

# 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 GIC siguiente es de operadores:

$$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid E^{\wedge} E \mid (E) \mid \text{id}$$

# Generalidades

Se establecen tres *relaciones de precedencia* :

$$<\cdot, \dot{=}, \cdot>$$

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" $b$
$a \dot{=} b$	$a$ "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:

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

Solo se trabaja  
con

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

a \ b	a	b	*	id	\$
a	Izq	dere			
+	>	<.	<.	>	
*	>	>	>	<.	>
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  $\div$ .
5. El **mango** se encuentra a la derecha del  $< \cdot$  y a la izquierda del  $\cdot >$ .

# Uso de Relaciones de Precedencia

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

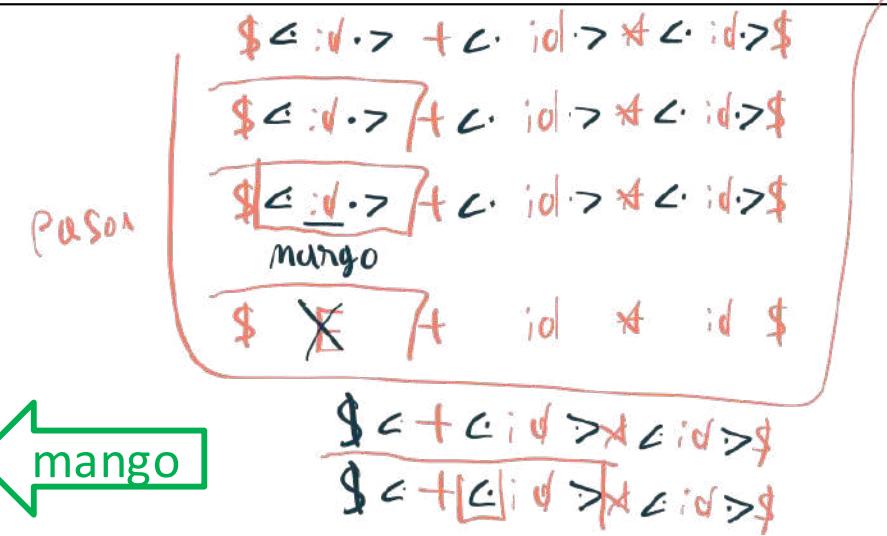
Cadena a reconocer:

w=id+id\*id

- Paso 1 → \$id+id\*id\$
- Paso 2 → \$<· id ·> + <· id ·> \* <· id ·> \$
- Paso 3 → \$<· id ·> + <· id ·> \* <· id ·> \$
- Paso 4 → \$<· id ·> + <· id ·> \* <· id ·> \$
- Paso 5 → \$<· id ·> + <· id ·> \* <· 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 ·>.
4. Devolverse hacia la izquierda hasta encontrar un <·, sobrepasando los ·=.
5. El mango se encuentra a la derecha del <· y a la izquierda del ·>.



# Uso de Relaciones de Precedencia

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

Cadena a reconocer:

w=id+id\*id

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

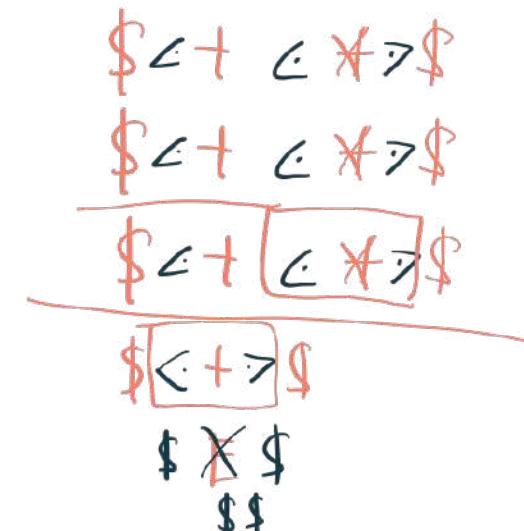
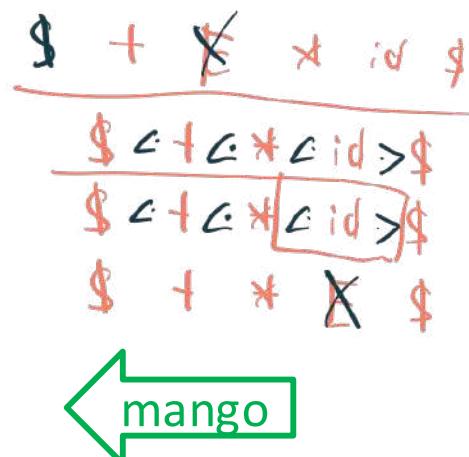
\$+id\*id\$

\$ <·+<·id·>\*<·id·>\$

\$ <·+<·id·>\*<·id·>\$

\$ <·+<·id·>\*<·id·>\$

\$ <·+<·id·>\*<·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 ·>.
4. Devolverse hacia la izquierda hasta encontrar un <·, sobrepasando los ≈.
5. El mango se encuentra a la derecha del <· y a la izquierda del ·>.

