

# **Análisis Sintáctico Ascendente**

Precedencia de Operadores

# Generalidades

Son un caso especial de Análisis Sintáctico  
Ascendente o de Desplazamiento y Reducción

Características:

- Ningún lado derecho (cuerpo) de la producción es  $\epsilon$ .
- No tiene dos No Terminales adyacentes.

Ejemplo: La GLC siguiente es de operadores:

$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid E^E \mid (E) \mid id$

# Generalidades

Se establecen tres *relaciones de precedencia* :

$$<\cdot, \doteq, \cdot>$$

Precedencia de operadores

Si  $a$  y  $b$  son símbolos terminales: operadores o cualquier otro símbolo, se tienen las siguientes *relaciones de precedencia*:

Relación	Significado
$a <\cdot b$	$a$ "cede la precedencia a " $b$
$a \doteq b$	$a$ "tiene la misma precedencia que " $b$
$a \cdot > b$	$a$ "tiene más precedencia que " $b$

# Uso de Relaciones de Precedencia

La GIC siguiente es de operadores:

$E \rightarrow E + E \mid E * E \mid \text{id}$

*Solo se trabaja con*

Se dará por sentada la siguiente tabla de Precedencia de Operadores:

a

b

IZ4

der

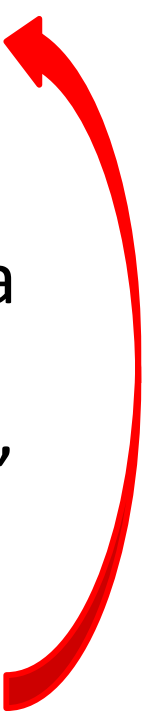
<div>a \ b</div>	+	*	id	\$
+	.>	<.	<.	.>
*	.>	.>	<.	.>
id	.>	.>		.>
\$	<.	<.	<.	

*orden de aparición en la gramática*

# Uso de Relaciones de Precedencia

Dada una cadena  $w$  se usará el siguiente procedimiento para hallar los mangos

Pasos:

1. Se inserta la cadena entre los símbolos  $\$$ :  $\$w\$$ .
  2. Se insertan las relaciones de precedencia entre los símbolos que conforman  $\$w\$$ , según la tabla de precedencia.
  3. Recorrer la cadena generada en 2 desde la izquierda hasta encontrar el primer  $\cdot>$ .
  4. Devolverse hacia la izquierda hasta encontrar un  $<\cdot$ , sobrepasando los  $\doteq$ .
  5. El mango se encuentra a la derecha del  $<\cdot$  y a la izquierda del  $\cdot>$ .
- 

# Uso de Relaciones de Precedencia

$\begin{matrix} a \backslash b \\ \text{Isq} \end{matrix}$	+	*	id	\$
+	$\cdot >$	$< \cdot$	$< \cdot$	$\cdot >$
*	$\cdot >$	$\cdot >$	$< \cdot$	$\cdot >$
id	$\cdot >$	$\cdot >$		$\cdot >$
\$	$< \cdot$	$< \cdot$	$< \cdot$	

## Pasos:

1. Se inserta la cadena entre los símbolos \$: \$w\$.
2. Se insertan las relaciones de precedencia entre los símbolos que conforman \$w\$, según la tabla de precedencia.
3. Recorrer la cadena generada en 2 desde la izquierda hasta encontrar el primer  $\cdot >$ .
4. Devolverse hacia la izquierda hasta encontrar un  $< \cdot$ , sobrepasando los  $\doteq$ . *el primer  $< \cdot$*
5. El **mango** se encuentra a la derecha del  $< \cdot$  y a la izquierda del  $\cdot >$ .

## Cadena a reconocer:

w=id+id\*id

- Paso 1
- Paso 2
- Paso 3
- Paso 4
- Paso 5

\$id+id\*id\$

\$<\cdot id\cdot>+<\cdot id\cdot>\*<\cdot id\cdot>\$

\$<\cdot id\cdot>+<\cdot id\cdot>\*<\cdot id\cdot>\$

\$<\cdot id\cdot>+<\cdot id\cdot>\*<\cdot id\cdot>\$

\$<\cdot id\cdot>+<\cdot id\cdot>\*<\cdot id\cdot>\$

Paso 3

mango

$\$ < \cdot id \cdot > + < \cdot id \cdot > * < \cdot id \cdot > \$$   
 $\$ < \cdot id \cdot > + < \cdot id \cdot > * < \cdot id \cdot > \$$   
 $\$ < \cdot id \cdot > + < \cdot id \cdot > * < \cdot id \cdot > \$$   
 $\$ \times + id \times id \$$

$\$ < + < id > * < id > \$$   
 $\$ < + < id > * < id > \$$

# Uso de Relaciones de Precedencia

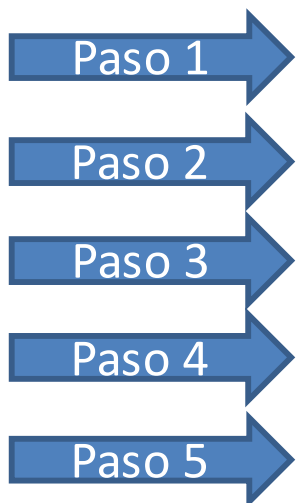
$a \backslash b$	+	*	id	\$
+	$\cdot >$	$< \cdot$	$< \cdot$	$\cdot >$
*	$\cdot >$	$\cdot >$	$< \cdot$	$\cdot >$
id	$\cdot >$	$\cdot >$		$\cdot >$
\$	$< \cdot$	$< \cdot$	$< \cdot$	

## Pasos:

1. Se inserta la cadena entre los símbolos \$:  $\$w\$$ .
2. Se insertan las relaciones de precedencia entre los símbolos que conforman  $\$w\$$ , según la tabla de precedencia.
3. Recorrer la cadena generada en 2 desde la izquierda hasta encontrar el primer  $\cdot >$ .
4. Devolverse hacia la izquierda hasta encontrar un  $< \cdot$ , sobrepasando los  $\doteq$ .
5. El **mango** se encuentra a la derecha del  $< \cdot$  y a la izquierda del  $\cdot >$ .

