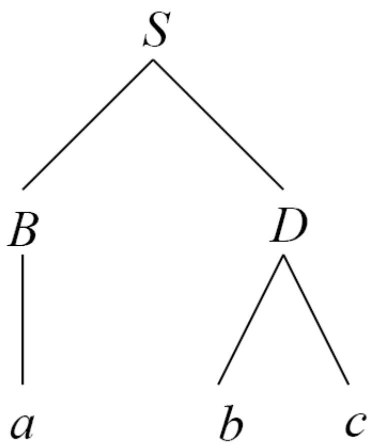
$\}$

Ambigüedad: lenguaje intrínsecamente ambiguo



Primera derivación por la izquierda

$$S \Rightarrow_1 \mathcal{A}\, C$$
$$\Rightarrow_4 a\, b\, C$$
$$\Rightarrow_6 a\, b\, c$$



Segunda derivación por la izquierda

$$S \Rightarrow_1 \mathcal{A}\, C$$
$$\Rightarrow_4 a\, b\, C$$
$$\Rightarrow_6 a\, b\, c$$

Vacuidad del lenguaje generado por una gramática de contexto libre

inicio

  $Viejo \leftarrow \varnothing$

  $Nuevo \leftarrow \{A \mid A \rightarrow x \in P \wedge x \in V_T^{\,*}\}$

  mientras $(Nuevo \neq Viejo) \wedge (S \notin Nuevo)$ **hacer**

    $Viejo \leftarrow Nuevo$

    $Nuevo \leftarrow Viejo \cup \{A \mid A \rightarrow \alpha \in P \wedge \alpha \in (V_T \cup Viejo)^*\}$

  fin mientras

  si $S \in Nuevo$

    entonces escribir *("La gramática genera un lenguaje no vacío")*

    si no escribir *("La gramática genera un lenguaje vacío")*

  fin si

fin

Vacuidad del lenguaje generado por una gramática de contexto libre

Ejemplo de aplicación del algoritmo

$\mathcal{P}_6 = \{$

$(1)\ \mathcal{S} \to \mathcal{A}\ b\ a$

$(2)\ \mathcal{A} \to \mathcal{B}\ \mathcal{D}\ b$

$(3)\ \mathcal{A} \to \mathcal{E}\ \mathcal{B}$

$(4)\ \mathcal{B} \to \mathcal{C}\ \mathcal{D}$

$(5)\ \mathcal{C} \to a\ b$

$(6)\ \mathcal{C} \to a\ \mathcal{A}$

$(7)\ \mathcal{D} \to b$

$(8)\ \mathcal{E} \to \mathcal{S}\ a$

$\}$

Paso	**Viejo**	**Nuevo**
0	$\emptyset$	$\{C, D\}$
1	$\{C, D\}$	$\{B, C, D\}$
2	$\{B, C, D\}$	$\{A, B, C, D\}$
3	$\{A, B, C, D\}$	$\{S, A, B, C, D\}$

Como $S \in$ Nuevo

$$\mathcal{L}(\mathcal{G}) \neq \emptyset$$

Algoritmo de *Supresión de símbolos no generadores*

- *Entrada:* $\mathcal{G} = (\mathcal{V}_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}, \mathcal{S})$
- *Salida:* $\mathcal{G}' = (\mathcal{V}'_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}', \mathcal{S})$ *sin símbolos no generadores*

[1] **inicio**

[2] $Viejo \leftarrow \varnothing$

[3] $Nuevo \leftarrow \{\mathcal{A} \mid \mathcal{A} \in \mathcal{V}_{\mathcal{N}} \wedge \exists \mathcal{A} \rightarrow x \in \mathcal{P} \wedge x \in \mathcal{V}_{\mathcal{T}}^{*}\}$

[4] **mientras** $(Nuevo \neq Viejo)$ **hacer**

[5] $Viejo \leftarrow Nuevo$

[6] $Nuevo \leftarrow Viejo \cup \{\mathcal{A} \mid \mathcal{A} \in \mathcal{V}_{\mathcal{N}} \wedge \exists \mathcal{A} \rightarrow \alpha \in \mathcal{P} \wedge \alpha \in (Viejo \cup \mathcal{V}_{\mathcal{T}})^{*}\}$

[7] **fin mientras**

[8] $\mathcal{V}'_{\mathcal{N}} \leftarrow Nuevo$

[9] $\mathcal{P}' \leftarrow \{\mathcal{A} \rightarrow \alpha \mid \mathcal{A} \rightarrow \alpha \in \mathcal{P} \wedge \mathcal{A} \in \mathcal{V}'_{\mathcal{N}} \wedge \alpha \in (\mathcal{V}'_{\mathcal{N}} \cup \mathcal{V}_{\mathcal{T}})^{*}\}$

[10] **fin**

Aplicación del algoritmo de *supresión de símbolos no generadores*

$\mathcal{P}_7 = \{$

$(1)\ S \rightarrow A\ B$

$(2)\ S \rightarrow A\ b$

$(3)\ A \rightarrow a\ C$

$(4)\ B \rightarrow b\ C\ a$

$(5)\ B \rightarrow D\ b\ E$

$(6)\ C \rightarrow b$

$(7)\ D \rightarrow F\ b$

$(8)\ E \rightarrow c\ a$

$(9)\ F \rightarrow a\ D$

$\}$

Paso	Viejo	Nuevo
0	$\emptyset$	$\{C, E\}$
1	$\{C, E\}$	$\{A, B, C, E\}$
2	$\{A, B, C, E\}$	$\{S, A, B, C, E\}$
3	$\{S, A, B, C, E\}$	$\{S, A, B, C, E\}$

Aplicación del algoritmo de *supresión de símbolos no generadores*

$\mathcal{P}_7 = \{$

(1) $S \longrightarrow A\ B$

(2) $S \longrightarrow A\ b$

(3) $A \longrightarrow a\ C$

(4) $B \longrightarrow b\ C\ a$

(5) $B \longrightarrow D\ b\ E$

(6) $C \longrightarrow b$

(7) $D \longrightarrow F\ b$

(8) $E \longrightarrow c\ a$

(9) $F \longrightarrow a\ D$

$\}$

Paso	Viejo	Nuevo
0	$\emptyset$	$\{C, E\}$
1	$\{C, E\}$	$\{A, B, C, E\}$
2	$\{A, B, C, E\}$	$\{S, A, B, C, E\}$
3	$\{S, A, B, C, E\}$	$\{S, A, B, C, E\}$

D y F no son generadores

Aplicación del algoritmo de *supresión de símbolos no generadores*

Gramática de contexto libre G'_7 sin símbolos no generadores

$\mathcal{P}'_7 = \{$

$(1)\ \mathcal{S} \rightarrow \mathcal{A}\ \mathcal{B}$

$(2)\ \mathcal{S} \rightarrow \mathcal{A}\ b$

$(3)\ \mathcal{A} \rightarrow a\ \mathcal{C}$

$(4)\ \mathcal{B} \rightarrow b\ \mathcal{C}\ a$

$(6)\ \mathcal{C} \rightarrow b$

$(8)\ \mathcal{E} \rightarrow c\ a$

$\}$

Algoritmo de Supresión de símbolos no accesibles

- Entrada: $G' = (V'_N, V_T, P', S)$ sin símbolos no generadores
- Salida: $G'' = (V''_N, V'_T, P'', S)$ sin símbolos no generadores o no accesibles

inicio

$Viejo \leftarrow \{S\}$

$Nuevo \leftarrow Viejo \cup \{X \mid X \in (V'_N \cup V_T) \wedge \exists S \rightarrow \alpha X \beta \in P' \wedge \alpha, \beta \in (V'_N \cup V_T)^* \}$

mientras $(Nuevo \neq Viejo)$ **hacer**

$Viejo \leftarrow Nuevo$

$Nuevo \leftarrow Viejo \cup \{X \mid \exists A \rightarrow \alpha X \beta \in P' \wedge A \in Viejo \wedge$

$X \in (V'_N \cup V_T) \wedge \alpha, \beta \in (V'_N \cup V_T)^* \}$

fin mientras

$V''_N \leftarrow Nuevo \cap V'_N$

$V'_T \leftarrow Nuevo \cap V_T$

$P'' \leftarrow \{A \rightarrow \alpha \mid A \rightarrow \alpha \in P' \wedge A \in V''_N \wedge \alpha \in (V''_N \cup V'_T)^* \}$

fin

Aplicación del algoritmo de *Supresión de símbolos no accesibles*

$\mathcal{P}'_7 = \{$

$(1)\ S \rightarrow A\ B$

$(2)\ S \rightarrow A\ b$

$(3)\ A \rightarrow a\ C$

$(4)\ B \rightarrow b\ C\ a$

$(6)\ C \rightarrow b$

$(8)\ E \rightarrow c\ a$

$\}$

Paso	Viejo	Nuevo
0	$\{S\}$	$\{S, A, B, b\}$
1	$\{S, A, B, b\}$	$\{S, A, B, C, a, b\}$
2	$\{S, A, B, C, a, b\}$	$\{S, A, B, C, a, b\}$

Aplicación del algoritmo de *Supresión de símbolos no accesibles*

$\mathcal{P}'_7 = \{$

(1) $S \rightarrow A\,B$

(2) $S \rightarrow A\,b$

(3) $A \rightarrow a\,C$

(4) $B \rightarrow b\,C\,a$

(6) $C \rightarrow b$

(8) $E \rightarrow c\,a$

$\}$

Paso	Viejo	Nuevo
0	$\{S\}$	$\{S, A, B, b\}$
1	$\{S, A, B, b\}$	$\{S, A, B, C, a, b\}$
2	$\{S, A, B, C, a, b\}$	$\{S, A, B, C, a, b\}$

E y c no son accesibles

Gramática de contexto libre G''_7

sin símbolos no accesibles

$\mathcal{P}''_7 = \{$

$(1)\ S \rightarrow A\ B$

$(2)\ S \rightarrow A\ b$

$(3)\ A \rightarrow a\ C$

$(4)\ B \rightarrow b\ C\ a$

$(6)\ C \rightarrow b$

$\}$

Correcta eliminación de símbolos inútiles

1. Eliminación de símbolos no generadores
2. Eliminación de símbolos no accesibles

- **Observación**
  - Si estos **pasos** se aplican en **orden inverso** entonces **no se garantiza** que se eliminen todos los símbolos inútiles

