

Definición S-atribuida (1): expresiones aritméticas

- Sólo se utilizan atributos *sintetizados*.

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $S \rightarrow E$	Imprimir ($E.valor$)
2 $E \rightarrow T$	$E.valor = T.valor$
3 $E \rightarrow E_1 + T$	$E.valor = E_1.valor + T.valor$
4 $T \rightarrow F$	$T.valor = F.valor$
5 $T \rightarrow T_1 * F$	$T.valor = T_1.valor * F.valor$
6 $F \rightarrow (E)$	$F.valor = E.valor$
7 $F \rightarrow \text{número}$	$F.valor = \text{número.valor_léxico}$
8 $F \rightarrow \text{identificador}$	$F.valor = \text{identificador.valor}$

Definición L-atribuida (2): declaración de variables en C

- Definición Basada en la Sintaxis con atributos *sintetizados* y *heredados*

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $D \rightarrow T L ;$	$L.h = T.tipo$
2 $T \rightarrow int$	$T.tipo = entero$
3 $T \rightarrow float$	$T.tipo = real$
4 $L \rightarrow identificador$	Añadir_tipo(identificador.tabla_símbolos, $L.h$)
5 $L \rightarrow L_1 , identificador$	$L_1.h = L.h$ Añadir_tipo(identificador.tabla_símbolos, $L.h$)

Definición L-atribuida (3):

Expresiones aritméticas *sin* recursividad por la izquierda

- Definición Basada en la Sintaxis con atributos *sintetizados* y *heredados*

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $E \rightarrow T R$	$R.h = T.valor$ $E.valor = R.s$
2 $T \rightarrow (E)$	$T.valor = E.valor$
3 $T \rightarrow \text{número}$	$T.valor = \text{número.valor_léxico}$
4 $T \rightarrow \text{identificador}$	$T.valor = \text{identificador.valor}$
5 $R \rightarrow + T R_1$	$R_1.h = R.h + T.valor$ $R.s = R_1.s$
6 $R \rightarrow - T R_1$	$R_1.h = R.h - T.valor$ $R.s = R_1.s$
7 $R \rightarrow \epsilon$	$R.s = R.h$

Esquema de Traducción (1): expresiones aritméticas

- *Sólo se utilizan atributos **sintetizados***

-
1. $S \rightarrow E \{ \text{Imprimir (E.valor)} \}$
 2. $E \rightarrow T \quad \{ \text{E.valor} = \text{T.valor} \}$
 3. $E \rightarrow E_1 + T \{ \text{E.valor} = \text{E}_1.\text{valor} + \text{T.valor} \}$
 4. $T \rightarrow F \quad \{ \text{T.valor} = \text{F.valor} \}$
 5. $T \rightarrow T_1 * F \{ \text{T.valor} = \text{T}_1.\text{valor} * \text{F.valor} \}$
 6. $F \rightarrow (E) \quad \{ \text{F.valor} = \text{E.valor} \}$
 7. $F \rightarrow \text{número} \quad \{ \text{F.valor} = \text{número.valor_lógico} \}$
 8. $F \rightarrow \text{identificador} \{ \text{F.valor} = \text{identificador.valor} \}$

Esquema de Traducción (2): declaración de variables en C

- Se utilizan atributos *heredados* y *sintetizados*

1. $D \rightarrow T \{ L.h = T. Tipo \} L ;$
2. $T \rightarrow \text{int} \{ T.tipo = \text{entero} \}$
3. $T \rightarrow \text{float} \{ T.tipo = \text{real} \}$
4. $L \rightarrow \text{identificador}$
 $\{ \text{Añadir_tipo}(\text{identificador.tabla_símbolos}, L.h) \}$
5. $L \rightarrow \{ L_1.h = L.h \} L_1 , \text{identificador}$
 $\{ \text{Añadir_tipo}(\text{identificador.tabla_símbolos}, L.h) \}$