## Cadena a reconocer:

$w = \text{id} + \text{id} * \text{id}$



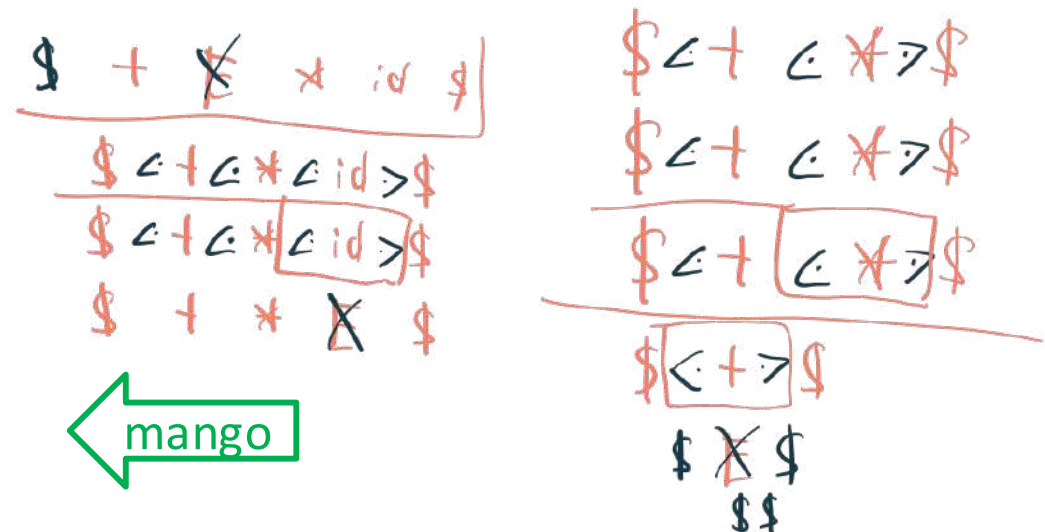
$\$ + \text{id} * \text{id} \$$

$\$ < \cdot + < \cdot \text{id} \cdot > * < \cdot \text{id} \cdot > \$$

$\$ < \cdot + < \cdot \text{id} \cdot > * < \cdot \text{id} \cdot > \$$

$\$ < \cdot + < \cdot \text{id} \cdot > * < \cdot \text{id} \cdot > \$$

$\$ < \cdot + < \cdot \text{id} \cdot > * < \cdot \text{id} \cdot > \$$



# Uso de Relaciones de Precedencia

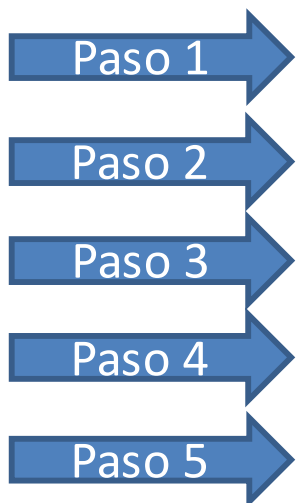
$a \backslash b$	+	*	id	\$
+	$\cdot >$	$< \cdot$	$< \cdot$	$\cdot >$
*	$\cdot >$	$\cdot >$	$< \cdot$	$\cdot >$
id	$\cdot >$	$\cdot >$		$\cdot >$
\$	$< \cdot$	$< \cdot$	$< \cdot$	

## Pasos:

1. Se inserta la cadena entre los símbolos \$:  $\$w\$$ .
2. Se insertan las relaciones de precedencia entre los símbolos que conforman  $\$w\$$ , según la tabla de precedencia.
3. Recorrer la cadena generada en 2 desde la izquierda hasta encontrar el primer  $\cdot >$ .
4. Devolverse hacia la izquierda hasta encontrar un  $< \cdot$ , sobrepasando los  $\doteq$ .
5. El **mango** se encuentra a la derecha del  $< \cdot$  y a la izquierda del  $\cdot >$ .

## Cadena a reconocer:

$w = \text{id} + \text{id} * \text{id}$



$\$+*id\$$

$\$<\cdot + <\cdot * <\cdot id \cdot >\$$

$\$<\cdot + <\cdot * <\cdot id \cdot >\$$

$\$<\cdot + <\cdot * <\cdot id \cdot >\$$

$\$<\cdot + <\cdot * <\cdot id \cdot >\$$

mango



# Uso de Relaciones de Precedencia

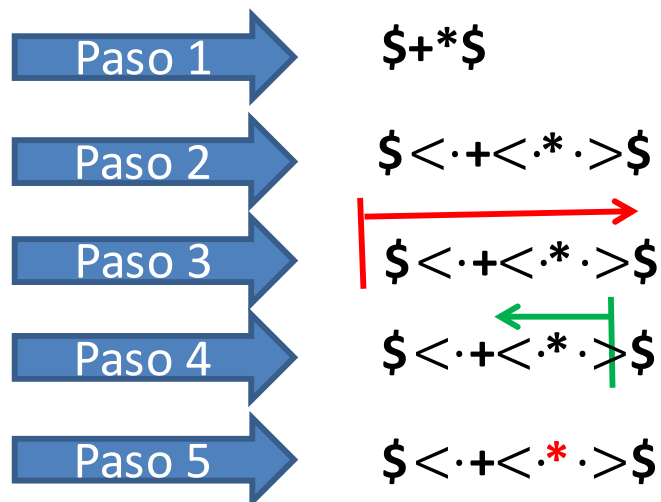
$a \backslash b$	+	*	id	\$
+	$\cdot >$	$< \cdot$	$< \cdot$	$\cdot >$
*	$\cdot >$	$\cdot >$	$< \cdot$	$\cdot >$
id	$\cdot >$	$\cdot >$		$\cdot >$
\$	$< \cdot$	$< \cdot$	$< \cdot$	

## Pasos:

1. Se inserta la cadena entre los símbolos \$:  $\$w\$$ .
2. Se insertan las relaciones de precedencia entre los símbolos que conforman  $\$w\$$ , según la tabla de precedencia.
3. Recorrer la cadena generada en 2 desde la izquierda hasta encontrar el primer  $\cdot >$ .
4. Devolverse hacia la izquierda hasta encontrar un  $< \cdot$ , sobrepasando los  $\doteq$ .
5. El **mango** se encuentra a la derecha del  $< \cdot$  y a la izquierda del  $\cdot >$ .