Aplicación incorrecta de los algoritmos de supresión de símbolos inútiles

1. Eliminación de símbolos no accesibles

$\mathcal{P}_7 = \{$

(1) $S \to A\,B$

(2) $S \to A\,b$

(3) $A \to a\,C$

(4) $B \to b\,C\,a$

(5) $B \to D\,b\,E$

(6) $C \to b$

(7) $D \to F\,b$

(8) $E \to c\,a$

(9) $F \to a\,D$

$\}$

Paso	Viejo	Nuevo
0	$\{S\}$	$\{S, A, B, b\}$
1	$\{S, A, B, b\}$	$\{S, A, B, C, D, E, a, b\}$
2	$\{S, A, B, C, D, E, a, b\}$	$\{S, A, B, C, D, E, F, a, b, c\}$
3	$\{S, A, B, C, D, E, F, a, b\}$	$\{S, A, B, C, D, E, F, a, b, c\}$

Aplicación incorrecta de los algoritmos de supresión de símbolos inútiles

1. Eliminación de símbolos no accesibles

$P_7 = \{$

$(1)\ S \rightarrow A\ B$

$(2)\ S \rightarrow A\ b$

$(3)\ A \rightarrow a\ C$

$(4)\ B \rightarrow b\ C\ a$

$(5)\ B \rightarrow D\ b\ E$

$(6)\ C \rightarrow b$

$(7)\ D \rightarrow F\ b$

$(8)\ E \rightarrow c\ a$

$(9)\ F \rightarrow a\ D$

$\}$

Paso	Viejo	Nuevo
0	$\{S\}$	$\{S, A, B, b\}$
1	$\{S, A, B, b\}$	$\{S, A, B, C, D, E, a, b\}$
2	$\{S, A, B, C, D, E, a, b\}$	$\{S, A, B, C, D, E, F, a, b, c\}$
3	$\{S, A, B, C, D, E, F, a, b\}$	$\{S, A, B, C, D, E, F, a, b, c\}$

- **Observación**
  - No se elimina **ningún** símbolo

Aplicación incorrecta de los algoritmos de supresión de símbolos inútiles

1. Eliminación de símbolos no generadores

$\mathcal{P}_7 = \{$

(1) $S \rightarrow A\ B$

(2) $S \rightarrow A\ b$

(3) $A \rightarrow a\ C$

(4) $B \rightarrow b\ C\ a$

(5) $B \rightarrow D\ b\ E$

(6) $C \rightarrow b$

(7) $D \rightarrow F\ b$

(8) $E \rightarrow c\ a$

(9) $F \rightarrow a\ D$

$\}$

Paso	Viejo	Nuevo
0	$\emptyset$	$\{C, E\}$
1	$\{C, E\}$	$\{A, B, C, E\}$
2	$\{A, B, C, E\}$	$\{S, A, B, C, E\}$
3	$\{S, A, B, C, E\}$	$\{S, A, B, C, E\}$

Aplicación incorrecta de los algoritmos de supresión de símbolos inútiles

2. Eliminación de símbolos no generadores

$P_7 = \{$

(1) $S \rightarrow A\,B$

(2) $S \rightarrow A\,b$

(3) $A \rightarrow a\,C$

(4) $B \rightarrow b\,C\,a$

(5) $B \rightarrow D\,b\,E$

(6) $C \rightarrow b$

(7) $D \rightarrow F\,b$

(8) $E \rightarrow c\,a$

(9) $F \rightarrow a\,D$

$\}$

Paso	Viejo	Nuevo
0	$\emptyset$	$\{C, E\}$
1	$\{C, E\}$	$\{A, B, C, E\}$
2	$\{A, B, C, E\}$	$\{S, A, B, C, E\}$
3	$\{S, A, B, C, E\}$	$\{S, A, B, C, E\}$

D y F no son generadores

Aplicación incorrecta de los algoritmos de supresión de símbolos inútiles

2. Eliminación de símbolos no generadores

$\mathcal{P}'_7 = \{$

$(1)\ S \to A\ B$

$(2)\ S \to A\ b$

$(3)\ A \to a\ C$

$(4)\ B \to b\ C\ a$

$(6)\ C \to b$

$(8)\ E \to c\ a$

$\}$

- **Observación**
  - o E y c se convierten en no **accesibles**

Algoritmo de *Obtención* de los símbolos *anulables*

- *Entrada:* $\mathcal{G}' = (\mathcal{V}'_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}', \mathcal{S})$ sin símbolos inútiles
- *Salida:* conjunto de símbolos anulables

[1] **inicio**

[2] $Viejo \leftarrow \varnothing$

[3] $Nuevo \leftarrow \{\mathcal{A} \mid \mathcal{A} \in \mathcal{V}_{\mathcal{N}} \wedge \exists\, \mathcal{A} \rightarrow \varepsilon \in \mathcal{P}\}$

[4] **mientras** $(Nuevo \neq Viejo)$ **hacer**

[5] $Viejo \leftarrow Nuevo$

[6] $Nuevo \leftarrow Viejo \cup \{\mathcal{A} \mid \mathcal{A} \in \mathcal{V}_{\mathcal{N}} \wedge \exists\, \mathcal{A} \rightarrow \alpha \in \mathcal{P} \wedge \alpha \in Viejo^{*}\}$

[7] **fin mientras**

[8] $Anulables \leftarrow Nuevo$

[9] **fin**

Aplicación del algoritmo de *Obtención de los símbolos anulables*

$P_8 = \{$

$(1) S \rightarrow A\,D$ $(2) S \rightarrow B$

$(3) A \rightarrow C\,D\,E$

$(4) B \rightarrow C\,E$

$(5) C \rightarrow S$ $(5) C \rightarrow a$ $(7) C \rightarrow \varepsilon$

$(8) D \rightarrow A$ $(9) D \rightarrow b$

$(10) E \rightarrow S$ $(11) E \rightarrow a$ $(12) E \rightarrow \varepsilon$

$\}$

Paso	**Viejo**	**Nuevo**
0	$\emptyset$	$\{C, E\}$
1	$\{C, E\}$	$\{B, C, E\}$
2	$\{B, C, E\}$	$\{S, B, C, E\}$
3	$\{S, B, C, E\}$	$\{S, B, C, E\}$

Gramática sin ε

• No posee reglas - ε

• o sólo posee la regla $S \rightarrow \varepsilon$ y S no aparece en la parte derecha de ninguna regla de producción

Algoritmo de *Obtención* de Gramática sin ε

- *Entrada:*
  - $\mathcal{G} = (\mathcal{V}_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}, \mathcal{S})$ *sin símbolos inútiles*
  - *El conjunto de símbolos anulables de* $\mathcal{G}$
- *Salida:* $\mathcal{G}' = (\mathcal{V}'_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}', \mathcal{S}')$ *gramática sin* ε

Algoritmo de *Obtención* de Gramática sin ε

inicio
    $\mathcal{P}' \leftarrow \{\mathcal{A} \rightarrow \alpha \mid \mathcal{A} \rightarrow \alpha \in \mathcal{P} \wedge \alpha \neq \varepsilon \wedge \alpha$ *no contiene ningún símbolo anulable* $\}$
    para $(\mathcal{A} \rightarrow \alpha \in \mathcal{P} \wedge \alpha$ *contiene algún símbolo anulable* $)$ ***hacer***
        si $\alpha = \alpha_0\ \mathcal{B}_1\, \alpha_1 \ldots\ \mathcal{B}_k\, \alpha_k$
            $\wedge\ \forall i\ (\mathcal{B}_i \in$ *Anulables y* α_i ***no*** *contiene símbolos anulables*$)$
        entonces
        $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathcal{A} \rightarrow \alpha_0\ X_1\, \alpha_1 \ldots\ X_k\, \alpha_k \mid$
            $\forall i\ (X_i = \mathcal{B}_i \vee X_i = \varepsilon) \wedge \alpha_0\ X_1\, \alpha_1 \ldots\ X_k\, \alpha_k \neq \varepsilon \}$
        fin si
    fin para
    si $S \in$ *Anulables* ***entonces***
        $\mathcal{V}'_{\mathcal{N}} \leftarrow \mathcal{V}'_{\mathcal{N}} \cup \{S'\}$
        $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{S' \rightarrow \varepsilon, S' \rightarrow S\}$
    fin si
fin

Aplicación del algoritmo de *Obtención de Gramática sin* ε

$\mathcal{P}_8 = \{$

$(1)\ S \longrightarrow A\,D \qquad (2)\ S \longrightarrow B$

$(3)\ A \longrightarrow C\,D\,E$

$(4)\ B \longrightarrow C\,E$

$(5)\ C \longrightarrow S \qquad (6)\ C \longrightarrow a \qquad (7)\ C \longrightarrow \varepsilon$

$(8)\ D \longrightarrow A \qquad (9)\ D \longrightarrow b$

$(10)\ E \longrightarrow S \qquad (11)\ E \longrightarrow a \qquad (12)\ E \longrightarrow \varepsilon$

$\}$

Aplicación del algoritmo de Obtención de Gramática sin ε

- *Obtención de los símbolos anulables*

$\mathcal{P}_8 = \{$

$(1)\ S \rightarrow A\,D \qquad (2)\ S \rightarrow B$

$(3)\ A \rightarrow C\,D\,E$

$(4)\ B \rightarrow C\,E$

$(5)\ C \rightarrow S \qquad (6)\ C \rightarrow a \qquad (7)\ C \rightarrow \varepsilon$

$(8)\ D \rightarrow A \qquad (9)\ D \rightarrow b$

$(10)\ E \rightarrow S \qquad (11)\ E \rightarrow a \qquad (12)\ E \rightarrow \varepsilon$

$\}$

Anulables $= \{S, B, C, E\}$

Aplicación del algoritmo de *Obtención* de Gramática *sin* ε

- *Paso 0:* *inclusión de las reglas que* ***no*** *contienen símbolos* ***anulables*** *y que no son reglas* ε

$\mathcal{P}'_8 = \{$

$(1)\ \mathcal{S} \rightarrow \mathcal{A}\ \mathcal{D}$

$(6)\ \mathcal{C} \rightarrow \boldsymbol{a}$

$(8)\ \mathcal{D} \rightarrow \mathcal{A}$ $\qquad (9)\ \mathcal{D} \rightarrow \boldsymbol{b}$

$(11)\ \mathcal{E} \rightarrow \boldsymbol{a}$

$\}$

Aplicación del algoritmo de Obtención de Gramática sin ε

•*Paso 1: inclusión de las reglas que se generan a partir de* $(2)\ S \rightarrow B$

$P'_8 = \{$

$(1)\ S \rightarrow A\,D \qquad (2)\ S \rightarrow B$

$(6)\ C \rightarrow a$

$(8)\ D \rightarrow A \qquad (9)\ D \rightarrow b$

$(11)\ E \rightarrow a$

$\}$

•Observación

La regla $S \rightarrow \varepsilon$ *no se añade*

Aplicación del algoritmo de Obtención de Gramática sin ε

• *Paso 2: inclusión de las reglas que se generan a partir de* $(3)\,A \longrightarrow C\,D\,E$