***Esquema de Traducción (3):
expresiones aritméticas *sin* recursividad por la izquierda***

- Se utilizan atributos *heredados* y *sintetizados*
-

1. $E \rightarrow T \{ \mathbf{R.h} = \mathbf{T.valor} \} R \{ \mathbf{E.valor} = \mathbf{R.s} \}$
2. $T \rightarrow (E) \{ \mathbf{T.valor} = \mathbf{E.valor} \}$
3. $T \rightarrow \mathbf{número} \{ \mathbf{T.valor} = \mathbf{número.valor_léxico} \}$
4. $T \rightarrow \mathbf{identificador} \{ \mathbf{T.valor} = \mathbf{identificador.valor} \}$
5. $R \rightarrow + T \{ \mathbf{R_1.h} = \mathbf{R.h} + \mathbf{T.valor} \} R_1 \{ \mathbf{R.s} = \mathbf{R_1.s} \}$
6. $R \rightarrow - T \{ \mathbf{R_1.h} = \mathbf{R.h} - \mathbf{T.valor} \} R_1 \{ \mathbf{R.s} = \mathbf{R_1.s} \}$
7. $R \rightarrow \varepsilon \{ \mathbf{R.s} = \mathbf{R.h} \}$

***Traductor Descendente Predictivo:
Tabla predictiva correspondiente al esquema de traducción (3)***

	+	-	()	número	identificador	\$
E			1		1	1	
T			2		3	4	
R	5	6		7			7

Traductor Descendente Predictivo del esquema de traducción (3)

Función E (): entero

variables

entero Tvalor {Atributo sintetizado de T}
 Rh {Atributo heredado de R}
 Rs {Atributo sintetizado de R}

inicio

si ((TOKEN = '(') o (TOKEN = número) o (TOKEN = identificador)

entonces

Tvalor = T()

Rh = Tvalor

Rs = R(Rh) {R tiene un atributo heredado}

{Valor del atributo sintetizado de E: E.valor}

Devolver Rs

{Se puede simplificar escribiendo **Devolver** R(T()) }

si no

Error(falta paréntesis izquierdo, número o identificador)

fin_si

fin

Función T (): entero

variables

entero Evalor {Atributo sintetizado de E}
 número_valor {Atributo sintetizado de número}
 identificador_valor {Atributo sintetizado de identificador}

inicio

si (TOKEN = '(')

entonces

avanzar (TOKEN)

 Evalor = E()

si (TOKEN = ')')

entonces **avanzar**(TOKEN)

 {Valor del atributo sintetizado de T: T.valor}

Devolver Evalor

si no

Error (falta paréntesis derecho)

fin_si

si no

si (TOKEN = número)

entonces número_valor = valor(TOKEN)

avanzar(TOKEN)

 {Valor del atributo sintetizado de T: T.valor}

Devolver número_valor

si no

si (TOKEN = identificador)

entonces

 identificador_valor = valor(TOKEN)

avanzar(TOKEN)

 {Valor del atributo sintetizado de T: T.valor}

Devolver identificador_valor

si no

Error (falta identificador, número o
 paréntesis izquierdo)

fin_si

fin_si

fin_si

fin

Función R (entero H): entero

{H es el argumento correspondiente al atributo heredado de R}

variables

entero Tvalor {Atributo sintetizado de T}
Rh1 {Atributo heredado de R1}
Rs1 {Atributo sintetizado de R1}

inicio

si (*TOKEN* = '+')

entonces

avanzar (*TOKEN*)

Tvalor = *T*()

Rh1 = *H* + *Tvalor*

Rs1 = *R* (*Rh1*)

Devolver *Rs1* {Valor del atributo sintetizado de R: R.s}

{Se puede simplificar como Devolver R(H + T())}

si no

si (*TOKEN* = '-')

entonces

avanzar (*TOKEN*)

Tvalor = *T*()

Rh1 = *H* - *Tvalor*

Rs1 = *R* (*Rh1*)

{Valor del atributo sintetizado de R: R.s}

Devolver *Rs1*

{Se puede simplificar como Devolver R(H - T())}

si no

{Valor del atributo sintetizado de R: R.s}

Devolver *H*;

{Se ha utilizado la regla $R \rightarrow \epsilon$ para los demás símbolos terminales, aunque en un principio sólo se podía aplicar a "(" y \$}

fin_si

fin_si

fin

Definición S-atribuida (4)

- Se ha modificado la definición S-atribuida (1) para utilizar “la pila valor” y realizar una evaluación *ascendente*
- Se pueden *conocer* “las posiciones” de los valores