## Cadena a reconocer:

$w = \text{id} + \text{id} * \text{id}$



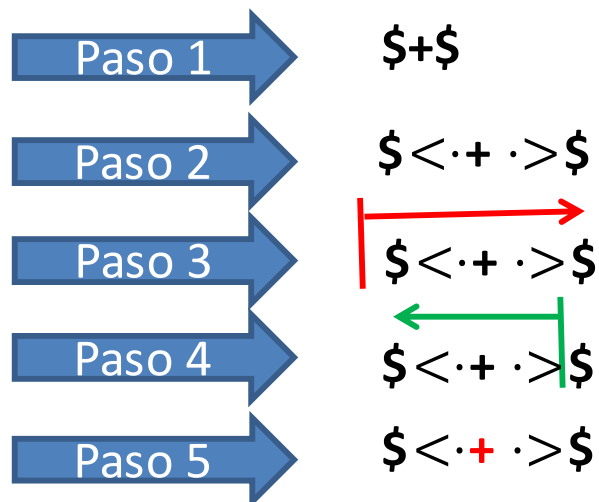
mango

# Uso de Relaciones de Precedencia

$a \backslash b$	+	*	id	\$
+	$\cdot >$	$< \cdot$	$< \cdot$	$\cdot >$
*	$\cdot >$	$\cdot >$	$< \cdot$	$\cdot >$
id	$\cdot >$	$\cdot >$		$\cdot >$
\$	$< \cdot$	$< \cdot$	$< \cdot$	

**Cadena a reconocer:**

$w = \text{id} + \text{id} * \text{id}$



**Pasos:**

1. Se inserta la cadena entre los símbolos  $\$$ :  $\$w\$$ .
2. Se insertan las relaciones de precedencia entre los símbolos que conforman  $\$w\$$ , según la tabla de precedencia.
3. Recorrer la cadena generada en 2 desde la izquierda hasta encontrar el primer  $\cdot >$ .
4. Devolverse hacia la izquierda hasta encontrar un  $< \cdot$ , sobrepasando los  $\doteq$ .
5. El **mango** se encuentra a la derecha del  $< \cdot$  y a la izquierda del  $\cdot >$ .

**Conclusión:**

En una forma de frase, todo **mango** se debe delimitar entre los símbolos  $< \cdot$  y  $\cdot >$ .