$P'_8 = \{$

$(1)\,S \longrightarrow A\,D \qquad (2)\,S \longrightarrow B$

$(3)\,A \longrightarrow C\,D\,E$

$(3a)\,A \longrightarrow D\,E \quad (3b)\,A \longrightarrow C\,D \quad (3c)\,A \longrightarrow D$

$(6)\,C \longrightarrow a$

$(8)\,D \longrightarrow A \qquad (9)\,D \longrightarrow b$

$(11)\,E \longrightarrow a$

$\}$

Aplicación del algoritmo de Obtención de Gramática sin ε

• *Paso 3: inclusión de las reglas que se generan a partir de* $(4)\ B \longrightarrow C\ E$

$P'_8 = \{$

$(1)\ S \longrightarrow A\ D \qquad (2)\ S \longrightarrow B$

$(3)\ A \longrightarrow C\ D\ E$

$(3a)\ A \longrightarrow D\ E \qquad (3b)\ A \longrightarrow C\ D \qquad (3c)\ A \longrightarrow D$

$(4)\ B \longrightarrow C\ E \qquad (4a)\ B \longrightarrow E \qquad (4b)\ B \longrightarrow C$

$(6)\ C \longrightarrow a$

$(8)\ D \longrightarrow A \qquad (9)\ D \longrightarrow b$

$(11)\ E \longrightarrow a$

$\}$

•Observación

La regla $B \rightarrow \varepsilon$ no se añade

Aplicación del algoritmo de *Obtención* *de Gramática* *sin* ε

• *Paso 3: inclusión de las reglas que se generan a partir de* $(5)\, C \rightarrow S$

$P'_8 = \{$

$(1)\, S \rightarrow A\, D \qquad (2)\, S \rightarrow B$

$(3)\, A \rightarrow C\, D\, E$

$(3a)\, A \rightarrow D\, E \qquad (3b)\, A \rightarrow C\, D \qquad (3c)\, A \rightarrow D$

$(4)\, B \rightarrow C\, E \qquad (4a)\, B \rightarrow E \qquad (4b)\, B \rightarrow C$

$(6)\, C \rightarrow a \qquad (5)\, C \rightarrow S$

$(8)\, D \rightarrow A \qquad (9)\, D \rightarrow b$

$(11)\, E \rightarrow a$

$\}$

•Observación

La regla $C \rightarrow \varepsilon$ no se añade

Aplicación del algoritmo de Obtención de Gramática sin ε

• *Paso 3: inclusión de las reglas que se generan a partir de (10) $E \rightarrow S$*

$P'_8 = \{$

$(1)\, S \rightarrow A\, D \qquad (2)\, S \rightarrow B$

$(3)\, A \rightarrow C\, D\, E$

$(3a)\, A \rightarrow D\, E \qquad (3b)\, A \rightarrow C\, D \qquad (3c)\, A \rightarrow D$

$(4)\, B \rightarrow C\, E \qquad (4a)\, B \rightarrow E \qquad (4b)\, B \rightarrow C$

$(6)\, C \rightarrow a \qquad (5)\, C \rightarrow S$

$(8)\, D \rightarrow A \qquad (9)\, D \rightarrow b$

$(11)\, E \rightarrow a \qquad (10)\, E \rightarrow S$

$\}$

•Observación

La regla $E \rightarrow \varepsilon$ no se añade

Aplicación del algoritmo de Obtención de Gramática sin ε

- *Paso final: como el símbolo S es anulable, se añaden dos reglas más*

$P'_8 = \{$ (1') $S' \rightarrow S$ (1'') $S' \rightarrow \varepsilon$

(1) $S \rightarrow A\,D$ (2) $S \rightarrow B$

(3) $A \rightarrow C\,D\,E$

(3a) $A \rightarrow D\,E$ (3b) $A \rightarrow C\,D$ (3c) $A \rightarrow D$

(4) $B \rightarrow C\,E$ (4a) $B \rightarrow E$ (4b) $B \rightarrow C$

(6) $C \rightarrow a$ (5) $C \rightarrow S$

(8) $D \rightarrow A$ (9) $D \rightarrow b$

(11) $E \rightarrow a$ (10) $E \rightarrow S$

$\}$

Aplicación del algoritmo de Obtención de Gramática sin ε

- *Resultado final*

$P'_8 = \{$ (1) $S' \rightarrow S$ (2) $S' \rightarrow \varepsilon$

(3) $S \rightarrow B$ (4) $S \rightarrow A\,D$

(5) $A \rightarrow C\,D\,E$ (6) $A \rightarrow D\,E$

(7) $A \rightarrow C\,D$ (8) $A \rightarrow D$

(9) $B \rightarrow C\,E$ (10) $B \rightarrow E$ (11) $B \rightarrow C$

(12) $C \rightarrow a$ (13) $C \rightarrow S$

(14) $D \rightarrow A$ (15) $D \rightarrow b$

(16) $E \rightarrow a$ (17) $E \rightarrow S$

$\}$

Eliminación de reglas unitarias

Regla unitaria: $\mathcal{A} \longrightarrow \mathcal{B}$ donde $\mathcal{A}, \mathcal{B} \in \mathcal{V}_{\mathcal{N}}$

$\mathcal{G}_4 = (\mathcal{V}_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}_4, \mathcal{S})$

$\mathcal{P}_4 = \{$

$(1)\ \mathcal{S} \longrightarrow identificador = \mathcal{E}$

$(2)\ \mathcal{E} \longrightarrow \mathcal{E} + \mathcal{T}$

$(3)\ \mathcal{E} \longrightarrow \mathcal{T}$

$(4)\ \mathcal{T} \longrightarrow \mathcal{T} * \mathcal{F}$

$(5)\ \mathcal{T} \longrightarrow \mathcal{F}$

$(6)\ \mathcal{F} \longrightarrow (\ \mathcal{E}\ )$

$(7)\ \mathcal{F} \longrightarrow identificador$

$(8)\ \mathcal{F} \longrightarrow número$

$\}$

Eliminación de reglas unitarias

Uso de las reglas unitarias

$$\mathcal{S} \Rightarrow_1 identificador = \mathcal{E}$$

$$\Rightarrow_2 identificador = \mathcal{E} + \mathcal{T}$$

$$\Rightarrow_3 identificador = \mathcal{T} + \mathcal{T}$$

$$\Rightarrow_5 identificador = \mathcal{F} + \mathcal{T}$$

$$\Rightarrow_7 identificador = identificador + \mathcal{T}$$

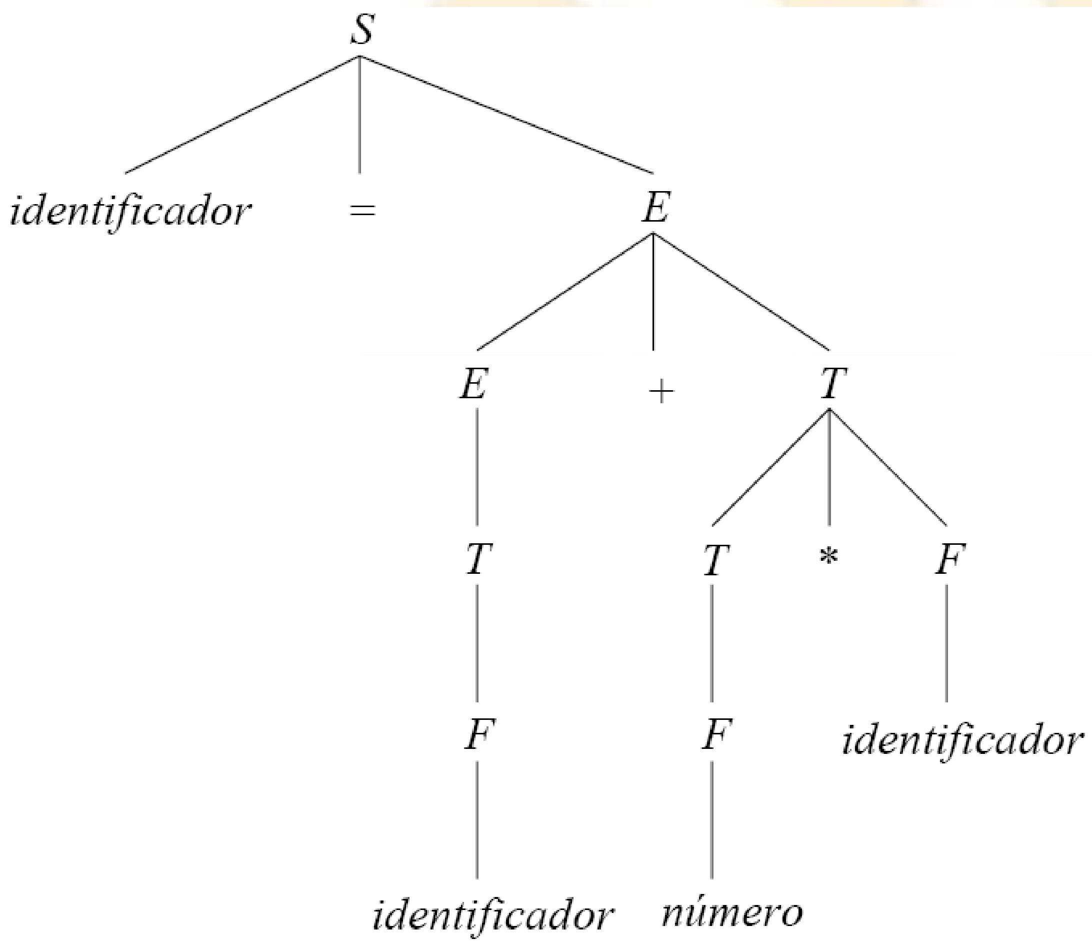
$$\Rightarrow_4 identificador = identificador + \mathcal{T} * \mathcal{F}$$