	<i>Regla de Producción</i>	<i>Acción semántica</i>
1	$S \rightarrow E$	Imprimir (valor[cima])
2	$E \rightarrow T$	valor[nueva-cima] = valor [cima] (superflua)
3	$E \rightarrow E_1 + T$	valor[nueva-cima] = valor [cima - 2] + valor[cima]
4	$T \rightarrow F$	valor[nueva-cima] = valor [cima] (superflua)
5	$T \rightarrow T_1 * F$	valor[nueva-cima] = valor [cima - 2] * valor[cima]
6	$F \rightarrow (E)$	valor[nueva-cima] = valor [cima - 1]
7	$F \rightarrow \text{número}$	valor[nueva-cima] = valor [cima] (superflua)
8	$F \rightarrow \text{identificador}$	valor[nueva-cima] = valor [cima] (superflua)

Notas:

- Al desplazar, **nueva-cima = cima + 1**
- Al reducir, si $A \rightarrow \alpha$ es la regla de producción utilizada entonces
nueva-cima = cima - | α | + 1

Tabla de Análisis Sintáctico LALR de la Definición S-atribuida (4)

	ACCIÓN							IR A			
	id	número	+	*	()	\$	S	E	T	F
0	d 7	d 6			d 5			1	2	3	4
1							Aceptar				
2	<u>r 1</u>	<u>r 1</u>	d 8	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	r 1				
3	<u>r 2</u>	<u>r 2</u>	r 2	d 9	<u>r 2</u>	r 2	r 2				
4	<u>r 4</u>	<u>r 4</u>	r 4	r 4	<u>r 4</u>	r 4	r 4				
5	d 7	d 6			d 5				10	3	4
6	<u>r 7</u>	<u>r 7</u>	r 7	r 7	<u>r 7</u>	r 7	r 7				
7	<u>r 8</u>	<u>r 8</u>	r 8	r 8	<u>r 8</u>	r 8	r 8				
8	d 7	d 6			d 5					11	4
9	d 7	d 6			d 5						12
10			d 8			d 13					
11	<u>r 3</u>	<u>r 3</u>	r 3	d 9	<u>r 3</u>	r 3	r 3				
12	<u>r 5</u>	<u>r 5</u>	r 5	r 5	<u>r 5</u>	r 5	r 5				
13	<u>r 6</u>	<u>r 6</u>	r 6	r 6	<u>r 6</u>	r 6	r 6				

Traducción Ascendente de la expresión: $3 * (4 + 5)$

<i>PILA de análisis sintáctico</i>	<i>ENTRADA</i>	<i>ACCIÓN</i>	<i>VALOR</i>
0	número * (número + número) \$	d 6	
0 número 6	* (número + número) \$	r 7 F → número	3
0 F 4	* (número + número) \$	r 5 T → F	3
0 T 3	* (número + número) \$	d 9	3
0 T 3 * 9	(número + número) \$	d 5	3 -
0 T 3 * 9 (5	número + número) \$	d 6	3 - -
0 T 3 * 9 (5 número 6	+ número) \$	r 7 F → número	3 - - 4
0 T 3 * 9 (5 F 4	+ número) \$	r 5 T → F	3 - - 4
0 T 3 * 9 (5 T 3	+ número) \$	r 2 E → T	3 - - 4
0 T 3 * 9 (5 E 10	+ número) \$	d 8	3 - - 4
0 T 3 * 9 (5 E 10 + 8	número) \$	d 6	3 - - 4 -
0 T 3 * 9 (5 E 10 + 8 número 6) \$	r 7 F → número	3 - - 4 - 5
0 T 3 * 9 (5 E 10 + 8 F 4) \$	r 5 T → F	3 - - 4 - 5
0 T 3 * 9 (5 E 10 + 8 T 11) \$	r 3 E → E + T	3 - - 4 - 5
0 T 3 * 9 (5 E 10) \$	d 13	3 - - 9
0 T 3 * 9 (5 E 10) 13	\$	r 6 F → (E)	3 - - 9 -
0 T 3 * 9 F 12	\$	r 5 T → T * F	3 - 9
0 T 3	\$	r 3 E → T	27
0 E 2	\$	r 1 S → E	27
0 S 1	\$	Aceptar	27