# Uso de Relaciones de Precedencia

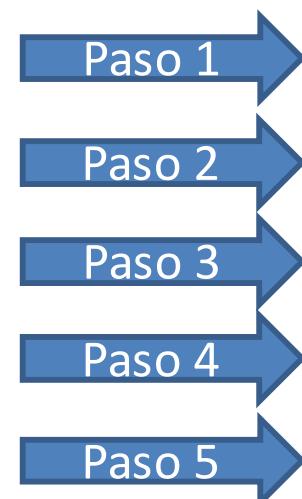
<i>a \ b</i>	+	*	id	\$
a	·>	<·	<·	·>
+	·>	·>	<·	·>
*	·>	·>	<·	·>
id	·>	·>		·>
\$	<·	<·	<·	

Cadena a reconocer:

w=id+id\*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 ·>.
4. Devolverse hacia la izquierda hasta encontrar un <·, sobrepasando los ≈.
5. El **mango** se encuentra a la derecha del <· y a la izquierda del ·>.



\$+\*id\$

\$<·+<·\* <·id·>\$

\$<·+<·\* <·id·>\$

\$<·+<·\* <·id·>\$

\$<·+<·\* <·id·>\$

mango

# Uso de Relaciones de Precedencia

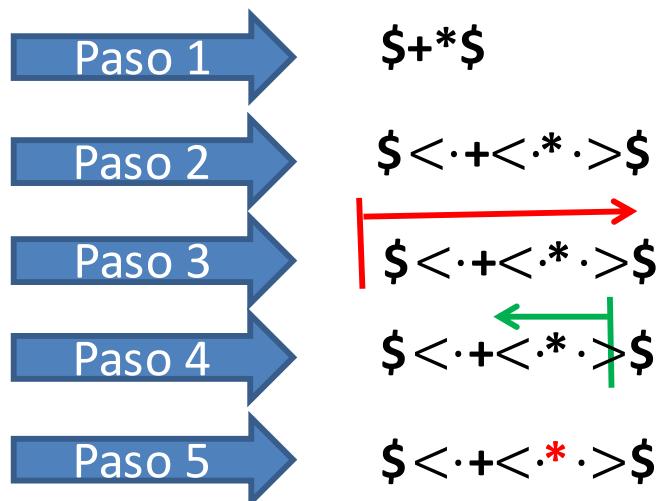
<i>a \ b</i>	+	*	id	\$
a	·>	<·	<·	·>
+	·>	·>	<·	·>
*	·>	·>	<·	·>
id	·>	·>		·>
\$	<·	<·	<·	

Cadena a reconocer:

w=id+id\*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 ·>.
4. Devolverse hacia la izquierda hasta encontrar un <·, sobrepasando los ≈.
5. El **mango** se encuentra a la derecha del <· y a la izquierda del ·>.



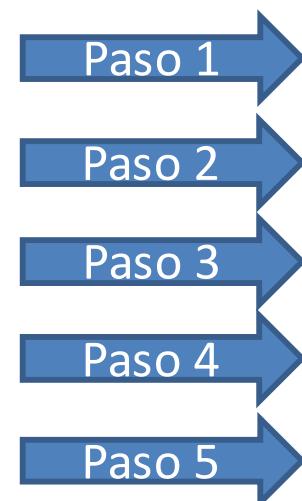
mango

# Uso de Relaciones de Precedencia

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

Cadena a reconocer:

w=id+id\*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 ·>.
4. Devolverse hacia la izquierda hasta encontrar un <·, sobrepasando los ≈.
5. El **mango** se encuentra a la derecha del <· y a la izquierda del ·>.