$$\Rightarrow_5 identificador = identificador + \mathcal{F} * \mathcal{F}$$

$$\Rightarrow_8 identificador = identificador + número * \mathcal{F}$$

$$\Rightarrow_7 identificador = identificador + número * identificador$$

Eliminación de reglas unitarias



Eliminación de reglas unitarias

Las reglas unitarias

$$(3)\ E \longrightarrow T$$

y $(5)\ T \longrightarrow F$

pueden ser **sustituidas** por las reglas

$$E \longrightarrow \textit{identificador}$$

$$E \longrightarrow \textit{número}$$

$$T \longrightarrow \textit{identificador}$$

$$T \longrightarrow \textit{número}$$

Eliminación de reglas unitarias

$$S \Rightarrow_1 identificador = E$$

$$\Rightarrow_2 identificador = E + T$$

$$\Rightarrow identificador = identificador + T$$

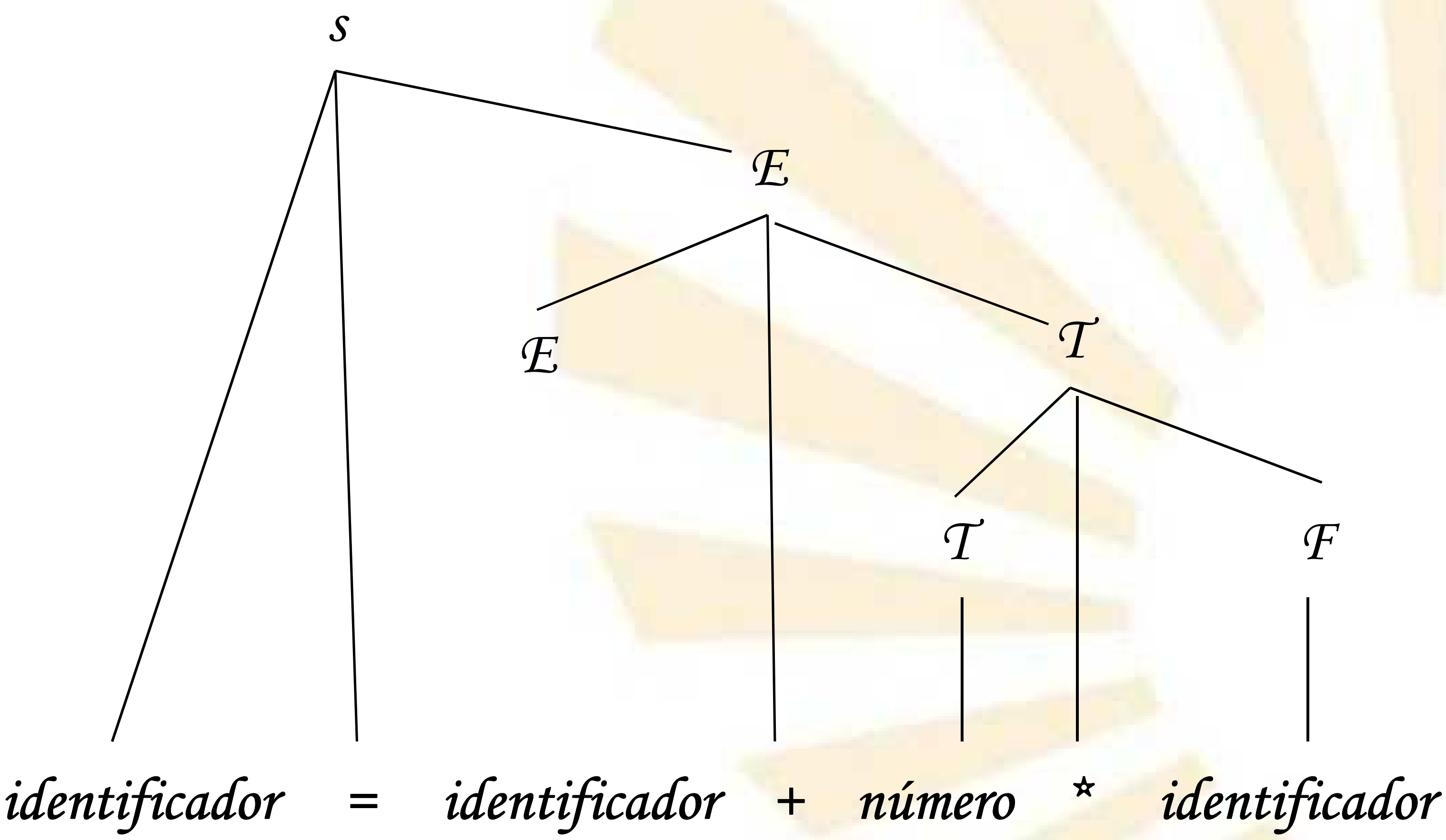
$$\Rightarrow_4 identificador = identificador + T * F$$

$$\Rightarrow identificador = identificador + número * F$$

$$\Rightarrow_7 identificador = identificador + número * identificador$$

Derivación sin reglas unitarias

Eliminación de reglas unitarias



Eliminación de reglas unitarias

- ***Algoritmo*** *para obtener los símbolos* ***no terminales*** *accesibles mediante* ***reglas unitarias***

- ***Algoritmo*** *para* ***eliminar*** *las reglas unitarias*

Algoritmo para obtener los símbolos no terminales accesibles mediante reglas unitarias

[1] *inicio*

[2] $Viejo \leftarrow \varnothing$

[3] $Nuevo \leftarrow \{A\}$

[4] *mientras* $(Nuevo \neq Viejo)$ *hacer*

[5] $Viejo \leftarrow Nuevo$

[6] $Nuevo \leftarrow Viejo \cup \{B \mid A \rightarrow B \in P \wedge A \in Viejo\}$

[7] *fin mientras*

[8] $N_A \leftarrow Nuevo$

[9] *fin*

Eliminación de reglas unitarias

$\mathcal{P}_4 = \{$

(1) $S \longrightarrow identificador = E$

(2) $E \longrightarrow E + T$

(3) $E \longrightarrow T$

(4) $T \longrightarrow T * F$

(5) $T \longrightarrow F$

(6) $F \longrightarrow (E)$

(7) $F \longrightarrow identificador$

(8) $F \longrightarrow número$

$\}$

Cálculo de $\mathcal{N}_E$

Paso	Viejo	Nuevo
0	$\emptyset$	$\{E\}$
1	$\{E\}$	$\{E, T\}$
2	$\{E, T\}$	$\{E, T, F\}$
3	$\{E, T, F\}$	$\{E, T, F\}$

$$\mathcal{N}_S = \{S\}$$

$$\mathcal{N}_E = \{E, T, F\}$$

$$\mathcal{N}_T = \{T, F\}$$

$$\mathcal{N}_F = \{F\}$$

Algoritmo para eliminar las reglas unitarias

- Entrada: $\mathcal{G} = (\mathcal{V}_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}, \mathcal{S})$ sin símbolos inútiles y $\mathcal{N}_{\mathcal{A}}$ $\forall \mathcal{A} \in \mathcal{V}_{\mathcal{N}}$
- Salida: $\mathcal{G}' = (\mathcal{V}'_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}', \mathcal{S}')$ gramática sin reglas unitarias

[1] inicio

[2] $\mathcal{P}' \leftarrow \varnothing$

[3] para cada $(\mathcal{A} \in \mathcal{V}_{\mathcal{N}})$ hacer

[4] para cada $(\mathcal{B} \in \mathcal{N}_{\mathcal{A}})$ hacer

[5] si $\mathcal{B} \rightarrow \alpha$ es no unitaria entonces

[6] $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathcal{A} \rightarrow \alpha\}$

[7] fin si

[8] fin para

[9] fin para

[10] $\mathcal{V}'_{\mathcal{N}} \leftarrow \{\mathcal{A} \mid \mathcal{A} \in \mathcal{V}_{\mathcal{N}} \wedge \exists\, \mathcal{A} \rightarrow \alpha \in \mathcal{P}'\}$

[11] fin

Ejemplo de eliminación de reglas unitarias

$\mathcal{G}_4 = (\mathcal{V}_\mathcal{N}, \mathcal{V}_\mathcal{T}, \mathcal{P}_4, \mathcal{S})$

$\mathcal{P}_4 = \{$

$(1)\ \mathcal{S} \longrightarrow identificador = \mathcal{E}$

$(2)\ \mathcal{E} \longrightarrow \mathcal{E} + \mathcal{T}$

$(3)\ \mathcal{E} \longrightarrow \mathcal{T}$

$(4)\ \mathcal{T} \longrightarrow \mathcal{T} * \mathcal{F}$

$(5)\ \mathcal{T} \longrightarrow \mathcal{F}$

$(6)\ \mathcal{F} \longrightarrow (\mathcal{E})$

$(7)\ \mathcal{F} \longrightarrow identificador$

$(8)\ \mathcal{F} \longrightarrow número$

$\}$

$\mathcal{N}_\mathcal{S} = \{\mathcal{S}\}$

$\mathcal{N}_\mathcal{E} = \{\mathcal{E}, \mathcal{T}, \mathcal{F}\}$

$\mathcal{N}_\mathcal{T} = \{\mathcal{T}, \mathcal{F}\}$

$\mathcal{N}_\mathcal{F} = \{\mathcal{F}\}$

Ejemplo de eliminación de reglas unitarias

Reglas de S

$\mathcal{P}'_4 = \{$

(1) $S \rightarrow identificador = E$

$\}$

$\mathcal{N}_S = \{S\}$

Ejemplo de eliminación de reglas unitarias

Reglas de $\mathcal{E}$

$\mathcal{P}'_4 = \{$

$(1)\ \mathcal{S} \longrightarrow identificador = \mathcal{E}$

$(2)\ \mathcal{E} \longrightarrow \mathcal{E} + \mathcal{T}$

$(3a)\ \mathcal{E} \longrightarrow \mathcal{T} * \mathcal{F}$

$(3b)\ \mathcal{E} \longrightarrow (\ \mathcal{E}\ )$ $(3c)\ \mathcal{E} \longrightarrow identificador$ $(3d)\ \mathcal{E} \longrightarrow número$

$\}$

$\boxed{\mathcal{N}_{\mathcal{E}} = \{\mathcal{E}, \mathcal{T}, \mathcal{F}\}}$

Ejemplo de eliminación de reglas unitarias

Reglas de T

$P'_4 = \{$

$(1)\; S \to \textit{identificador} = E$

$N_T = \{T, F\}$

$(2)\; E \to E + T$

$(3a)\; E \to T * F$

$(3b)\; E \to (\, E\,) \quad (3c)\; E \to \textit{identificador} \quad (3d)\; E \to \textit{número}$

$(4)\; T \to T * F$

$(4a)\; T \to (\, E\,) \quad (4b)\; T \to \textit{identificador} \quad (4c)\; T \to \textit{número}$

$\}$

Ejemplo de eliminación de reglas unitarias

Reglas de $\mathcal{F}$

$\mathcal{P}'_4 = \{$

$(1)\ \mathcal{S} \rightarrow identificador = \mathcal{E}$

$\boxed{\mathcal{N}_{\mathcal{F}} = \{\mathcal{F}\}}$

$(2)\ \mathcal{E} \rightarrow \mathcal{E} + \mathcal{T}$

$(3a)\ \mathcal{E} \rightarrow \mathcal{T} * \mathcal{F}$

$(3b)\ \mathcal{E} \rightarrow (\ \mathcal{E}\ )$ $\quad (3c)\ \mathcal{E} \rightarrow identificador$ $\quad (3d)\ \mathcal{E} \rightarrow número$

$(4)\ \mathcal{T} \rightarrow \mathcal{T} * \mathcal{F}$

$(4a)\ \mathcal{T} \rightarrow (\ \mathcal{E}\ )$ $\quad (4b)\ \mathcal{T} \rightarrow identificador$ $\quad (4c)\ \mathcal{T} \rightarrow número$