Esquema de traducción (4): uso de acciones intercaladas

- Este esquema de traducción permite **transformar**
 - expresiones aritméticas escritas con notación “**infija**”: $10 - (2 + 3)$
 - a expresiones escritas con notación “**postfija**”: $10 (2 3 +) -$

-
1. $E \rightarrow T R$
 2. $R \rightarrow + T \{ \text{imprimir}(\text{"+"}) \} R$
 3. $R \rightarrow - T \{ \text{imprimir}(\text{"-"}) \} R$
 4. $R \rightarrow \varepsilon$
 5. $T \rightarrow (\{ \text{imprimir}(\text{"("}) \} E) \{ \text{imprimir}(\text{")"}) \}$
 6. $T \rightarrow \text{número} \{ \text{imprimir}(\text{número.valor_léxico}) \}$
 7. $T \rightarrow \text{identificador} \{ \text{imprimir}(\text{identificador.valor}) \}$

Esquema de Traducción (5)

- Se han usado de “**símbolos marcadores**” para eliminar las “acciones intercaladas” del esquema de traducción (4)

-
1. $E \rightarrow T R$
 2. $R \rightarrow + T M_1 R$
 3. $R \rightarrow - T M_2 R$
 4. $R \rightarrow \varepsilon$
 5. $T \rightarrow (M_3 E) \{ \text{imprimir}(\text{"("}) \}$
 6. $T \rightarrow \text{número} \{ \text{imprimir}(\text{número.valor_léxico}) \}$
 7. $T \rightarrow \text{identificador} \{ \text{imprimir}(\text{identificador.valor}) \}$

 8. $M_1 \rightarrow \varepsilon \{ \text{imprimir}(\text{"+"}) \}$
 9. $M_2 \rightarrow \varepsilon \{ \text{imprimir}(\text{"-"}) \}$
 10. $M_3 \rightarrow \varepsilon \{ \text{imprimir}(\text{"("}) \}$

Tabla de Análisis Sintáctico SLR del Esquema de Traducción (5)

	ACCIÓN							IR A					
	+	-	()	número	identificador	\$	E	R	T	M ₁	M ₂	M ₃
0			d3		d4	d5		1		2			
1							ACEPTAR						
2	d7	d8		r4			r4		6				
3			r10		r10	r10							9
4	r6	r6		r6			r6						
5	r7	r7		r7			r7						
6				r1			r1						
7			d3		d4	d5				10			
8			d3		d4	d5				11			
9			d3		d4	d5		12		2			
10	r8	r8		r8			r8				13		
11	r9	r9		r9			r9					14	
12				d15									
13	d7	d8		r4			r4		16				
14	d7	d8		r4			r4		17				
15	r5	r5		r5			r5						
16				r2			r2						
17				r3			r3						

Definición L-atribuida (2): declaración de variables en C

- La gramática subyacente es *recursiva por la izquierda*
- Se indican *“las reglas de copia”*

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $D \rightarrow T L ;$	$L.h = T.tipo$ (<i>regla de copia</i>)
2 $T \rightarrow int$	$T.tipo = entero$
3 $T \rightarrow float$	$T.tipo = real$
4 $L \rightarrow identificador$	Añadir_tipo($identificador.tabla_símbolos, L.h$)
5 $L \rightarrow L_1 , identificador$	$L_1.h = L.h$ (<i>regla de copia</i>) Añadir_tipo($identificador.tabla_símbolos, L.h$)

Definición L-atribuida (5): declaración de variables en C

- Se ha modificado la definición L-atribuida (2) para utilizar “*la pila valor*” y realizar una evaluación *ascendente*.
- Se pueden *conocer* “*las posiciones*” de los valores.
- La gramática subyacente es *recursiva por la izquierda*.