## Conclusión:

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



# 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  $>$

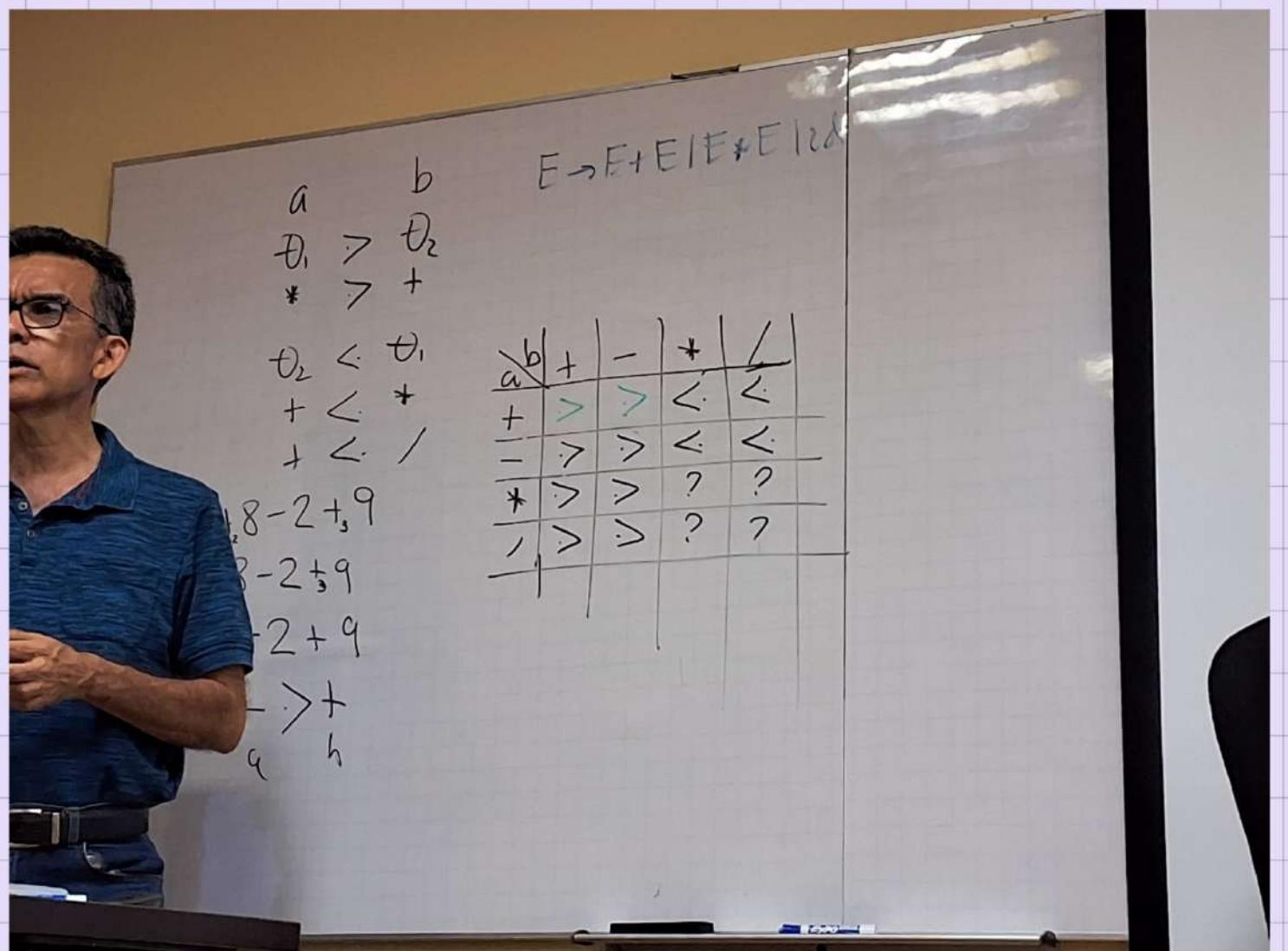
$$E \rightarrow E+E|E*E|2d$$

a	b			
$\theta_1$	$\theta_2$			
*	+			
$\theta_2 < \theta_1$	$\cancel{a} b$	+	-	$\cancel{+}$
+	+	?	?	$<$
+	/	-	?	$<$
		*	>	?
		/	>	?

$$E \rightarrow E+E|E*E|2d$$

a	b			
$\theta_1$	$\theta_2$			
*	+			
$\theta_2 < \theta_1$	$\cancel{a} b$	+	-	$\cancel{+}$
+	+	?	?	$<$
+	/	-	?	$<$
		*	>	?
		/	>	?