$(6)\ \mathcal{F} \rightarrow (\ \mathcal{E}\ )$

$(7)\ \mathcal{F} \rightarrow identificador$

$(8)\ \mathcal{F} \rightarrow número$

$\}$

Eliminación de reglas unitarias

- **Observación**
  - Al eliminar las reglas unitarias, pueden surgir símbolos **inútiles**
  - Ejemplo: gramática sin épsilon

$\mathcal{P} = \{$

$S' \rightarrow \varepsilon \mid S$

$S \rightarrow A\,D \mid B$

$A \rightarrow C\,D\,E \mid D\,E \mid C\,D \mid D$

$B \rightarrow C\,E \mid E \mid C$

$C \rightarrow S \mid a$

$D \rightarrow A \mid b$

$E \rightarrow S \mid a$

$\}$

$\mathcal{N}_{S'} = \{S', S, B, C, E\}$

$\mathcal{N}_{S} = \{S, B, C, E\}$

$\mathcal{N}_{A} = \{A, D\}$

$\mathcal{N}_{B} = \{S, B, C, E\}$

$\mathcal{N}_{C} = \{S', S, B, C, E\}$

$\mathcal{N}_{D} = \{A, D\}$

$\mathcal{N}_{E} = \{S, B, C, E\}$

Eliminación de reglas unitarias

$\mathcal{P}' = \{$

$\mathcal{S}' \to \varepsilon \mid \mathcal{A}\,\mathcal{D} \mid \mathcal{C}\,\mathcal{E} \mid a \mid c$

$\mathcal{S} \to \mathcal{A}\,\mathcal{D} \mid \mathcal{C}\,\mathcal{E} \mid a \mid c$

$\mathcal{A} \to \mathcal{C}\,\mathcal{D}\,\mathcal{E} \mid \mathcal{D}\,\mathcal{E} \mid \mathcal{C}\,\mathcal{D} \mid b$

$\mathcal{B} \to \mathcal{C}\,\mathcal{E} \mid \mathcal{A}\,\mathcal{D} \mid a \mid c$

$\mathcal{C} \to \mathcal{A}\,\mathcal{D} \mid \mathcal{C}\,\mathcal{E} \mid a \mid c$

$\mathcal{D} \to \mathcal{C}\,\mathcal{D}\,\mathcal{E} \mid \mathcal{D}\,\mathcal{E} \mid \mathcal{C}\,\mathcal{D} \mid b$

$\mathcal{E} \to \mathcal{A}\,\mathcal{D} \mid \mathcal{C}\,\mathcal{E} \mid a \mid c$

$\}$

- **Observación**
  - Símbolos no accesibles:
  - $\mathcal{S}$ y $\mathcal{B}$
  - No están en la parte derecha de ninguna regla

RECURSIVIDAD POR LA IZQUIERDA

Gramática que genera ***asignaciones múltiples***

$\mathcal{P} = \{$

$(1)\ \mathcal{S} \rightarrow \mathcal{L}\ \mathcal{E}$

$(2)\ \mathcal{L} \rightarrow \mathcal{L}\ \textbf{identificador} =$

$(3)\ \mathcal{L} \rightarrow \textbf{identificador} =$

$(4)\ \mathcal{E} \rightarrow \mathcal{E} + \mathcal{T}$

...

$\}$

RECURSIVIDAD POR LA IZQUIERDA

Gramática que genera ***asignaciones múltiples***

$\mathcal{P} = \{$

$(1)\ \mathcal{S} \rightarrow \mathcal{L}\ \mathcal{E}$

$(2)\ \mathcal{L} \rightarrow \mathcal{L}\ \ \textbf{identificador} =$

$(3)\ \mathcal{L} \rightarrow\ \ \textbf{identificador} =$

$(4)\ \mathcal{E} \rightarrow \mathcal{E} + \mathcal{T}$

$\ldots$

$\}$

Ejemplo de derivación

$\mathcal{S} \Rightarrow_1 \mathcal{L}\ \mathcal{E}$

$\Rightarrow_2 \mathcal{L}\ \ \textbf{identificador} = \mathcal{E}$

$\Rightarrow_3 \textbf{identificador} =\ \ \textbf{identificador} = \mathcal{E}$

RECURSIVIDAD POR LA IZQUIERDA

- *Gramática que genera **lista de parámetros** de un procedimiento o función:*

$$
\begin{aligned}
&\mathcal{P} = \{\\
&\quad S \rightarrow \textbf{identificador} \, (\, L \,)\\
&\quad L \rightarrow L \, , \, \textbf{identificador}\\
&\quad L \rightarrow \textbf{identificador}\\
&\quad \ldots\\
&\}
\end{aligned}
$$

- *Gramática que genera componentes de un **array** de varias dimensiones:*

$$
\begin{aligned}
&\mathcal{P} = \{\\
&\quad S \rightarrow \mathit{identificador} \, D\\
&\quad D \rightarrow D \, [\, \textbf{número} \,]\\
&\quad D \rightarrow [\, \textbf{número} \,]\\
&\quad \ldots\\
&\}
\end{aligned}
$$

RECURSIVIDAD POR LA IZQUIERDA

- **Eliminación** de la recursividad **inmediata** por la izquierda
- **Eliminación** de la recursividad **general** por la izquierda

Eliminación de la recursividad por la izquierda

- *Entrada:* $\mathcal{G} = (\mathcal{V}_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}, \mathcal{S})$ *con recursividad inmediata por la izquierda*
- *Salida:* $\mathcal{G}' = (\mathcal{V}'_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}', \mathcal{S}')$ *sin reglas recursivas por la izquierda*

inicio
  $\mathcal{P}' \leftarrow \emptyset$
  para $\mathcal{A} \in \mathcal{V}_{\mathcal{N}}$ **hacer**
    si $\mathcal{A}$ *no tiene producciones recursivas*
      entonces $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathcal{A} \rightarrow \alpha \mid \mathcal{A} \rightarrow \alpha \in \mathcal{P}\}$
      si no
        si $\mathcal{A} \rightarrow \mathcal{A}\alpha_1 \mid \mathcal{A}\alpha_2 \mid \ldots \mid \mathcal{A}\alpha_p \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_q \in \mathcal{P}$
        donde $\forall i \in \{1, \ldots, p\}\ \alpha_i \neq \varepsilon \wedge \forall j \in \{1, \ldots, q\}\ \beta_j$ *no empieza por* $\mathcal{A}$
        entonces
          $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathcal{A} \rightarrow \beta_j, \mathcal{A} \rightarrow \beta_j \mathcal{A}' \mid \forall j \in \{1, \ldots, q\}$
            $\cup \{\mathcal{A}' \rightarrow \alpha_i, \mathcal{A}' \rightarrow \alpha_i \mathcal{A}' \mid \forall i \in \{1, \ldots, p\}\}$
          donde $\mathcal{A}'$ *es un nuevo símbolo no terminal*
        fin si
    fin si
  fin para
fin

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad inmediata

$\mathcal{P} = \{$

$\mathcal{S} \longrightarrow identificador = \mathcal{E}$

$\mathcal{E} \longrightarrow \mathcal{E} + \mathcal{T} \mid \mathcal{T} * \mathcal{F} \mid (\mathcal{E}) \mid identificador \mid número$

$\mathcal{T} \longrightarrow \mathcal{T} * \mathcal{F} \mid (\mathcal{E}) \mid identificador \mid número$

$\mathcal{F} \longrightarrow (\mathcal{E}) \mid identificador \mid número$

$\}$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad inmediata

Regla de S

$\mathcal{P}' = \{$

$S \rightarrow identificador = E$

$\}$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad inmediata

Reglas de E

$P' = \{$

$S \rightarrow identificador = E$

$E \rightarrow T * F \mid (E) \mid identificador \mid número \mid$

$T * F\ E' \mid (E)E' \mid identificador\ E' \mid número\ E'$

$E' \rightarrow +T \mid +TE'$

$\}$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad inmediata

Reglas de T

$P' = \{$

$S \rightarrow identificador = E$

$E \rightarrow T * F \mid (E) \mid identificador \mid número \mid$

$T * F\ E' \mid (E)E' \mid identificador\ E' \mid número\ E'$

$E' \rightarrow +T \mid +TE'$

$T \rightarrow (E) \mid identificador \mid número$

$(E)\ T' \mid identificador\ T' \mid número\ T'$

$T' \rightarrow *F \mid *FT'$

$\}$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad inmediata

Reglas de $\mathcal{F}$

$\mathcal{P}' = \{$

$\mathcal{S} \rightarrow identificador = \mathcal{E}$

$\mathcal{E} \rightarrow \mathcal{T} * \mathcal{F} \mid (\mathcal{E}) \mid identificador \mid número \mid$

$\mathcal{T} * \mathcal{F}\, \mathcal{E}' \mid (\mathcal{E})\mathcal{E}' \mid identificador\, \mathcal{E}' \mid número\, \mathcal{E}'$

$\mathcal{E}' \rightarrow + \mathcal{T} \mid + \mathcal{T}\mathcal{E}'$

$\mathcal{T} \rightarrow (\mathcal{E}) \mid identificador \mid número$

$(\mathcal{E})\, \mathcal{T}' \mid identificador\, \mathcal{T}' \mid número\, \mathcal{T}'$