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $D \rightarrow TL ;$	
2 $T \rightarrow \text{int}$	valor[nueva_cima] = entero (superflua)
3 $T \rightarrow \text{float}$	valor[nueva_cima] = real (superflua)
4 $L \rightarrow \text{identificador}$	Añadir_tipo(valor[cima],valor[cima-1])
5 $L \rightarrow L_1 , \text{identificador}$	valor[cima-2] = valor[cima-3] (superflua) Añadir_tipo(valor[cima],valor[cima-3])

Tabla de Análisis Sintáctico LALR de la Definición L-atribuida (5)

	;	int	float	identificador	,	\$		D	T	L
0		d 3	d 4					1	2	
1						ACEPTAR				
2				d 6						5
3	<u>r 2</u>	<u>r 2</u>	<u>r 2</u>	r 2	<u>r 2</u>	<u>r 2</u>				
4	<u>r 3</u>	<u>r 3</u>	<u>r 3</u>	r 3	<u>r 3</u>	<u>r 3</u>				
5	d 7				d 8					
6	<u>r 4</u>	<u>r 4</u>	<u>r 4</u>	<u>r 4</u>	r 4	r 4				
7	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	r 1				
8				d 9						
9	r 5	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>	r 5	r 5				

Definición L-atribuida (6): declaración de variables en C

- Se ha modificado la Definición L-atribuida (5) para que la gramática subyacente sea **recursiva por la derecha**
- Las posiciones de los valores de los atributos **no son predecibles**

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $D \rightarrow T L ;$	$L.h = T.tipo$ (<i>regla de copia</i>)
2 $T \rightarrow int$	$T.tipo = entero$
3 $T \rightarrow float$	$T.tipo = real$
4 $L \rightarrow identificador$	Añadir_tipo(identificador.tabla_símbolos, $L.h$)
5* $L \rightarrow identificador , L_1$	Añadir_tipo(identificador.tabla_símbolos, $L.h$) $L_1.h = L.h$

Tabla de Análisis Sintáctico LALR de la Definición L-atribuida (6)

	;	int	float	identificador	,	\$		D	T	L
0		d 3	d 4					1	2	
1						ACEPTAR				
2				d 6						5
3	<u>r 2</u>	<u>r 2</u>	<u>r 2</u>	r 2	<u>r 2</u>	<u>r 2</u>				
4	<u>r 3</u>	<u>r 3</u>	<u>r 3</u>	r 3	<u>r 3</u>	<u>r 3</u>				
5	d 7									
6	r 4	<u>r 4</u>	<u>r 4</u>	<u>r 4</u>	d 8	<u>r 4</u>				
7	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	r 1				
8				d 6						9
9	r 5	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>				

Definición L-atribuida (7): declaración de variables en C

- Se han introducido “**símbolos marcadores**” en la Definición L-atribuida (6)
- Las posiciones de los valores de los atributos **sí son predecibles**

Regla de Producción	Acción semántica
1* $D \rightarrow T M_1 L ;$	$M_1.h = T.tipo$ $L.h = M_1.s$
2 $T \rightarrow int$	$T.tipo = entero$
3 $T \rightarrow float$	$T.tipo = real$
4 $L \rightarrow identificador$	Añadir_tipo(identificador.tabla_símbolos,L.h)
5* $L \rightarrow identificador , M_2 L_1$	$M_2.h = L.h$ $L_1.h = M_2.s$ Añadir_tipo(identificador.tabla_símbolos,L.h)
6* $M_1 \rightarrow \epsilon$	$M_1.s = M_1.h$
7* $M_2 \rightarrow \epsilon$	$M_2.s = M_2.h$

- **Nota importante:** no se debe confundir el significado de los subíndices
 - M_1 y M_2 son símbolos diferentes
 - L_1 representa una nueva aparición del símbolo L

Definición L-atribuida (8): declaración de variables en C

- Se ha modificado la definición L-atribuida (7) para utilizar “la pila valor” y realizar una evaluación ascendente.
- Se pueden *conocer* “las posiciones” de los valores.

<i>Regla de Producción</i>	<i>Acción semántica</i>
1* $D \rightarrow T M_1 L ;$	
2 $T \rightarrow \text{int}$	valor[nueva_cima] = entero (superflua)
3 $T \rightarrow \text{float}$	valor[nueva_cima] = real (superflua)
4 $L \rightarrow \text{identificador}$	Añadir_tipo(valor[cima],valor[cima-1])
5* $L \rightarrow \text{identificador} , M_2 L_1$	Añadir_tipo(valor[cima-3],valor[cima-1])
6* $M_1 \rightarrow \epsilon$	valor[nueva_cima] = valor[cima]
7* $M_2 \rightarrow \epsilon$	valor[nueva_cima] = valor[cima-2]