**mango**

# Establecer Relaciones de Precedencia

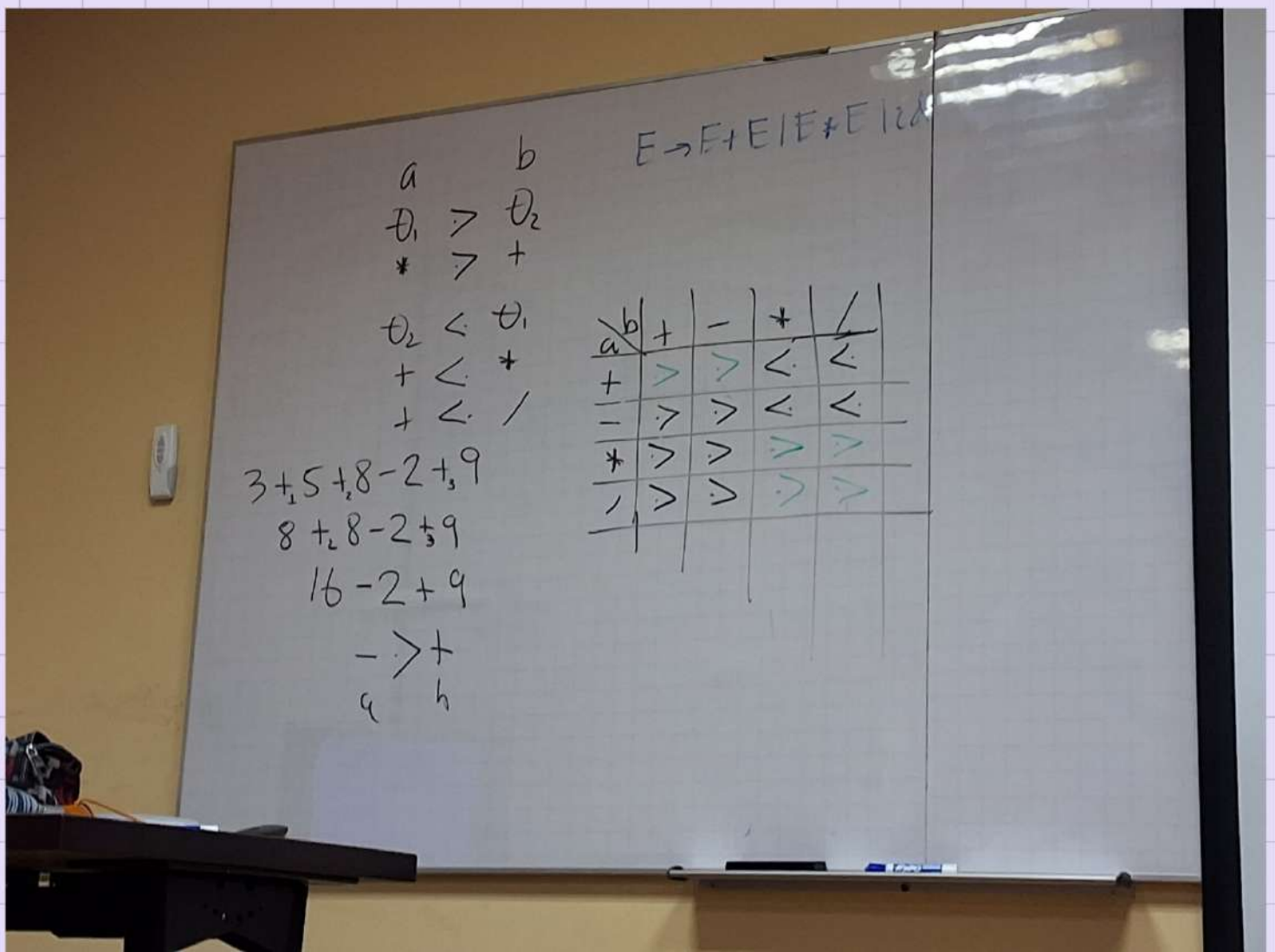
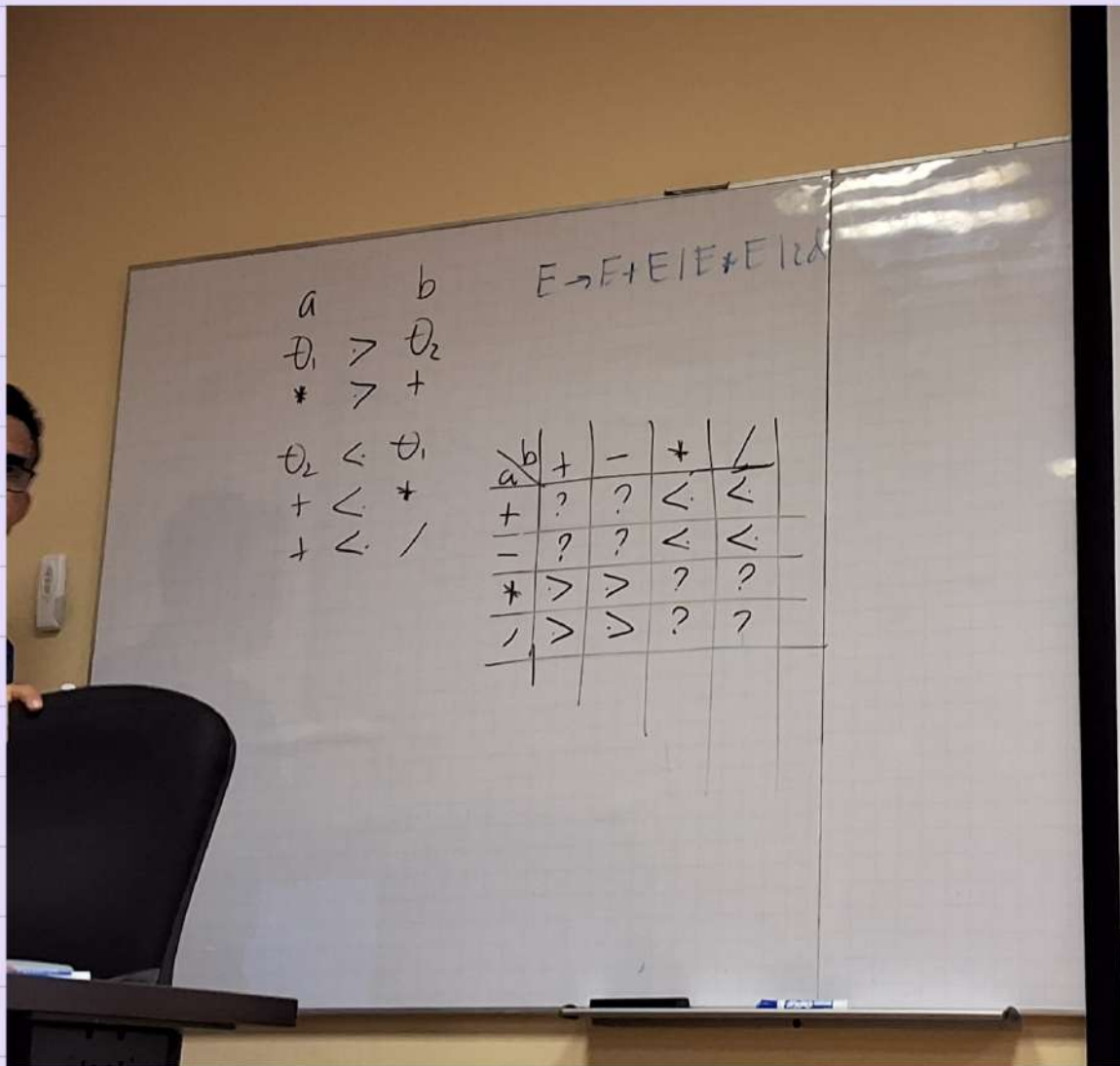
Se definen las siguientes reglas para seleccionar mangos apropiados

Pasos:

1. Si el operador  $\theta_1$  tiene mayor precedencia que el operador  $\theta_2$ , hágase  $\theta_1 \cdot > \theta_2$  y  $\theta_2 < \cdot \theta_1$ .

**Ejemplo:** Si  $*$  tiene mayor precedencia que  $+$ , hágase  $* \cdot > +$  y  $+ < \cdot *$

Todo mango se encuentra entre  $<$  y  $>$





# Establecer Relaciones de Precedencia

2. Si  $\theta_1$  y  $\theta_2$  son operadores de igual precedencia (pueden ser el mismo operador) y asociativos por izquierda, entonces hágase  $\theta_1 \cdot > \theta_2$  y  $\theta_2 \cdot > \theta_1$ .  
Si  $\theta_1$  y  $\theta_2$  son operadores de igual precedencia (pueden ser el mismo operador) y asociativos por derecha, entonces hágase  $\theta_1 < \cdot \theta_2$  y  $\theta_2 < \cdot \theta_1$ .

**Ejemplo:** Si  $+$  y  $-$  son asociativos por la izquierda, hágase  $+\cdot > +$ ,  $+\cdot > -$ ,  $-\cdot > -$  y  $-\cdot > +$

**Ejemplo:** Si  $^{\wedge}$  es asociativo por la derecha, hágase  $^{\wedge} < \cdot ^{\wedge}$

# Establecer Relaciones de Precedencia

3. Hágase

**Ejemplo<sub>1</sub>:** Reconocer **id**, luego de Der. md.:

$E \Rightarrow \text{id}$







# Establecer Relaciones de Precedencia

3. Hágase  
\$<·id  
id·>\$

## Conclusión:

En una forma de frase, todo **mango** se debe delimitar entre los símbolos <· y ·>.