$\mathcal{T}' \rightarrow * \mathcal{F} \mid * \mathcal{F}\mathcal{T}'$

$\mathcal{F} \rightarrow (\mathcal{E}) \mid identificador \mid número$

$\}$

Eliminación de la recursividad por la izquierda

Derivación con la gramática original

$$\mathcal{S} \Rightarrow identificador = \mathcal{E}$$

$$\Rightarrow identificador = \mathcal{E} + \mathcal{T}$$

$$\Rightarrow identificador = identificador + \mathcal{T}$$

$$\Rightarrow identificador = identificador + identificador$$

Derivación con la gramática sin recursividad por la izquierda

$$\mathcal{S} \Rightarrow identificador = \mathcal{E}$$

$$\Rightarrow identificador = identificador\ \mathcal{E}'$$

$$\Rightarrow identificador = identificador + \mathcal{T}$$

$$\Rightarrow identificador = identificador + identificador$$

Eliminación de la recursividad general por la izquierda

- *Entrada:* $G = (V_N, V_T, P, S)$ *gramática de contexto libre* ***propia****, es decir, sin ciclos, sin reglas* ***épsilon*** *ni símbolos inútiles.*
- *Salida:* $G' = (V'_N, V_T, P', S')$ ***sin*** *recursividad por la izquierda*

inicio

  $P' \leftarrow \emptyset$

  Ordénense los símbolos de $V_N = \{A_1, A_2, \ldots, A_n\}$

  para *i* ***de*** *1* ***a*** *n* ***hacer***

    para *j* ***de*** *1* ***a*** *i -1* ***hacer***

      si $A_i \rightarrow A_j \alpha \in P$

        entonces

          $P' \leftarrow P' \cup \{A_i \rightarrow \beta_k \alpha \mid A_j \rightarrow \beta_k$ *es una regla* ***actual*** *de* $A_j\}$

      fin si

    fin para

    Eliminar la recursividad ***inmediata*** *de* A_i

  fin para

fin

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

$\mathcal{P} = \{$

$(1)\ \mathcal{S} \rightarrow \mathcal{A}\ \mathcal{B}$

$(2)\ \mathcal{S} \rightarrow c$

$(3)\ \mathcal{A} \rightarrow \mathcal{B}\ b$

$(4)\ \mathcal{A} \rightarrow \mathcal{S}\ d$

$(5)\ \mathcal{A} \rightarrow a$

$(6)\ \mathcal{B} \rightarrow \mathcal{S}\ b$

$(7)\ \mathcal{B} \rightarrow \mathcal{A}\ a$

$\}$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

$P = \{$

$(1)\ S \rightarrow A\ B$

$(2)\ S \rightarrow c$

$(3)\ A \rightarrow B\ b$

$(4)\ A \rightarrow S\ d$

$(5)\ A \rightarrow a$

$(6)\ B \rightarrow S\ b$

$(7)\ B \rightarrow A\ a$

$\}$

- *Ordenamiento de los símbolos no terminales: {S, A, B}*

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

- *Paso **exterior** 1: reglas de S*

    - *Paso **interior** 1: S **no** tiene ninguna producción que comience por un símbolo con un número de orden inferior al suyo.*

    - *Eliminación de la recursividad inmediata: S **no** tiene recursividad inmediata por la izquierda.*

$\mathcal{P}' = \{$

$$S \rightarrow \mathcal{A}\,\mathcal{B} \mid c$$

$\}$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

- *Paso **exterior** 2: reglas de* A
  - *Paso **interior** 1: la regla número (4)* $A \rightarrow S\ d$ *se sustituye por las reglas* $A \rightarrow A\ B\ d \mid c\ d$

    quedando las reglas de A *de la siguiente forma:*

    $$A \rightarrow A\ B\ d \mid B\ b \mid c\ d \mid a$$

  - *Eliminación de la recursividad inmediata:*

    $$A \rightarrow B\ b \mid c\ d \mid a \mid B\ b\ A' \mid c\ d\ A' \mid a\ A'$$

    $$A' \rightarrow B\ d \mid B\ d\ A'$$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

$\mathcal{P}' = \{$

$S \longrightarrow \mathcal{A}\,\mathcal{B} \mid c$

$\mathcal{A} \longrightarrow \mathcal{B}\, b \mid c\, d \mid a \mid \mathcal{B}\, b\, \mathcal{A}' \mid c\, d\, \mathcal{A}' \mid a\, \mathcal{A}'$

$\mathcal{A}' \longrightarrow \mathcal{B}\, d \mid \mathcal{B}\, d\, \mathcal{A}'$

$\}$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

- *Paso **exterior** 3: reglas de B*
  - *Paso **interior** 1: la regla (6) $B \rightarrow S\ b$ se sustituye por las reglas*

$$B \rightarrow A\ B\ b \mid c\ b$$

quedando las reglas de B de la siguiente forma:

$$B \rightarrow A\ B\ b \mid c\ b \mid A\ a$$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

- *Paso **exterior** 3: reglas de B*
  - *Paso **interior** 2: la regla $B \rightarrow A\ B\ b$ se sustituye por las reglas*

$$B \rightarrow B\ b\ B\ b \mid c\ d\ B\ b \mid a\ B\ b \mid$$
$$B\ b\ A'\ B\ b \mid c\ d\ A'\ B\ b \mid a\ A'\ B\ b$$

y la regla $B \rightarrow A\ a$ se sustituye por las reglas

$$B \rightarrow B\ b\ a \mid c\ d\ a \mid a\ a \mid B\ b\ A'\ a \mid c\ d\ A'\ a \mid a\ A'\ a$$

quedando las reglas de B de la siguiente forma:

$$B \rightarrow B\ b\ B\ b \mid B\ b\ A'\ B\ b \mid B\ b\ a \mid B\ b\ A'\ a \mid$$
$$c\ d\ B\ b \mid a\ B\ b \mid c\ d\ A'\ B\ b \mid a\ A'\ B\ b \mid$$
$$b\ a \mid c\ d\ a \mid a\ a \mid c\ d\ A'\ a \mid a\ A'\ a \mid$$
$$c\ b$$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

- *Paso **exterior** 3: reglas de B*
  - *Eliminación de la recursividad inmediata:*

$$
\begin{aligned}
B \rightarrow\ & c\ d\ B\ b \mid a\ B\ b \mid c\ d\ A'\ B\ b \mid a\ A'\ B\ b \mid \\
& b\ a \mid c\ d\ a \mid a\ a \mid c\ d\ A'\ a \mid a\ A'\ a \mid \\
& c\ b \mid \\
& c\ d\ B\ b\ B' \mid a\ B\ b\ B' \mid c\ d\ A'\ B\ b\ B' \mid a\ A'\ B\ b\ B' \mid \\
& b\ a\ B' \mid c\ d\ a\ B' \mid a\ a\ B' \mid c\ d\ A'\ a\ B' \mid a\ A'\ a\ B' \mid \\
& c\ b\ B' \\
B' \rightarrow\ & b\ B\ b \mid b\ A'\ B\ b \mid b\ a \mid b\ A'\ a \mid \\
& b\ B\ b\ B' \mid b\ A'\ B\ b\ B' \mid b\ a\ B' \mid b\ A'\ a\ B'
\end{aligned}
$$

Eliminación de la recursividad por la izquierda

Ejemplo de **eliminación** de la recursividad general

$\mathcal{P}' = \{$

$S \rightarrow \mathcal{A}\,\mathcal{B} \mid c$

$\mathcal{A} \rightarrow \mathcal{B}\,b \mid c\,d \mid a \mid \mathcal{B}\,b\,\mathcal{A}' \mid c\,d\,\mathcal{A}' \mid a\,\mathcal{A}'$

$\mathcal{A}' \rightarrow \mathcal{B}\,d \mid \mathcal{B}\,d\,\mathcal{A}'$

$\mathcal{B} \rightarrow c\,d\,\mathcal{B}\,b \mid a\,\mathcal{B}\,b \mid c\,d\,\mathcal{A}'\,\mathcal{B}\,b \mid a\,\mathcal{A}'\,\mathcal{B}\,b \mid b\,a \mid c\,d\,a \mid a\,a \mid c\,d$
$\mathcal{A}'\,a \mid a\,\mathcal{A}'\,a \mid c\,b \mid c\,d\,\mathcal{B}\,b\,\mathcal{B}' \mid a\,\mathcal{B}\,b\,\mathcal{B}' \mid c\,d\,\mathcal{A}'\,\mathcal{B}\,b\,\mathcal{B}' \mid$
$a\,\mathcal{A}'\,\mathcal{B}\,b\,\mathcal{B}' \mid b\,a\,\mathcal{B}' \mid c\,d\,a\,\mathcal{B}' \mid a\,a\,\mathcal{B}' \mid c\,d\,\mathcal{A}'\,a\,\mathcal{B}' \mid$
$a\,\mathcal{A}'\,a\,\mathcal{B}' \mid c\,b\,\mathcal{B}'$

$\mathcal{B}' \rightarrow b\,\mathcal{B}\,b \mid b\,\mathcal{A}'\,\mathcal{B}\,b \mid b\,a \mid b\,\mathcal{A}'\,a \mid b\,\mathcal{B}\,b\,\mathcal{B}' \mid b\,\mathcal{A}'\,\mathcal{B}\,b\,\mathcal{B}' \mid b\,a\,\mathcal{B}' \mid$
$b\,\mathcal{A}'\,a\,\mathcal{B}'$

$\}$

Factorización por la izquierda

- Considérense las reglas

$$S \rightarrow \textbf{si}\ E\ \textbf{entonces}\ S\ \ \textbf{si no}\ S\ \textbf{fin si} \mid \textbf{si}\ E\ \textbf{entonces}\ S\ \ \textbf{fin si}$$

- Al leer el componente léxico **si** no se sabe aún qué regla elegir para expandir la sentencia S

- En general,
  - Si A tiene dos reglas de la forma $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$
  - y la cadena a analizar comienza por una cadena no vacía derivada a partir de α, no se sabe si expandir A a $\alpha\beta_1$ o a $\alpha\beta_2$
  - Por tanto, se hace la siguiente **transformación**

$$A \rightarrow \alpha A'$$
$$A' \rightarrow \beta_1 \mid \beta_2$$

Factorización por la izquierda

- *Entrada:* $\mathcal{G} = (\mathcal{V}_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}, \mathcal{S})$ *gramática de contexto libre propia*
- *Salida:* $\mathcal{G}' = (\mathcal{V}'_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}', \mathcal{S}')$ *gramática factorizada por la izquierda*

inicio

  para $(\mathcal{A} \in \mathcal{V}_{\mathcal{N}})$ ***hacer***

    mientras *$\mathcal{A}$ tenga dos reglas actuales con el mismo prefijo* ***hacer***

      si α *es el* ***prefijo más largo*** *de dos o más alternativas de $\mathcal{A}$ y* $\alpha \neq \varepsilon$

        entonces *sustituir todas las reglas*

        $\mathcal{A} \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \ldots. \mid \alpha\beta_p \mid \gamma_1 \mid \ldots \mid \gamma_q$

        donde $\forall\, i \in \{1, \ldots q\}$ γ_i ***no*** *empieza por* α

        por las reglas $\mathcal{A} \rightarrow \alpha\mathcal{A}' \mid \gamma_1 \mid \ldots \mid \gamma_q$

        $\mathcal{A}' \rightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_p$

      fin si

    fin mientras

  fin para

fin

Factorización por la izquierda

Ejemplo

$$\mathcal{P} = \{$$

$$\mathcal{S} \rightarrow \mathcal{A}\,\mathcal{B}\,c \mid \mathcal{A}\,\mathcal{B}\,d\,e \mid \mathcal{A}\,\mathcal{B}\,d\,f \mid \mathcal{A}\,\mathcal{B}\,\mathcal{S}$$

$$\mathcal{A} \rightarrow a$$

$$\mathcal{B} \rightarrow b$$

$$\}$$

Factorización por la izquierda

Ejemplo

- *Reglas de S*

$$S \rightarrow A\,B\,c \mid \underline{A\,B\,d}\,e \mid \underline{A\,B\,d}\,f \mid A\,B\,S$$