Tabla de Análisis Sintáctico LALR de la Definición L-atribuida (8)

	;	int	float	identificador	,	\$		D	T	L	M ₁	M ₂
0		d 3	d 4					1	2			
1						ACEPTAR						
2	<u>r 6</u>	<u>r 6</u>	<u>r 6</u>	r 6	<u>r 6</u>	<u>r 6</u>					5	
3	<u>r 2</u>	<u>r 2</u>	<u>r 2</u>	r 2	<u>r 2</u>	<u>r 2</u>						
4	<u>r 3</u>	<u>r 3</u>	<u>r 3</u>	r 3	<u>r 3</u>	<u>r 3</u>						
5				d 7						6		
6	d 8											
7	r 4	<u>r 4</u>	<u>r 4</u>	<u>r 4</u>	d 9	<u>r 4</u>						
8	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	r 1						9
9	<u>r 7</u>	<u>r 7</u>	<u>r 7</u>	r 7	<u>r 7</u>	<u>r 7</u>						10
10				d 7						11		
11	r 5	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>						

Traducción Ascendente de *int a, b, c;*

<i>PILA de análisis sintáctico</i>	<i>ENTRADA</i>	<i>ACCIÓN</i>	<i>VALOR</i>
0	int id, id, id; \$	d 3	
0 int 3	id, id, id; \$	r 2: T → int	entero
0 T 2	id, id, id; \$	r 6: M ₁ → ε	entero
0 T 2 M ₁ 5	id, id, id; \$	d 7	entero entero
0 T 2 M ₁ 5 id 7	, id, id; \$	d 9	entero entero a
0 T 2 M ₁ 5 id 7, 9	id, id; \$	r 7: M ₂ → ε	entero entero a ,
0 T 2 M ₁ 5 id 7, 9 M ₂ 10	id, id; \$	d 7	entero entero a , entero
0 T 2 M ₁ 5 id 7, 9 M ₂ 10 id 7	, id; \$	d 9	entero entero a , entero b
0 T 2 M ₁ 5 id 7, 9 M ₂ 10 id 7, 9	id; \$	r 7: M ₂ → ε	entero entero a , entero b ,
0 T 2 M ₁ 5 id 7, 9 M ₂ 10 id 7, 9 M ₂ 10	id; \$	d 7	entero entero a , entero b , entero
0 T 2 M ₁ 5 id 7, 9 M ₂ 10 id 7, 9 M ₂ 10 id 7	; \$	r 4: L → id	entero entero a , entero b , entero c
0 T 2 M ₁ 5 id 7, 9 M ₂ 10 id 7, 9 M ₂ 10 id 7, 9 M ₂ 10 L 11	; \$	r 5: L → id , M ₂ L	entero entero a , entero b , entero c
0 T 2 M ₁ 5 id 7, 9 M ₂ 10 L 11	; \$	r 5: L → id , M ₂ L	entero entero a , entero b
0 T 2 M ₁ 5 L 6	; \$	d 8	entero entero a
0 T 2 M ₁ 5 L 6 ; 8	\$	r 1: D → T M ₁ L	entero entero entero a ;
0 D 1	\$	Aceptar	entero

- Se indican en color azul los valores de los atributos utilizados en las acciones semánticas
- Se indican en color granate los símbolos que forman parte del pivote

- **Definición S-atribuida (9): declaración de variables en C**
- Sólo se utilizan atributos *sintetizados*
- **Inconveniente:** la gramática **“oculta”** la sintaxis del lenguaje

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $D \rightarrow L \text{ identificador ;}$	Añadir_tipo(identificador.tabla_símbolos,L.s)
2 $L \rightarrow L_1 \text{ identificador ,}$	Añadir_tipo(identificador.tabla_símbolos, L ₁ .s) L.s = L ₁ .s
3 $L \rightarrow T$	L.s = T.tipo
4 $T \rightarrow \text{int}$	T.tipo = entero
5 $T \rightarrow \text{float}$	T.tipo = real

*Definición basada en la sintaxis (10): declaración de variables en **Pascal***

- *Se utilizan atributos heredados y sintetizados.*
- *Esta definición **no** es L-atribuida*
- *La gramática “refleja con claridad” la sintaxis del lenguaje Pascal*