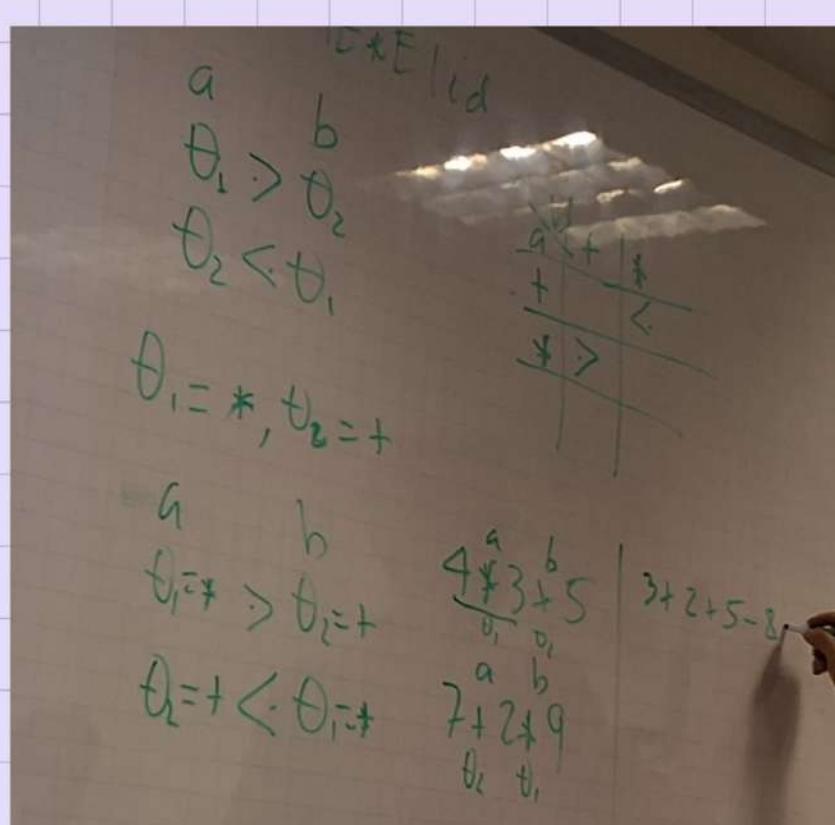
$3 + 5 + 8 - 2 + 9$   
 $8 + 8 - 2 + 9$   
 $16 - 2 + 9$   
 $- > +$   
 $a h$



el  $\wedge$ -CIRCUNFLEJO tiene asociatividad por derecha, ns: que su precedencia cuando es igual es invertida

### Gramática de operadores:

- no contiene  $E$ , no tiene 2 no terminales doyacente



# 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.:

$E \Rightarrow id$

E → E+E | E×E | (E) | ; q

# 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:

$E \Rightarrow \$ < \cdot \underline{id} \cdot > \$$

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
$$\begin{array}{l} \$ < \cdot id \\ id \cdot > \$ \\ id \cdot > \theta \end{array}$$

**Conclusión:**

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

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

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

$\Rightarrow \$ < \cdot \underline{id} \cdot > + \quad id \quad \$$  Cuál es el mango?  
 $\theta_1$

# Establecer Relaciones de Precedencia

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

**Conclusión:**

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

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

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

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

Cuál es el mango?

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

# 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:  $E \Rightarrow E+E \Rightarrow E+id \Rightarrow id+id$

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

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

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

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

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

Ya están resueltos, no aporta una nueva regla

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

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

$\theta < \cdot id$

$\$ < \cdot \theta$

$\theta \cdot > \$$

**Conclusión:**

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

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

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

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

$\Rightarrow \$ < \cdot id \cdot >^* (id + id) + id \$$

# Establecer Relaciones de Precedencia

3. Hágase

$$\begin{array}{l} \$ < \cdot id \\ id \cdot > \$ \\ id \cdot > \theta \quad (< \cdot id \\ \theta < \cdot id \quad id \cdot > ) \\ \$ < \cdot \theta \\ \theta \cdot > \$ \end{array}$$

**Conclusión:**

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

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

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

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

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

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

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

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

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

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

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

**Conclusión:**

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

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

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

$\Rightarrow \$ E^* ( \underline{\underline{E+id}} ) + id \$$  Cuál es el mango?