  - $\alpha_1 = A\,B\,d$ *es el prefijo más largo de dos reglas de S.*
  - *Las reglas de S se sustituyen por las siguientes:*

$$S \rightarrow \underline{A\,B}\,d\,S' \mid \underline{A\,B}\,c \mid \underline{A\,B}\,S$$

$$S' \rightarrow e \mid f$$

  - $\alpha_2 = A\,B$ *es ahora el prefijo más largo de tres producciones actuales de S.*
  - *Por tanto, las reglas de S se sustituyen por las siguientes:*

$$S \rightarrow A\,B\,S''$$

$$S'' \rightarrow d\,S' \mid c \mid S$$

$$S' \rightarrow e \mid f$$

- *Las reglas de A y B no requieren ser factorizadas*

Eliminación de la recursividad inmediata y factorización por la izquierda

- *Entrada:* $\mathcal{G} = (\mathcal{V}_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}, \mathcal{S})$ *con recursividad inmediata por la izquierda*
- *Salida:* $\mathcal{G}' = (\mathcal{V}'_{\mathcal{N}}, \mathcal{V}_{\mathcal{T}}, \mathcal{P}', \mathcal{S}')$ ***sin*** *reglas recursivas por la izquierda y factorizada por la izquierda*

inicio

$\mathcal{P}' \leftarrow \emptyset$

para $\mathcal{A} \in \mathcal{V}_{\mathcal{N}}$ ***hacer***

si $\mathcal{A}$ *no tiene producciones recursivas*

entonces $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathcal{A} \rightarrow \alpha \mid \mathcal{A} \rightarrow \alpha \in \mathcal{P}\}$

si no ***si*** $\mathcal{A} \rightarrow \mathcal{A}\alpha_1 \mid \mathcal{A}\alpha_2 \mid \ldots \mid \mathcal{A}\alpha_p \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_q \in \mathcal{P}$

donde $\forall i \in \{1, \ldots, p\}\ \alpha_i \neq \varepsilon \wedge \forall j \in \{1, \ldots, q\}\ \beta_j$ *no empieza por* $\mathcal{A}$

entonces $\mathcal{P}' \leftarrow \mathcal{P}' \cup \{\mathcal{A} \rightarrow \beta_j \mathcal{A}' \mid \forall j \in \{1, \ldots, q\}\}$

$\cup \{\mathcal{A}' \rightarrow \alpha_i \mathcal{A}', \mathcal{A}' \rightarrow \varepsilon \mid \forall i \in \{1, \ldots, p\}\}$

donde $\mathcal{A}'$ *es un nuevo símbolo no terminal*

fin si

fin si

fin para

fin

Eliminación de la recursividad inmediata y factorización por la izquierda

Ejemplo: gramática sin reglas unitarias

$\mathcal{P} = \{$

$\mathcal{S} \longrightarrow identificador = \mathcal{E}$

$\mathcal{E} \longrightarrow \mathcal{E} + \mathcal{T} \;|\; \mathcal{T} * \mathcal{F} \;|\; (\mathcal{E}) \;|\; identificador \;|\; número$

$\mathcal{T} \longrightarrow \mathcal{T} * \mathcal{F} \;|\; (\mathcal{E}) \;|\; identificador \;|\; número$

$\mathcal{F} \longrightarrow (\mathcal{E}) \;|\; identificador \;|\; número$

$\}$

Eliminación de la recursividad inmediata y factorización por la izquierda

Ejemplo: gramática factorizada y sin recursividad inmediata por la izquierda

$$
\begin{aligned}
\mathcal{P}' = \{ & \\
& S \rightarrow identificador = E \\
& E \rightarrow T * F E' \mid (E) E' \mid identificador\ E' \mid número\ E' \\
& E' \rightarrow + T E' \mid \varepsilon \\
& T \rightarrow (E) T' \mid identificador\ T' \mid número\ T' \\
& T' \rightarrow * F T' \mid \varepsilon \\
& F \rightarrow (E) \mid identificador \mid número \\
\} &
\end{aligned}
$$

Formas normales de las gramáticas de contexto libre

- **Forma normal de Chomsky: FNC**

$$\mathcal{A} \longrightarrow \mathcal{B}\ \mathcal{C}$$

$$\mathcal{A} \longrightarrow a$$

donde $\mathcal{A}, \mathcal{B}, \mathcal{C} \in \mathcal{V}_{\mathcal{N}}$ *y* $\boldsymbol{a} \in \mathcal{V}_{\mathcal{T}}$

- **Forma normal de Greibach: FNG**

$$\mathcal{A} \longrightarrow a\,\alpha$$

donde $\alpha \in \mathcal{V}_{\mathcal{N}}^{*}$

Forma normal de Chomsky

$\mathcal{P} = \{$

$(1)\ S \rightarrow A\ B$

$(2)\ A \rightarrow A\ B$

$(3)\ A \rightarrow a$

$(4)\ B \rightarrow B\ B$

$(5)\ B \rightarrow b$

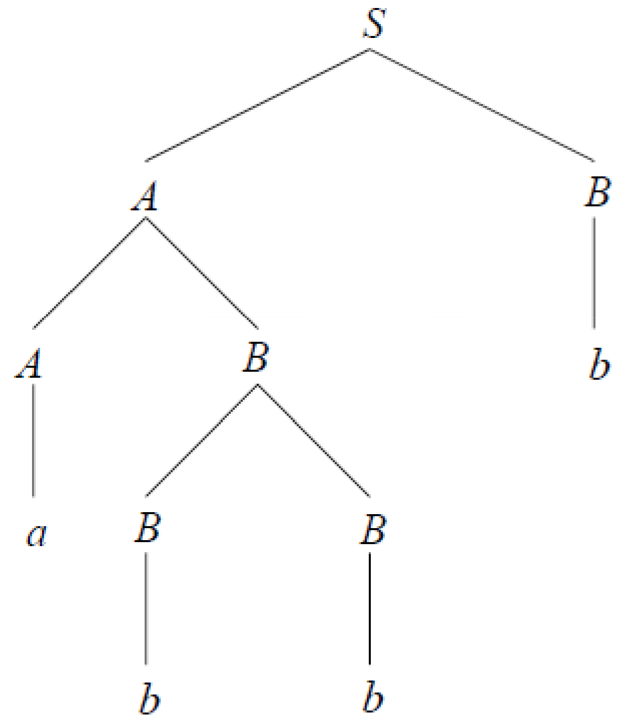
$\}$

$S \Rightarrow_1 A\ B \Rightarrow_2 A\ B\ B$

$\Rightarrow_3 a\ B\ B \Rightarrow_4 a\ B\ B\ B$

$\Rightarrow_5 a\ b\ B\ B \Rightarrow_5 a\ b\ b\ B \Rightarrow_6 a\ b\ b\ b$

Árbol binario



Forma normal de Chomsky

Algoritmo

- *Entrada:* $G = (V_N, V_T, P, S)$ *gramática de contexto libre propia*
- *Salida:* $G'' = (V''_N, V_T, P'', S')$ *gramática en la forma normal de Chomsky*
- *Paso 1: generación* $G' = (V'_N, V_T, P', S)$

  donde las reglas de P' son de la forma

  $$A \rightarrow B_1 B_2 \ldots B_k \quad \text{donde } k \geq 2$$

  $$A \rightarrow a$$

  - $L(G') = L(G) - \varepsilon$
- *Paso 2:*
  - *Generación de G'' que estará en la forma normal de Chomsky*
  - $L(G'') = L(G) - \varepsilon$

Forma normal de Chomsky

- *Paso 1: generación* $G' = (V'_N, V_T, P', S)$
  - *Si* $A \rightarrow X_1 X_2 \ldots X_k \in P$
    - *Si* $k = 1$ *entonces* $A \rightarrow X\_1$ *se añade a* P'
    - *Si* $k \geq 2$ *entonces* $A \rightarrow B_1 B_2 \ldots B_k$ *se añade a* P'

    donde

    - *si* $X_i \in V_N$ *entonces* $B_i = X_i$
    - *si* $X_i = a_i \in V_T$ *entonces* B_i *es un nuevo símbolo* ***no terminal*** *y se también añade la regla* $B_i \rightarrow a$ P'
  - *Se verifica que* $L(G') = L(G) - \varepsilon$

Forma normal de Chomsky

- *Paso 2: generación* $G'' = (V''_N, V_T, P'', S)$
  - Si $A \to a \in P'$ entonces $A \to a \in P''$
  - Si $A \to B_1 B_2 \in P'$ entonces $A \to B_1 B_2 \in P''$
  - Si $A \to B_1 B_2 \ldots B_k \in P'$ y $k > 2$

    entonces

    $A \to B_1 C_1$

    $C_1 \to B_2 C_2$

    …

    $C_{k-3} \to B_{k-2} C_{k-2}$

    $C_{k-2} \to B_{k-1} B_k$
  - Se verifica que $L(G'') = L(G) - \varepsilon$

Forma normal de Chomsky

Ejemplo

$\mathcal{P} = \{$

$\mathcal{S} \longrightarrow a\,\mathcal{A}\,\mathcal{B}$

$\mathcal{A} \longrightarrow a\,\mathcal{B}\,b$

$\mathcal{A} \longrightarrow a$

$\mathcal{B} \longrightarrow b\,b$

$\}$

Forma normal de Chomsky

Ejemplo

$\mathcal{P} = \{$

$\mathcal{S} \longrightarrow a\,\mathcal{A}\,\mathcal{B}$

$\mathcal{A} \longrightarrow a\,\mathcal{B}\,b$

$\mathcal{A} \longrightarrow a$

$\mathcal{B} \longrightarrow b\,b$

$\}$

Paso 1:

$\mathcal{P}' = \{$

$\mathcal{S} \longrightarrow \mathcal{B}_1\,\mathcal{A}\,\mathcal{B}$

$\mathcal{A} \longrightarrow \mathcal{B}_1\,\mathcal{B}\,\mathcal{B}_2$

$\mathcal{A} \longrightarrow a$

$\mathcal{B} \longrightarrow \mathcal{B}_2\,\mathcal{B}_2$

$\mathcal{B}_1 \longrightarrow a$

$\mathcal{B}_2 \longrightarrow b$

$\}$

Forma normal de Chomsky

Ejemplo

$\mathcal{P} = \{$

$S \rightarrow a\, A\, B$

$A \rightarrow a\, B\, b$

$A \rightarrow a$

$B \rightarrow b\, b$

$\}$

Paso 1:

$\mathcal{P}' = \{$

$S \rightarrow B_1\, A\, B$

$A \rightarrow B_1\, B\, B_2$

$A \rightarrow a$

$B \rightarrow B_2\, B_2$

$B_1 \rightarrow a$

$B_2 \rightarrow b$

$\}$

Paso 2:

$\mathcal{P}'' = \{$

$S \rightarrow B_1\, C_1$

$C_1 \rightarrow A\, B$

$A \rightarrow B_1\, C_2$

$C_2 \rightarrow B\, B_2$

$A \rightarrow a$

$B \rightarrow B_2 B_2$

$B_1 \rightarrow a$

$B_2 \rightarrow b$

$\}$

Forma normal de Greibach

- Reglas de la forma $\mathcal{A} \rightarrow a\,\alpha$ donde $\alpha \in \mathcal{V}_{\mathcal{N}}^{*}$

- **Algoritmo**
  - **Paso 1:** aplicación del algoritmo que elimina la recursividad **general** por la izquierda.
  - **Paso 2**: transformación de las reglas de los símbolos **no terminales de la gramática original.**
  - **Paso 3**: transformación de las reglas de los símbolos **no terminales** obtenidos al eliminar la **recursividad inmediata** por la izquierda.