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $D \rightarrow L : T ;$	$L.h = T.tipo$ (No L-atribuida)
2 $L \rightarrow \text{identificador}$	Añadir_tipo(identificador.tabla_símbolos, L.h)
3 $L \rightarrow L_1 , \text{identificador}$	Añadir_tipo(identificador.tabla_símbolos, L.h) $L_1.h = L.h$
4 $T \rightarrow \text{integer}$	$T.tipo = \text{entero}$
5 $T \rightarrow \text{real}$	$T.tipo = \text{real}$

Definición S-atribuida (11): declaración de variables en *Pascal*

- Sólo se utilizan atributos *sintetizados*.
- La gramática “*oculta*” la sintaxis del lenguaje *Pascal*

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $D \rightarrow \text{identificador } L$	Añadir_tipo(identificador.tabla_símbolos, L.s)
2 $L \rightarrow , \text{ identificador } L_1$	Añadir_tipo(identificador.tabla_símbolos, L ₁ .s) L.s = L ₁ .s
3 $L \rightarrow : T ;$	L.s = T.tipo
4 $T \rightarrow \text{integer}$	T.tipo = entero
5 $T \rightarrow \text{real}$	T.tipo = real

Definición S-atribuida (12): declaración de variables en Pascal

- Se ha modificado la definición S-atribuida (11) para utilizar “*la pila valor*” y realizar una evaluación *ascendente*.
- Se pueden *conocer* “*las posiciones*” de los valores.

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $D \rightarrow \text{identificador } L$	Añadir_tipo(valor[cima-1],valor[cima])
2 $L \rightarrow , \text{ identificador } L_1$	Añadir_tipo(valor[cima-1],valor[cima]) valor[nueva_cima]=valor[cima]
3 $L \rightarrow : T ;$	valor[nueva_cima]=valor[cima-1]
4 $T \rightarrow \text{integer}$	valor[nueva_cima]=entero (superflua)
5 $T \rightarrow \text{real}$	valor[nueva_cima]=real (superflua)

Tabla de Análisis Sintáctico LALR de la Definición S-atribuida (12)

	identificador	,	:	integer	real	\$		D	L	T
0	d 2							1		
1						ACEPTAR				
2		d 4	d 5						3	
3	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	<u>r 1</u>	r 1				
4	d 6									
5				d 8	d 9					7
6		d 4	d 5						10	
7	<u>r 3</u>	<u>r 3</u>	<u>r 3</u>	<u>r 3</u>	<u>r 3</u>	r 3				
8	<u>r 4</u>	<u>r 4</u>	<u>r 4</u>	<u>r 4</u>	<u>r 4</u>	r 4				
9	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>	<u>r 5</u>	r 5				
10	<u>r 2</u>	<u>r 2</u>	<u>r 2</u>	<u>r 2</u>	<u>r 2</u>	r 2				

Definición S-atribuida (13): Generación de código intermedio para expresiones aritméticas

<i>Regla de Producción</i>	<i>Acción semántica</i>
1 $S \rightarrow id = E$	$S.código = E.código$ generar (id.nombre '=' E.variable)
2 $E \rightarrow T$	$E.variable = T.variable$ $E.código = T.código$
3 $E \rightarrow E_1 + T$	$E.variable = nueva_temporal()$ $E.código = E_1.código$ $T.código$ generar(E.variable '=' $E_1.variable$ '+' T.variable)
4 $T \rightarrow F$	$T.variable = F.variable$ $T.código = F.código$
5 $T \rightarrow T_1 * F$	$T.variable = nueva_temporal()$ $T.código = T_1.código$ $F.código$ generar(T.variable '=' $T_1.variable$ '*' F.variable)
6 $F \rightarrow (E)$	$F.variable = E.variable$ $F.código = E.código$
7 $F \rightarrow número$	$F.variable = nueva_temporal()$ $F.código = generar(F.variable '=' número.valor_léxico)$
8 $F \rightarrow id$	$F.variable = id.nombre$ $F.código = ''$