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

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

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

# Establecer Relaciones de Precedencia

3. Hágase

$\$ < \cdot id$

$id \cdot > \$$

$id \cdot > \theta$

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

$\theta < \cdot id$

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

$\$ < \cdot \theta$

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

$\theta \cdot > \$$

$\theta \cdot > )$

$\$ < ( \quad ) > )$   
 $) > \$$   
 $C < C$

**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 \ $$   
 $\Rightarrow \$ \ * ( < \cdot + \cdot > ) + \ id \ $$   
 $\Rightarrow \$ \ * ( + < \cdot id \cdot > ) + \ id \ $$   
 $\Rightarrow \$ \ * ( < \cdot id \cdot > + \ id ) + \ id \ $$

Cuál es el mango?

como ambos salen alavez  
tienen la misma  
precedencia

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

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

(id+id)\*id

a b	x	c	id	
\$	x			
id	x			
)	x	x		

No Se Puede

)C

((C )>))

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

(( ))

(id+id)\*id

(( ))

id\*(catid)

(( ))

id\*((id+id)\*id)

( \$

\$ id\*id X \$

\$ < (  
(< (  
)>))

# 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  $\cdot>$ .

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

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

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

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

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

$\$ < \cdot ( < \cdot$

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

# Tabla de Precedencia de Operadores

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

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

<i>a</i>	<i>b</i>	+	*	(	)	id	\$
<i>a</i>		↓	↖	↖	↗	↖	↗
+		↓	↗	↖	↗	↖	↗
*		↗	↗	↖	↗	↖	↗
(		↖	↖	↖	÷	↖	X
)		↗	↗	X	↗	X	↗
id		↗	↗	X	↗	X	↗
\$		↖	↖	↖	X	↖	X

\$ < · id	( < · id	( = · )
id · > \$	id · > )	\$ < · (
id · > θ	( < · θ	) · > \$
θ < · id	θ · > )	( < · (
\$ < · θ	θ < · (	) · > (
θ · > \$	) · > θ	) · > )

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:

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

# Tabla de Precedencia de Operadores

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

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

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

$$\begin{array}{llll}
 \$ < \cdot id & (< \cdot id & ( & \doteq \\
 id \cdot > \$ & id \cdot > ) & \$ < \cdot ( & \\
 id \cdot > \theta & (< \cdot \theta & ) \cdot > \$ & \\
 \theta < \cdot id & \theta \cdot > ) & \{ < \cdot ( & \\
 \$ < \cdot \theta & \theta \cdot > ( & ) \cdot > \} & \\
 \theta \cdot > \$ & ) \cdot > \theta & \cdot > ) &
 \end{array}$$

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:

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

# 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
```

# 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**

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

## GIC de operadores:

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

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

# Algoritmo de Análisis Sintáctico Por Precedencia de Operadores

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

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

## GIC de operadores:

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

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

Pila	Entrada	Acción
\$ <span style="color: red;">C</span> <span style="color: red;">A</span>	<span style="color: red;">b</span> <span style="color: red;">id*(id+id)+id\$</span>	desplazar
\$ id <span style="color: red;">Reemp</span>	<span style="color: red;">&gt;</span> <span style="color: red;">*(id+id)+id\$</span>	reducir
\$ <del>E</del> <span style="color: red;">&lt;</span> <span style="color: red;">*</span>	<span style="color: red;">(</span> <span style="color: red;">*(id+id)+id\$</span>	
\$ <del>E</del> <del>* E</del> <span style="color: red;">&lt;</span> <span style="color: red;">*</span>	<span style="color: red;">(</span> <span style="color: red;">id+id)</span> <span style="color: red;">id\$</span>	
\$ <del>E</del> <del>* E</del> <span style="color: red;">&lt;</span> <span style="color: red;">(</span>	<span style="color: red;">id</span> <span style="color: red;">id\$</span>	
\$ <del>E</del> <del>* E</del> <span style="color: red;">&lt;</span> <span style="color: red;">(</span> <span style="color: red;">id</span>	<span style="color: red;">+</span> <span style="color: red;">id\$</span>	
\$ <del>E</del> <del>* E</del> <span style="color: red;">&lt;</span> <span style="color: red;">(</span> <span style="color: red;">id</span> <span style="color: red;">+</span>	<span style="color: red;">id</span> <span style="color: red;">id\$</span>	
\$ <del>E</del> <del>* E</del> <span style="color: red;">&lt;</span> <span style="color: red;">(</span> <span style="color: red;">id</span> <span style="color: red;">+</span> <span style="color: red;">id</span>	<span style="color: red;">)</span> <span style="color: red;">id\$</span>	
\$ <del>E</del> <del>* E</del> <span style="color: red;">&lt;</span> <span style="color: red;">(</span> <span style="color: red;">id</span> <span style="color: red;">+</span> <span style="color: red;">id</span> <span style="color: red;">)</span>	<span style="color: red;">+</span> <span style="color: red;">id\$</span>	

# 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**

<i>a</i>	<i>b</i>	+	*	(	)	id	\$
<i>a</i>	$\diagup$						
+	$\cdot >$	$<.$	$<.$	$\cdot >$	$<.$	$\cdot >$	
*	$\cdot >$						
(	$<.$	$<.$	$<.$	$\doteq$	$<.$		
)	$\cdot >$	$\cdot >$		$\cdot >$		$\cdot >$	
id	$\cdot >$	$\cdot >$		$\cdot >$		$\cdot >$	
\$	$<.$	$<.$	$<.$			$<.$	

## GIC de operadores:

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

w= id \* (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**

<i>a</i>	<i>b</i>	+	*	(	)	id	\$
<i>a</i>	$\diagup$						
+	$\cdot >$	$<.$	$<.$	$\cdot >$	$<.$	$\cdot >$	
*	$\cdot >$	$\cdot >$	$\cdot <.$	$\cdot >$	$<.$	$\cdot >$	
(	$<.$	$<.$	$<.$	$\doteq$	$<.$		
)	$\cdot >$	$\cdot >$		$\cdot >$		$\cdot >$	
id	$\cdot >$	$\cdot >$		$\cdot >$		$\cdot >$	
\$	$<.$	$<.$	$<.$			$<.$	

# GIC de operadores:

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

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

Pila	Entrada	Acción
\$	$\text{id}^*(\text{id}+\text{id})+\text{id}\$$	desplazar
\$id	$^*(\text{id}+\text{id})+\text{id}\$$	reducir
\$	$^*(\text{id}+\text{id})+\text{id}\$$	desplazar
\$*	$(\text{id}+\text{id})+\text{id}\$$	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**

<i>a</i>	<i>b</i>	+	*	(	)	id	\$
<i>a</i>	\diagup						
+	\cdot >	<.	<.	\cdot >	<.	\cdot >	
*	\cdot >	\cdot >	\cdot >	\cdot >	\cdot >	\cdot >	
(	<.	<.	<.	\doteq	<.		
)	\cdot >	\cdot >		\cdot >		\cdot >	
id	\cdot >	\cdot >		\cdot >		\cdot >	
\$	<.	<.	<.			<.	

# GIC de operadores:

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

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

Pila	Entrada	Acción
\$	$\text{id}^*(\text{id}+\text{id})+\text{id}\$$	desplazar
\$id	$^*(\text{id}+\text{id})+\text{id}\$$	reducir
\$	$^*(\text{id}+\text{id})+\text{id}\$$	desplazar
\$*	$(\text{id}+\text{id})+\text{id}\$$	desplazar
\$*(	$\text{id}+\text{id})+\text{id}\$$	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 ()
end
  
```

<i>a</i>	<i>b</i>	+	*	(	)	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
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+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 ()
end
  
```

<i>a</i>	<i>b</i>	+	*	(	)	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
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+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 ()
end
  
```

a \ 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
\$	*(id+id)+id\$	desplazar
\$*	(id+id)+id\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	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 ()
end
  
```

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

# 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 ()
end
  
```

a \ 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
\$	*(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

# 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 ()
end
  
```

<i>a</i>	<i>b</i>	+	*	(	)	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\$	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
\$*(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 ()
end
  
```

a \ 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\$	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>l</u> <i>Teeme</i>	+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 ()
end
  
```

<i>a \ b</i>	+	*	(	)	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\$	desplazar
\$*(	id+id)+id\$	desplazar
\$*(id	+id)+id\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)+id\$	reducir
\$*(+	)+id\$	reducir
\$*(	)+id\$	desplazar
\$*()	+id\$	reducir
\$* <del>t</del> > <del>Teem</del>	+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 ()
end
  
```

a \ 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\$	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 ()
end
  
```

a \ 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\$	reducir
\$*(	+id)+id\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)+id\$	reducir
\$*(+	)+id\$	reducir
\$*(	)+id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$	+id\$	desplazar
\$+ ↲ ↴	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 ()
end
  
```

a \ 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\$	desplazar
\$*(+	id)+id\$	desplazar
\$*(+id	)+id\$	reducir
\$*(+	)+id\$	reducir
\$*(	)+id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$	+id\$	desplazar
\$+	id\$	desplazar
\$+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 ()
end
  
```

<i>a \ b</i>	+	*	(	)	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\$	desplazar
\$*(+id	)+id\$	reducir
\$*(+	)+id\$	reducir
\$*(	)+id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$	+id\$	desplazar
\$+	id\$	desplazar
\$+id	\$	reducir
\$+ <u>E+E</u> feeme	\$	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 ()
end
  