Forma normal de Greibach

- **Entrada**: G gramática en la forma normal de Chomsky
- **Paso 1:** eliminación de la recursividad **general** por la izquierda.
  - Se generan reglas de producción de la forma:

$$\mathcal{A}_i \rightarrow \mathcal{A}_j\ \gamma\ \ \forall\ j > i \wedge i, j \in \{1,2,\ \ldots, n\}$$

$$\mathcal{A}_i \rightarrow a\gamma$$

$$\mathcal{A}'_i \rightarrow \gamma$$

$$donde\ \mathcal{A}_i, \mathcal{A}_j \in \mathcal{V}_{\mathcal{N}},\ a \in \mathcal{V}_{\mathcal{T}}$$

  - $\forall i \in \{1,2,\ldots,n\}, \mathcal{A}'_i$ ha sido generado al eliminar la recursividad **inmediata** por la izquierda de $\mathcal{A}_i$
  - $\gamma \in (\mathcal{V}_{\mathcal{N}} \cup \{\mathcal{A}'_1, \ldots, \mathcal{A}'_n\})^*$.
  - En particular, la reglas del símbolo $\mathcal{A}_n$ serán de la forma $\mathcal{A}_n \rightarrow a\gamma$ y ya estarán en la forma normal de Greibach.

Forma normal de Greibach

- **Paso 2:** transformación de las reglas de los símbolos **no terminales** de la gramática **original**

[1] inicio

[2] para i de n-1 a 1 hacer

[3] para j de i+1 a n hacer

[4] para cada regla actual de la forma $A_i \rightarrow A_j \gamma$ *hacer*

[5] si $A_j \rightarrow a_1 \alpha_1 | a_2 \alpha_2 | \ldots | a_p \alpha_p$ *son las reglas actuales de* A_j

[6] entonces $A_i \rightarrow a_1 \alpha_1 \gamma | a_2 \alpha_2 \gamma | \ldots | a_p \alpha_p \gamma$

[7] pasan a ser reglas actuales de de A_i

[8] fin si

[9] fin para

[10] fin para

[12] fin para

[13] fin

Forma normal de Greibach

- **Paso 2:**
  - o Al finalizar el paso 2, las reglas obtenidas serán de la forma

$$\mathcal{A}_i \rightarrow a\gamma, \forall \mathcal{A}_i \in \mathcal{V}_{\mathcal{N}}$$
$$\mathcal{A}'_i \rightarrow \gamma$$

o En particular, la reglas de los símbolos originales ya estarán en la forma normal de **Greibach**

Forma normal de Greibach

- **Paso 3:** transformación de las reglas de los símbolos **no terminales** obtenidos al eliminar la recursividad **inmediata**

[1] inicio

[2] para i de 1 a m hacer

[3] para j de 1 a n hacer

[4] para cada regla actual de la forma $A'_i \rightarrow A_j\gamma$ hacer

[5] si $A_j \rightarrow a_1\alpha_1 | a_2\ \alpha_2 | \ldots | a_p\alpha_p$ son las reglas actuales de A_j

[6] entonces $A'_i \rightarrow a_1\alpha_1\gamma\ | a_2\ \alpha_2\gamma\ | \ldots | a_p\alpha_p\gamma$

[7] pasan a ser reglas actuales de de A'_i

[8] fin si

[9] fin para

[10] fin para

[12] fin para

[13] fin

Forma normal de Greibach

- **Paso 3:**
  - Al finalizar el paso 3, las reglas obtenidas serán de la forma

$$\mathcal{A}_i \rightarrow a\gamma\,,\ \forall \mathcal{A}_i \in \mathcal{V}_\mathcal{N}$$

$$\mathcal{A}'_i \rightarrow a\gamma$$

  - En particular, todas la reglas estarán en la forma normal de **Greibach**

Forma normal de Greibach

Ejemplo

- Entrada: gramática en la forma normal de Chomsky

$$\mathcal{P} = \{$$

$$(1)\ \mathcal{S} \longrightarrow \mathcal{A}\ \mathcal{B}$$

$$(2)\ \mathcal{A} \longrightarrow \mathcal{S}\ \mathcal{B}$$

$$(3)\ \mathcal{A} \longrightarrow a$$

$$(4)\ \mathcal{B} \longrightarrow \mathcal{B}\ \mathcal{A}$$

$$(5)\ \mathcal{B} \longrightarrow d$$

$$\}$$

Forma normal de Greibach

Ejemplo

- Paso 1: aplicación del algoritmo de eliminación de la recursividad general
  - o *Reglas de S*
    - ▪ *La regla de S no necesita transformarse*

$$S \rightarrow A\ B$$

Forma normal de Greibach

Ejemplo

- Paso 1: aplicación del algoritmo de eliminación de la recursividad general
  - *Reglas de* $\mathcal{A}$
    - *Se sustituye* $\mathcal{S}$ *por su alternativa en la regla* $\mathcal{A} \rightarrow \mathcal{S}\ \mathcal{B}$ *generando la regla:* $\mathcal{A} \rightarrow \mathcal{A}\ \mathcal{B}\ \mathcal{B}$
    - *Eliminación de la recursividad por la izquierda de las reglas de* $\mathcal{A}$.

Las reglas actuales de $\mathcal{A}$ *son* $\mathcal{A} \rightarrow \mathcal{A}\ \mathcal{B}\ \mathcal{B} \mid a$

y las reglas que se generan al eliminar la recursividad inmediata son:

$$\mathcal{A} \rightarrow a \mid a\ \mathcal{A}'$$

$$\mathcal{A}' \rightarrow \mathcal{B}\ \mathcal{B} \mid \mathcal{B}\ \mathcal{B}\ \mathcal{A}'$$

Forma normal de Greibach

Ejemplo

- Paso 1: aplicación del algoritmo de eliminación de la recursividad general
  - *Reglas de $\mathcal{B}$*
    - *No se tiene que sustituir ningún símbolo en las reglas de $\mathcal{B}$.*
    - *Eliminación de la recursividad por la izquierda de las reglas de $\mathcal{B}$.*

      Las reglas actuales de $\mathcal{B}$ son $\mathcal{B} \longrightarrow \mathcal{B}\,\mathcal{A} \mid d$

      y las reglas que se generan al eliminar la recursividad inmendiata son:

      $\mathcal{B} \longrightarrow d \mid d\,\mathcal{B}'$

      $\mathcal{B}' \longrightarrow \mathcal{A} \mid \mathcal{A}\,\mathcal{B}'$

Forma normal de Greibach

Ejemplo

- Paso 1: gramática generada

$\mathcal{P}_1 = \{$

$$\mathcal{S} \longrightarrow \mathcal{A}\,\mathcal{B}$$

$$\mathcal{A} \longrightarrow a \mid a\,\mathcal{A}'$$

$$\mathcal{B} \longrightarrow d \mid d\,\mathcal{B}'$$

$$\mathcal{A}' \longrightarrow \mathcal{B}\,\mathcal{B} \mid \mathcal{B}\,\mathcal{B}\,\mathcal{A}'$$

$$\mathcal{B}' \longrightarrow \mathcal{A} \mid \mathcal{A}\,\mathcal{B}'$$

$\}$

Forma normal de Greibach

Ejemplo

- Paso 2:
  - *Las reglas de $\mathcal{B}$ ya están en la forma normal de Greibach.*
  - *Las reglas de $\mathcal{A}$ ya están en la forma normal de Greibach.*
  - *Se sustituye $\mathcal{A}$ por sus alternativas en la regla $\mathcal{S} \rightarrow \mathcal{A}\ \mathcal{B}$ generando las siguientes reglas de $\mathcal{S}$:*

$$\mathcal{S} \rightarrow a\,\mathcal{B} \mid a\,\mathcal{A}'\,\mathcal{B}$$

Forma normal de Greibach

Ejemplo

- Paso 2: gramática generada

$\mathcal{P}_2 = \{$

$$\mathcal{S} \to a\,\mathcal{B} \mid a\,\mathcal{A}'\,\mathcal{B}$$

$$\mathcal{A} \to a \mid a\,\mathcal{A}'$$

$$\mathcal{B} \to d \mid d\,\mathcal{B}'$$

$$\mathcal{A}' \to \mathcal{B}\,\mathcal{B} \mid \mathcal{B}\,\mathcal{B}\,\mathcal{A}'$$

$$\mathcal{B}' \to \mathcal{A} \mid \mathcal{A}\,\mathcal{B}'$$

$\}$

Forma normal de Greibach

Ejemplo

- Paso 3:
  - o *Trasformación de las reglas de A':* $A' \rightarrow B\,B \mid B\,B\,A'$

    Se sustituye B por sus alternativas, generándose las siguientes reglas de A'

    $$A' \rightarrow d\,B \mid d\,B'\,B \mid d\,B\,A' \mid d\,B'\,B\,A'$$

  - o *Transformación de las reglas de B':* $B' \rightarrow A \mid A\,B'$

    Se sustituye A por sus alternativas, generándose las siguientes reglas de B'

    $$B' \rightarrow a \mid a\,A' \mid a\,B' \mid a\,A'\,B'$$

Forma normal de Greibach

Ejemplo

- Paso 3: gramática generada en la forma normal de **Greibach**

$$\mathcal{P}_3 = \{$$

$$\mathcal{S} \rightarrow a\,\mathcal{B} \mid a\,\mathcal{A}'\mathcal{B}$$

$$\mathcal{A} \rightarrow a \mid a\,\mathcal{A}'$$

$$\mathcal{B} \rightarrow d \mid d\,\mathcal{B}'$$

$$\mathcal{A}' \rightarrow d\,\mathcal{B} \mid d\,\mathcal{B}'\mathcal{B} \mid d\,\mathcal{B}\,\mathcal{A}' \mid d\,\mathcal{B}'\mathcal{B}\,\mathcal{A}'$$

$$\mathcal{B}' \rightarrow a \mid a\,\mathcal{A}' \mid a\,\mathcal{B}' \mid a\,\mathcal{A}'\mathcal{B}'$$

$$\}$$

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 8.- Gramáticas de Contexto Libre