**Tabla de Análisis Sintáctico LALR
de la Definición S-atribuida (13)**

	identificador	=	+	*	()	número	\$		S	E	T	F
0	d 2									1			
1								Aceptar					
2		d 3											
3	d 4				d 6		d 5				7	8	9
4			r 8			r 8		r 8					
5			r 7			r 7		r 7					
6	d 4				d 6		d 5				10	8	9
7			d 11			r 1		r 1					
8			r 2	d 12		r 2		r 2					
9			r 4			r 4		r 4					
10			d 11			d 13							
11	d 4				d 6		d 5					14	9
12	d 4				d 6		d 5						15
13			r 6			r 6		r 6					
14			r 3	d 13		r 3		r 3					
15			r 5			r 5		r 5					

Traducción Ascendente de $a = 3 * b + 2$

<i>PILA de análisis sintáctico</i>	<i>ENTRADA</i>	<i>ACCIÓN</i>	<i>PILA DE VALORES</i>	<i>VALORES</i>
0	id = n * id + n \$	d 2		
0 id 2	= n * id + n \$	d 3	a	
0 id 2 = 3	n * id + n \$	d 5	a =	
0 id 2 = 3 número 5	* id + n \$	r 7: F → número	a = 3	
0 id 2 = 3 F 9	* id + n \$	r 4: T → F	a = F. variable F.código	F. variable: t1 F. código: t1 = 3
0 id 2 = 3 T 8	* id + n \$	d 12	a = T. variable T.código	T. variable: t1 T. código: t1 = 3
0 id 2 = 3 T 8 * 12	id + n \$	d 4	a = T. variable T.código *	T. variable: t1 T. código: t1 = 3
0 id 2 = 3 T 8 * 12 id 4	+ n \$	r 8 : F → id	a = T. variable T.código * b	T. variable: t1 T. código: t1 = 3
0 id 2 = 3 T 8 * 12 F 15	+ n \$	r 5 : T → T * F	a = T. variable T.código * F.variable F. código	T. variable: t1 T. código: t1 = 3 F. variable: b F. código:
0 id 2 = 3 T 8	+ n \$	r 2 : E → T	a = E. variable E.código	E. variable: t2 E. código: t1 = 3 t2 = t1 * b

<i>PILA de análisis sintáctico</i>	<i>ENTRADA</i>	<i>ACCIÓN</i>	<i>PILA DE VALORES</i>	<i>VALORES</i>
0 id 2 = 3 E 7	+ n \$	d 11	a = E. variable E.código	E. variable: t2 E. código: t1 = 3 t2 = t1 * b
0 id 2 = 3 E 7 + 11	n \$	d 5	a = E. variable E.código +	E. variable: t2 E. código: t1 = 3 t2 = t1 * b
0 id 2 = 3 E 7 + 11 n 5	\$	r 7 : F → número	a = E. variable E.código + 2	E. variable: t2 E. código: t1 = 3 t2 = t1 * b
0 id 2 = 3 E 7 + 11 F 9	\$	r 4: T → F	a = E. variable E.código + F. variable F.código	E. variable: t2 E. código: t1 = 3 t2 = t1 * b F. variable: t3 F. código: t3 = 2

<i>PILA de análisis sintáctico</i>	<i>ENTRADA</i>	<i>ACCIÓN</i>	<i>PILA DE VALORES</i>	<i>VALORES</i>
0 id 2 = 3 E 7 + 11 T 14	\$	r 4: E → E + T	a = E. variable E.código + T. variable T.código	E. variable: t2 E. código: t1 = 3 t2 = t1 * b T. variable: t3 T. código: t3 = 2
0 id 2 = 3 E 7	\$	r 1: S → id = E	a = E. variable E.código	E. variable: t4 E. código: t1 = 3 t2 = t1 * b t3 = 2 t4 = t2 + t3
0 S 1	\$	Aceptar	S. código	S. código: t1 = 3 t2 = t1 * b t3 = 2 t4 = t2 + t3 a = t4

Definición S-atribuida (13): Generación de código intermedio para expresiones aritméticas

Ejemplo: $a = 3 * b + 2$

- Se genera el código

$t1 = 3$

$t2 = t1 * b$

$t3 = 2$

$t4 = t2 + t3$

$a = t4$

- Este código se puede optimizar:
 - Paso 1: eliminación de asignación de constantes a variables

$t2 = 3 * b$

$t4 = t2 + 2$

$a = t4$

- Paso 2: eliminación de propagación de copias

$t2 = 3 * b$

$a = t2 + 2$