Derivación más derecha:  $E \Rightarrow \underline{id}$

**Ejemplo<sub>1</sub>:** Derivando hacia atrás:

$E \Rightarrow \$ \textcolor{red}{< \cdot id \cdot} \textcolor{red}{>} \$$

**Cuál es el mango?**

# Establecer Relaciones de Precedencia

3. Hágase  
     $\$ < \cdot id$   
     $id \cdot > \$$

**Ejemplo<sub>2</sub>:** Reconocer **id+id**, luego de Der. md.:

$E \Rightarrow E+E \Rightarrow E+id \Rightarrow id+id$

# Establecer Relaciones de Precedencia

3. Hágase  
\$<·id  
id·>\$  
id·>θ

## Conclusión:

En una forma de frase, todo **mango** se debe delimitar entre los símbolos <· y ·>.

Derivación más derecha:  $E \Rightarrow E+E \Rightarrow E+id \Rightarrow \underline{id+id}$

**Ejemplo<sub>2</sub>:** Derivando hacia atrás:

$\Rightarrow \$<\underline{id}\cdot>+ \quad id \quad \$$     **Cuál es el mango?**  
θ<sub>1</sub>

# Establecer Relaciones de Precedencia

3. Hágase  
\$<·id  
id ·> \$  
id ·> θ  
θ <·id

## Conclusión:

En una forma de frase, todo **mango** se debe delimitar entre los símbolos <· y ·>.

Derivación más derecha:  $E \Rightarrow E+E \Rightarrow \underline{E+id} \Rightarrow id+id$

**Ejemplo<sub>2</sub>:** Derivando hacia atrás:

$\Rightarrow \$ E + < \cdot id \cdot > \$$

$\Rightarrow \$ < \cdot id \cdot > + \quad id \quad \$$

Cuál es el mango?

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta$

$\theta < \cdot id$

$\$ < \cdot \theta$

$\theta \cdot > \$$

**Conclusión:**

En una forma de frase, todo **mango** se debe delimitar entre los símbolos  $< \cdot$  y  $\cdot >$ .

Derivación más derecha:  $E \Rightarrow \underline{E+E} \Rightarrow E+id \Rightarrow id+id$

**Ejemplo<sub>2</sub>:** Derivando hacia atrás:

$\Rightarrow \$ \langle \overset{\theta_1}{E} \cdot + \cdot E \rangle \$$

$\Rightarrow \$ \quad + \langle \cdot id \cdot \rangle \$$

$\Rightarrow \$ \langle \cdot id \cdot \rangle + \quad id \quad \$$

Cuál es el mango?

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta$

$\theta < \cdot id$

$\$ < \cdot \theta$

$\theta \cdot > \$$

**Ejemplo<sub>3</sub>:** Reconocer  **$id*(id+id)+id$** , luego de

Der. md.:  $E \Rightarrow E+E \Rightarrow E+id \Rightarrow E^*E+id \Rightarrow E^*(E)+id \Rightarrow$

$E^*(E+E)+id \Rightarrow E^*(E+id)+id \Rightarrow E^*(id+id)+id \Rightarrow$

$id*(id+id)+id$

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta$

$\theta < \cdot id$

$\$ < \cdot \theta$

$\theta \cdot > \$$

**Conclusión:**

En una forma de frase, todo **mango** se debe delimitar entre los símbolos  $< \cdot$  y  $\cdot >$ .

Derivación más derecha:  $E \Rightarrow E+E \Rightarrow E+id \Rightarrow E^*E+id \Rightarrow E^*(E)+id \Rightarrow E^*(E+E)+id \Rightarrow$   
 $E^*(E+id)+id \Rightarrow E^*(id+id)+id \Rightarrow \underline{id^*(id+id)+id}$

**Ejemplo<sub>3</sub>:** Derivando hacia atrás:

$\Rightarrow \$ < \cdot id \cdot > * ( id + id ) + id \$$  **Cuál es el mango?**

Ya están resueltos, no aporta una nueva regla

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta$       $( < \cdot id$

$\theta < \cdot id$

$\$ < \cdot \theta$

$\theta \cdot > \$$

**Conclusión:**

En una forma de frase, todo **mango** se debe delimitar entre los símbolos  $< \cdot$  y  $\cdot >$ .

Derivación más derecha:  $E \Rightarrow E+E \Rightarrow E+id \Rightarrow E^*E+id \Rightarrow E^*(E)+id \Rightarrow E^*(E+E)+id \Rightarrow$   
 $E^*(E+id)+id \Rightarrow \underline{E^*(id+id)+id} \Rightarrow id^*(id+id)+id$

**Ejemplo<sub>3</sub>:** Derivando hacia atrás:

$\Rightarrow \$ \cancel{E} * ( < \cdot \cancel{id} \cdot > + id ) + id \$$      Cuál es el mango?

$\Rightarrow \$ < \cdot \underline{id} \cdot > * ( id + id ) + id \$$



# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta \quad (< \cdot id$

$\theta < \cdot id \quad id \cdot > )$

$\$ < \cdot \theta$

$\theta \cdot > \$$

**Conclusión:**

En una forma de frase, todo **mango** se debe delimitar entre los símbolos  $< \cdot$  y  $\cdot >$ .

Derivación más derecha:  $E \Rightarrow E+E \Rightarrow E+id \Rightarrow E^*E+id \Rightarrow E^*(E)+id \Rightarrow E^*(E+E)+id \Rightarrow$   
 $E^*(E+id)+id \Rightarrow E^*(id+id)+id \Rightarrow id^*(id+id)+id$

**Ejemplo<sub>3</sub>:** Derivando hacia atrás:

$\Rightarrow \$ E * ( \cancel{E} + \underline{< \cdot id \cdot >} ) + id \$$

$\Rightarrow \$ * ( \underline{< \cdot id \cdot >} + id ) + id \$$

$\Rightarrow \$ \underline{< \cdot id \cdot >} * ( id + id ) + id \$$

Cuál es el mango?

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta \quad (< \cdot id$

$\theta < \cdot id \quad id \cdot > )$

$\$ < \cdot \theta \quad (< \cdot \theta$

$\theta \cdot > \$ \quad \theta \cdot > )$

**Conclusión:**

En una forma de frase, todo **mango** se debe delimitar entre los símbolos  $< \cdot$  y  $\cdot >$ .