```

a \ 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\$	reducir
\$*(+	) +id\$	reducir
\$*(	) +id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$	+id\$	desplazar
\$+	id\$	desplazar
\$+id	\$	reducir
\$+	\$	reducir
\$	\$	aceptar

# 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 ()
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\$	reducir
\$*(	) + id\$	desplazar
\$*()	+id\$	reducir
\$*	+id\$	reducir
\$	+id\$	desplazar
\$+	id\$	desplazar
\$+id	\$	reducir
\$+	\$	reducir
\$	\$	aceptar

Tomado de Compiladores. Principios, Técnicas y Herramientas, Aho, Sethi, Ullman. Pearson

# 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

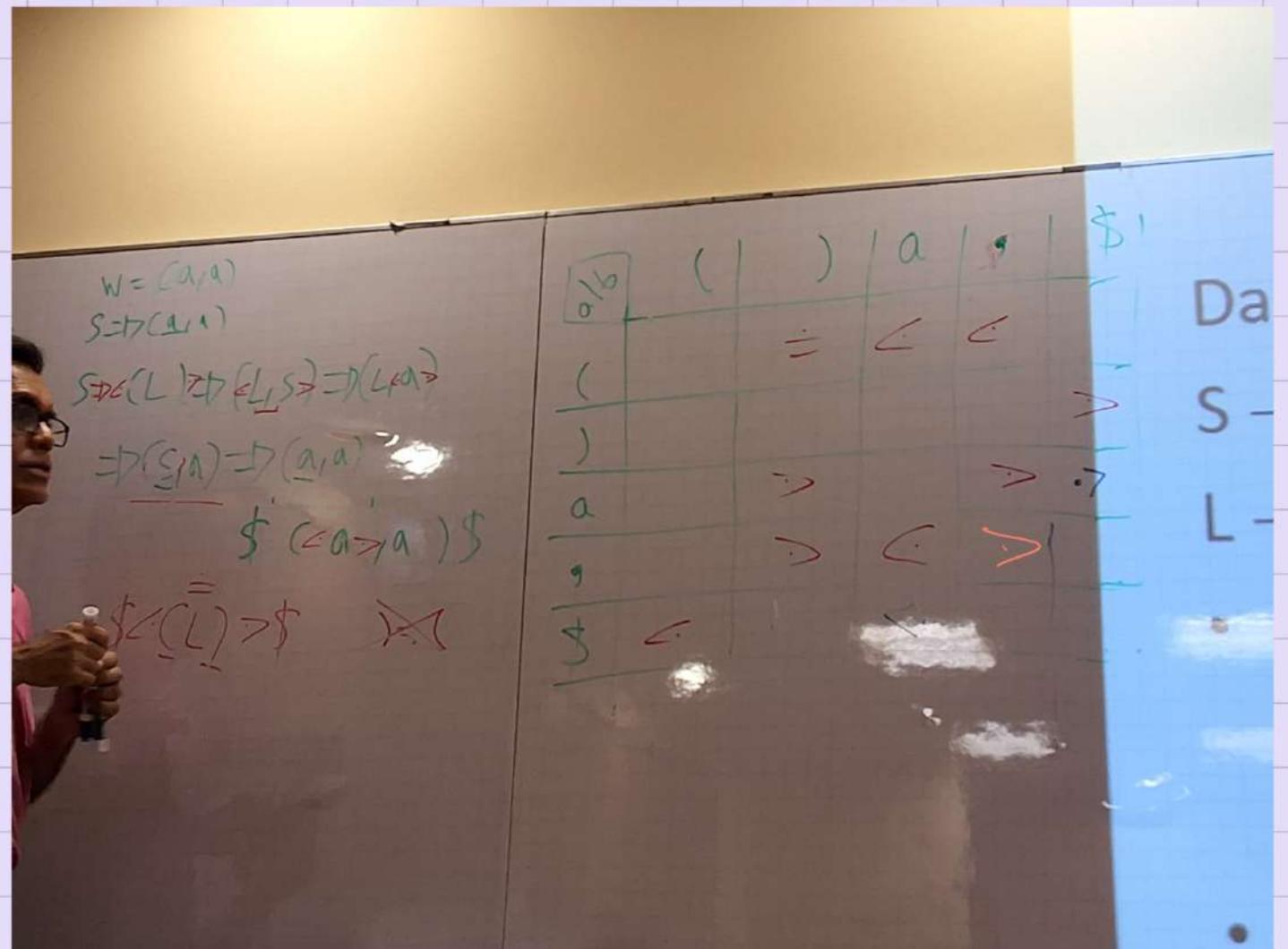
Dada la siguiente GIC:

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

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

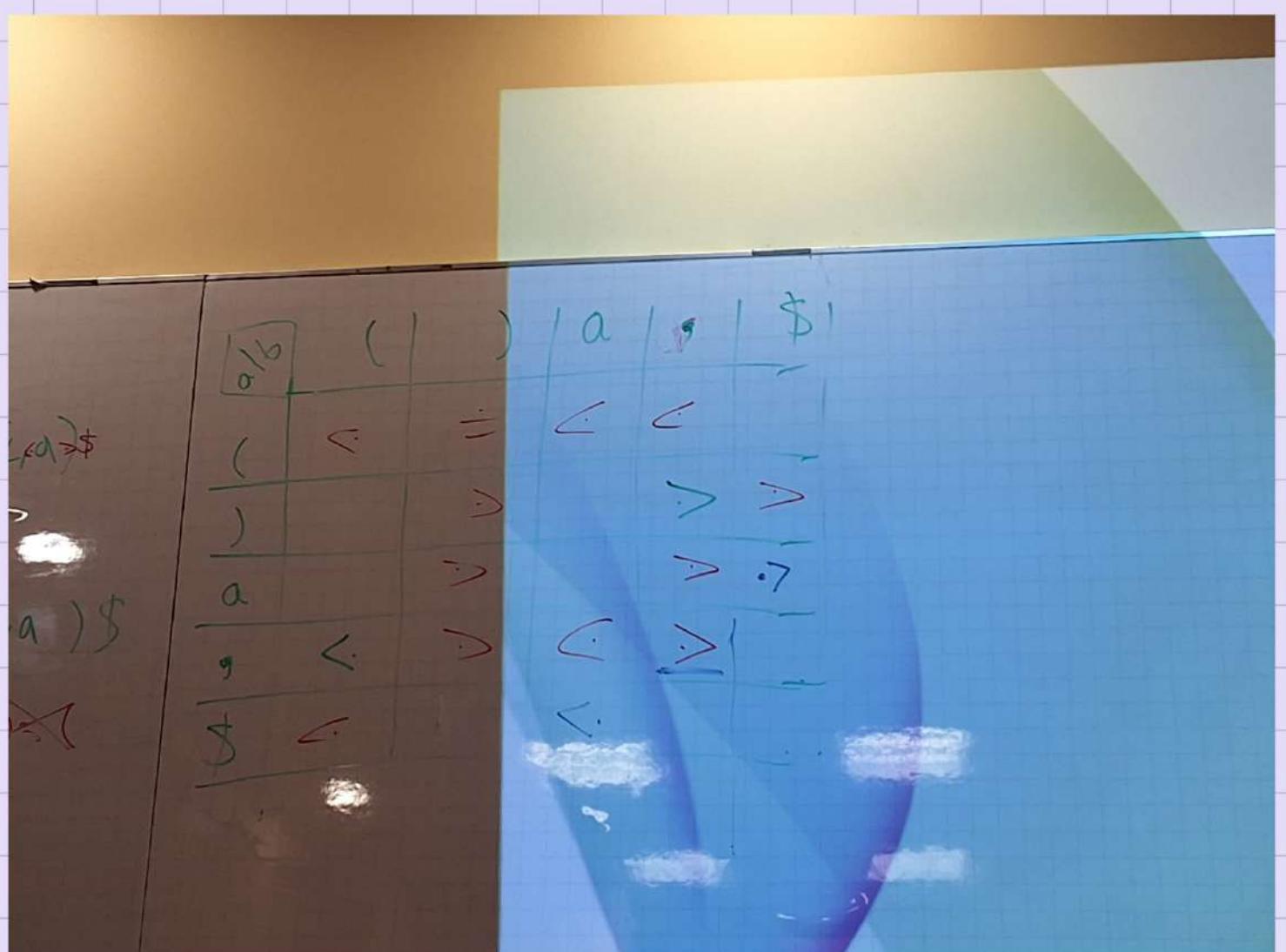
W = a   S  $\Rightarrow$   $\lambda$    \$ < a > \$

	(	)	a	,	\$
(					
)					
a					↗
,					
\$				↖	



$S \Rightarrow^* (\underline{L}) \Rightarrow^* (S) \Rightarrow^* (\underline{(L)}) \Rightarrow^* (\underline{(S)}) \Rightarrow^* (a)$

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



$W = (\alpha, \alpha, \alpha)$

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

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

$(L, \alpha, \alpha)$