Derivación más derecha:  $E \Rightarrow E+E \Rightarrow E+id \Rightarrow E^*E+id \Rightarrow E^*(E)+id \Rightarrow \underline{E^*(E+E)+id} \Rightarrow E^*(E+id)+id \Rightarrow E^*(id+id)+id \Rightarrow id^*(id+id)+id$

**Ejemplo<sub>3</sub>:** Derivando hacia atrás:

$\Rightarrow \$ E * ( \overset{E+E}{\underline{E \cdot + \cdot E}} ) + id \$$

$\Rightarrow \$ * ( + < \cdot \underline{id} \cdot > ) + id \$$

$\Rightarrow \$ * ( < \cdot \underline{id} \cdot > + id ) + id \$$

$\Rightarrow \$ < \cdot \underline{id} \cdot > * ( id + id ) + id \$$

Cuál es el mango?

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta$      $( < \cdot id$      $\theta < \cdot ($

$\theta < \cdot id$      $id \cdot > )$      $) \cdot > \theta$

$\$ < \cdot \theta$      $( < \cdot \theta$      $( \doteq )$

$\theta \cdot > \$$      $\theta \cdot > )$

**Conclusión:**

En una forma de frase, todo **mango** se debe delimitar entre los símbolos  $< \cdot$  y  $\cdot >$ .

$\$ < ($      $) > )$   
 $) > \$$   
 $( < ($

Derivación más derecha:  $E \Rightarrow E+E \Rightarrow E+id \Rightarrow E * E+id \Rightarrow \underline{E * (E)+id} \Rightarrow E * (E+E)+id \Rightarrow$   
 $E * (E+id)+id \Rightarrow E * (id+id)+id \Rightarrow id * (id+id)+id$

**Ejemplo<sub>3</sub>:** Derivando hacia atrás:

$\Rightarrow \$ E * < \cdot ( \underline{E} ) \cdot > + id \$$

$\Rightarrow \$ * ( < \cdot \underline{+} \cdot > ) + id \$$

$\Rightarrow \$ * ( + < \cdot \underline{id} \cdot > ) + id \$$

$\Rightarrow \$ * ( < \cdot \underline{id} \cdot > + id ) + id \$$

**Cuál es el mango?**

como ambos salen a la vez  
 tienen la misma  
 precedencia

$\theta < ( \overset{E}{\doteq} ) > \theta$

$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

$\begin{matrix} a & b \\ ) & ( \end{matrix}$

$(id + id) * id$

$\backslash b$	$)$	$($	$id$	
$\$$	X			No Se puede $) ($
$id$		X		
$)$		X	X	

$( ( ( \quad ) > )$

$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

$\begin{matrix} a & b \\ ) & ( \end{matrix}$

$\begin{matrix} ) & ) \\ \$ & ( \end{matrix}$

$\begin{matrix} id & ( \end{matrix}$

$\begin{matrix} ( & ) \\ ( & ) \end{matrix}$

$\begin{matrix} < & ) \\ ) & - id \end{matrix}$

$\begin{matrix} > & ) \\ > & \$ \end{matrix}$

$(id + id) * id$

$id * (id + id)$

$id + ((id + id) * id)$

$( \quad \$$

$\$ id * id \quad X \quad \$$

$\$ < ($

$( < ($

$) > )$

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta$

$\theta < \cdot id$

$\$ < \cdot \theta$

$\theta \cdot > \$$

$( < \cdot id$

$id \cdot > )$

$( < \cdot \theta$

$\theta \cdot > )$

**Conclusión:**

En una forma de frase, todo **mango** se debe delimitar entre los símbolos  $< \cdot$  y  $\cdot >$ .

$\theta < \cdot ($

$) \cdot > \theta$

$( \equiv )$

$\$ < \cdot ($

$) \cdot > \$$

$( < \cdot ($

$) \cdot > )$

**Ejemplo<sub>4</sub>:** Cuáles serían las relaciones de precedencia de:

$\$ < \cdot ($        $( < \cdot ($

$) \cdot > \$$        $) \cdot > )$

# Tabla de Precedencia de Operadores

Para la GIC siguiente de operadores, obtener la tabla de precedencia de operadores:

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$a \backslash b$	+	*	(	)	id	\$
+	$\rightarrow$	$\leftarrow$	$\leftarrow$	$\rightarrow$	$\leftarrow$	$\rightarrow$
*	$\rightarrow$	$\rightarrow$	$\leftarrow$	$\rightarrow$	$\leftarrow$	$\rightarrow$
(	$\leftarrow$	$\leftarrow$	$\leftarrow$	$\doteq$	$\leftarrow$	$\times$
)	$\rightarrow$	$\rightarrow$	$\times$	$\rightarrow$	$\times$	$\rightarrow$
id	$\rightarrow$	$\rightarrow$	$\times$	$\rightarrow$	$\times$	$\rightarrow$
\$	$\leftarrow$	$\leftarrow$	$\leftarrow$	$\times$	$\leftarrow$	$\times$

$\$ < \cdot id$   
 $id \cdot > \$$   
 $id \cdot > \theta$   
 $\theta < \cdot id$   
 $\$ < \cdot \theta$   
 $\theta \cdot > \$$

$(< \cdot id$   
 $id \cdot > )$   
 $(< \cdot \theta$   
 $\theta \cdot > )$   
 $\theta < \cdot ($   
 $\theta < \cdot ($   
 $) \cdot > \theta$

$( \doteq )$   
 $\$ < \cdot ($   
 $) \cdot > \$$   
 $(< \cdot ($   
 $) \cdot > )$

Revisar el Ejemplo 4.28 del libro guía (pag 213). NO considerar el -E.

Construir la Tabla de Precedencia sin -E y comparar con el ejemplo:

$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid E \wedge E \mid (E) \mid id$

# Tabla de Precedencia de Operadores

Para la GIC siguiente de operadores, obtener la tabla de precedencia de operadores:

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$a \backslash b$	+	*	(	)	id	\$
+	$\cdot >$	$< \cdot$	$< \cdot$	$\cdot >$	$< \cdot$	$\cdot >$
*	$\cdot >$	$\cdot >$	$< \cdot$	$\cdot >$	$< \cdot$	$\cdot >$
(	$< \cdot$	$< \cdot$	$< \cdot$	$\doteq$	$< \cdot$	
)	$\cdot >$	$\cdot >$		$\cdot >$		$\cdot >$
id	$\cdot >$	$\cdot >$		$\cdot >$		$\cdot >$
\$	$< \cdot$	$< \cdot$	$< \cdot$		$< \cdot$	

$\$ < \cdot id$   
 $id \cdot > \$$   
 $id \cdot > \theta$   
 $\theta < \cdot id$   
 $\$ < \cdot \theta$   
 $\theta \cdot > \$$   
 $( < \cdot id$   
 $id \cdot > )$   
 $( < \cdot \theta$   
 $\theta \cdot > )$   
 $( < \cdot ($   
 $\theta < \cdot ($   
 $) \cdot > \theta$   
 $( \doteq )$   
 $\$ < \cdot ($   
 $) \cdot > \$$   
 $( < \cdot ($   
 $) \cdot > )$

Revisar el Ejemplo 4.28 del libro guía (pag 213). NO considerar el -E.

Construir la Tabla de Precedencia sin -E y comparar con el ejemplo:

$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid E \wedge E \mid (E) \mid id$

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```
(1)  apuntar ae al primer simbolo de w$;  
(2)  repeat forever  
(3)    if $ está en la cima de la pila y ae apunta a $ then  
(4)      return  
(5)    else begin  
(6)      sea a el simbolo terminal más a la cima de la pila y  
(7)      sea b el simbolo apuntado por ae;  
(8)      if  $a < \cdot b$  o  $a \doteq b$  then begin  
(9)        meter b en la pila;  
(10)       avanzar ae al siguiente simbolo de entrada;  
(11)     end;  
(12)    else if  $a \cdot > b$  then      /* reduce */  
(13)      repeat  
(14)        extraer el elemento de la cima de la pila  
(15)      until el terminal de la cima de la pila esté relacionado por  $< \cdot$   
(16)        con el terminal más recientemente extraído de la pila.  
(17)    else error ()  
(18)  end
```



# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1)  apuntar ae al primer símbolo de w$;
(2)  repeat forever
(3)      if $ está en la cima de la pila y ae apunta a $ then
(4)          return
(5)      else begin
(6)          sea a el símbolo terminal más a la cima de la pila y
(7)          sea b el símbolo apuntado por ae;
(8)          if  $a < b$  o  $a \doteq b$  then begin
(9)              meter b en la pila;
(10)             avanzar ae al siguiente símbolo de entrada;
(11)         end;
(12)         else if  $a > b$  then          /* reduce */
(13)             repeat
(14)                 extraer el elemento de la cima de la pila
(15)                 until el terminal de la cima de la pila esté relacionado por  $<$ 
(16)                     con el terminal más recientemente extraído de la pila.
(17)             else error ()
(18)         end

```

$a \backslash b$	+	*	(	)	id	\$
+	$\cdot >$	$< \cdot$	$< \cdot$	$\cdot >$	$< \cdot$	$\cdot >$
*	$\cdot >$	$\cdot >$	$< \cdot$	$\cdot >$	$< \cdot$	$\cdot >$
(	$< \cdot$	$< \cdot$	$< \cdot$	$\equiv$	$< \cdot$	
)	$\cdot >$	$\cdot >$		$\cdot >$		$\cdot >$
id	$\cdot >$	$\cdot >$		$\cdot >$		$\cdot >$
\$	$< \cdot$	$< \cdot$	$< \cdot$		$< \cdot$	

## GIC de operadores:

$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

**w= id \* (id+id)+id**

[illegible]

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return accept
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a < b o a = b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a > b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

$a \backslash b$	+	*	(	)	id	\$
+	.>	<.	<.	.>	<.	.>
*	.>	.>	<.	.>	<.	.>
(	<.	<.	<.	=	<.	
)	.>	.>		.>		.>
id	.>	.>		.>		.>
\$	<.	<.	<.		<.	

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$E	(id+id)+id\$	
\$E*	(id+id)+id\$	
\$E*(id)	+id\$	
\$E*(id+)	id\$	
\$E*(id+id)	)\$	
\$E*(id+id+)	id\$	
\$E*(id+id+id)	\$	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar

<i>a</i> \ <i>b</i>	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar

$\begin{smallmatrix} a \backslash b \\ \end{smallmatrix}$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar

$\begin{smallmatrix} a \backslash b \\ \hline \end{smallmatrix}$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir

$\begin{smallmatrix} a \backslash b \\ \hline \end{smallmatrix}$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	



# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar

$\begin{smallmatrix} a \backslash b \\ \hline \end{smallmatrix}$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar

$\begin{smallmatrix} a \backslash b \\ \end{smallmatrix}$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	



# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	) + id\$	reducir

$\begin{smallmatrix} a \backslash b \\ \end{smallmatrix}$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)id\$	reducir
\$*(+	)id\$	reducir

$\begin{smallmatrix} a \backslash b \\ \end{smallmatrix}$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)id\$	reducir
\$*(+id	)id\$	reducir
\$*(+id	)id\$	desplazar

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)id\$	reducir
\$*(+	)id\$	reducir
\$*(	)id\$	desplazar
\$*( <u>)</u>	+id\$	reducir

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$(+	id)+id\$	desplazar
\$(+id	)id\$	reducir
\$(+)	)id\$	reducir
\$*(	)id\$	desplazar
\$*()	+id\$	reducir
\$* > temp	+id\$	reducir

$\begin{smallmatrix} a \backslash b \\ \end{smallmatrix}$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1)  apuntar ae al primer símbolo de w$;
(2)  repeat forever
(3)      if $ está en la cima de la pila y ae apunta a $ then
(4)          return
(5)      else begin
(6)          sea a el símbolo terminal más a la cima de la pila y
(7)          sea b el símbolo apuntado por ae;
(8)          if  $a < b$  o  $a \doteq b$  then begin
(9)              meter b en la pila;
(10)             avanzar ae al siguiente símbolo de entrada;
(11)         end;
(12)         else if  $a > b$  then          /* reduce */
(13)             repeat
(14)                 extraer el elemento de la cima de la pila
(15)                 until el terminal de la cima de la pila esté relacionado por  $<$ 
(16)                     con el terminal más recientemente extraído de la pila.
(17)             else error ()
(18)         end

```

$a \backslash b$	+	*	(	)	id	\$
+	.>	<.	<.	.>	<.	.>
*	.>	.>	<.	.>	<.	.>
(	<.	<.	<.	$\equiv$	<.	
)	.>	.>		.>		.>
id	.>	.>		.>		.>
\$	<.	<.	<.		<.	

## GIC de operadores:

$$E \rightarrow E + E \mid E * E \mid (E) \mid id$$

**w= id \* (id+id)+id**

Pila	Entrada	Acción
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	) +id\$	reducir
\$*(+	) +id\$	reducir
\$*(	) +id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$E	+id\$	desplazar



# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```



(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)id\$	reducir
\$*(+	)id\$	reducir
\$*(	)id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$	+id\$	desplazar
\$+  	id\$	desplazar

$a \backslash b$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a < b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a > b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <
(16)          con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)id\$	reducir
\$*(+id	)id\$	reducir
\$*(	)id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$	+id\$	desplazar
\$+	id\$	desplazar
\$+id > Teore	\$	reducir

$\begin{smallmatrix} a \backslash b \\ \end{smallmatrix}$	+	*	(	)	id	\$
+	.>	<.	<.	.>	<.	.>
*	.>	.>	<.	.>	<.	.>
(	<.	<.	<.	≡	<.	
)	.>	.>		.>		.>
id	.>	.>		.>		.>
\$	<.	<.	<.		<.	



# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$*(+id	id)+id\$	desplazar
\$*(+id	) + id\$	reducir
\$*(+id	) + id\$	reducir
\$*(+id	) + id\$	desplazar
\$*(+id	+ id\$	reducir
\$*(+id	+ id\$	reducir
\$*(+id	+ id\$	desplazar
\$*(+id	id\$	desplazar
\$*(+id	\$	reducir
\$*(+id	\$	reducir

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a < b o a ≡ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a · > b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

**GIC de operadores:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

$w = id * (id + id) + id$

Pila	Entrada	Acción
\$*(+id	) + id\$	reducir
\$*(+id	) + id\$	reducir
\$*(	) + id\$	desplazar
\$*(	) + id\$	reducir
\$*	) + id\$	reducir
\$	) + id\$	desplazar
\$+	) + id\$	desplazar
\$+id	) + id\$	reducir
\$+	) + id\$	reducir
\$	) + id\$	aceptar

$a \backslash b$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	≡	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

```

(1) apuntar ae al primer símbolo de w$;
(2) repeat forever
(3)   if $ está en la cima de la pila y ae apunta a $ then
(4)     return
(5)   else begin
(6)     sea a el símbolo terminal más a la cima de la pila y
(7)     sea b el símbolo apuntado por ae;
(8)     if a <· b o a ÷ b then begin
(9)       meter b en la pila;
(10)      avanzar ae al siguiente símbolo de entrada;
(11)    end;
(12)    else if a ·> b then /* reduce */
(13)      repeat
(14)        extraer el elemento de la cima de la pila
(15)        until el terminal de la cima de la pila esté relacionado por <·
(16)        con el terminal más recientemente extraído de la pila.
(17)      else error ()
(18)    end

```

$a \backslash b$	+	*	(	)	id	\$
+	·>	<·	<·	·>	<·	·>
*	·>	·>	<·	·>	<·	·>
(	<·	<·	<·	÷	<·	
)	·>	·>		·>		·>
id	·>	·>		·>		·>
\$	<·	<·	<·		<·	

**GIC:**

$E \rightarrow E + E \mid E * E \mid (E) \mid id$

**w = id \* (id+id)+id**

Pila	Entrada	Acción
\$	id*(id+id)+id\$	desplazar
\$id	*(id+id)+id\$	reducir
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)id\$	reducir
\$*(+id	)id\$	reducir
\$*(	)id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$	+id\$	desplazar
\$+	id\$	desplazar
\$+id	\$	reducir
\$+	\$	reducir
\$	\$	aceptar

# Ejercicio

Dada la siguiente **GIC**:

$$S \rightarrow (L) \mid a$$

$$L \rightarrow L,S \mid S$$

- Obtener la tabla de precedencia de operadores (Recuerde utilizar derivación más derecha al revés).
- Reconocer la cadena  $(a,(a,a))$ .

# Ejercicio

Dada la siguiente **GIC**:

$S \rightarrow (L) \mid a$

$L \rightarrow L,S \mid S$

$W = a \quad S \rightarrow a \quad \$ \prec a \succ \$$

	(	)	<i>a</i>	,	\$
(					
)					
<i>a</i>					$\succ$
,					
\$			$\prec$		

$$W = (a, a)$$

$$S \Rightarrow (a, a)$$

$$S \Rightarrow (L) \Rightarrow (L, S) \Rightarrow (L, a)$$

$$\Rightarrow (S, a) \Rightarrow (a, a)$$

$$\$(a \rightarrow a)\$$$

$$\$(L) \Rightarrow \$$$

a/b	(	)	a	\$
(		=	<	<
)				>
a		>		>
,		>	<	>
\$	<			

Se ignora el no terminal

Se ignora el terminal

$$S \Rightarrow (L) \Rightarrow (S) \Rightarrow (L) \Rightarrow (S) \Rightarrow (a)$$

# cuando el mango es un no terminal se ignora y se salta a la siguiente derivación

$$(a) \Rightarrow \$$$

$$a) \Rightarrow \$$$

$$\$($$

a/b	(	)	a	\$
(	<	=	<	<
)		>		>
a		>		>
,	<	>	<	>
\$	<		<	

$$W = (a, a, a)$$

$$S \Rightarrow (L) \Rightarrow (LS) \Rightarrow (L, a)$$

$$\Rightarrow (L^x S^x, a) \Rightarrow (L^x a, a) \Rightarrow (LS, a, a) \Rightarrow (L^x a, a, a)$$

$$(L, a, a)$$