

Tema 7

Análisis Sintáctico Ascendente



Universidad
Europea

LAUREATE INTERNATIONAL UNIVERSITIES

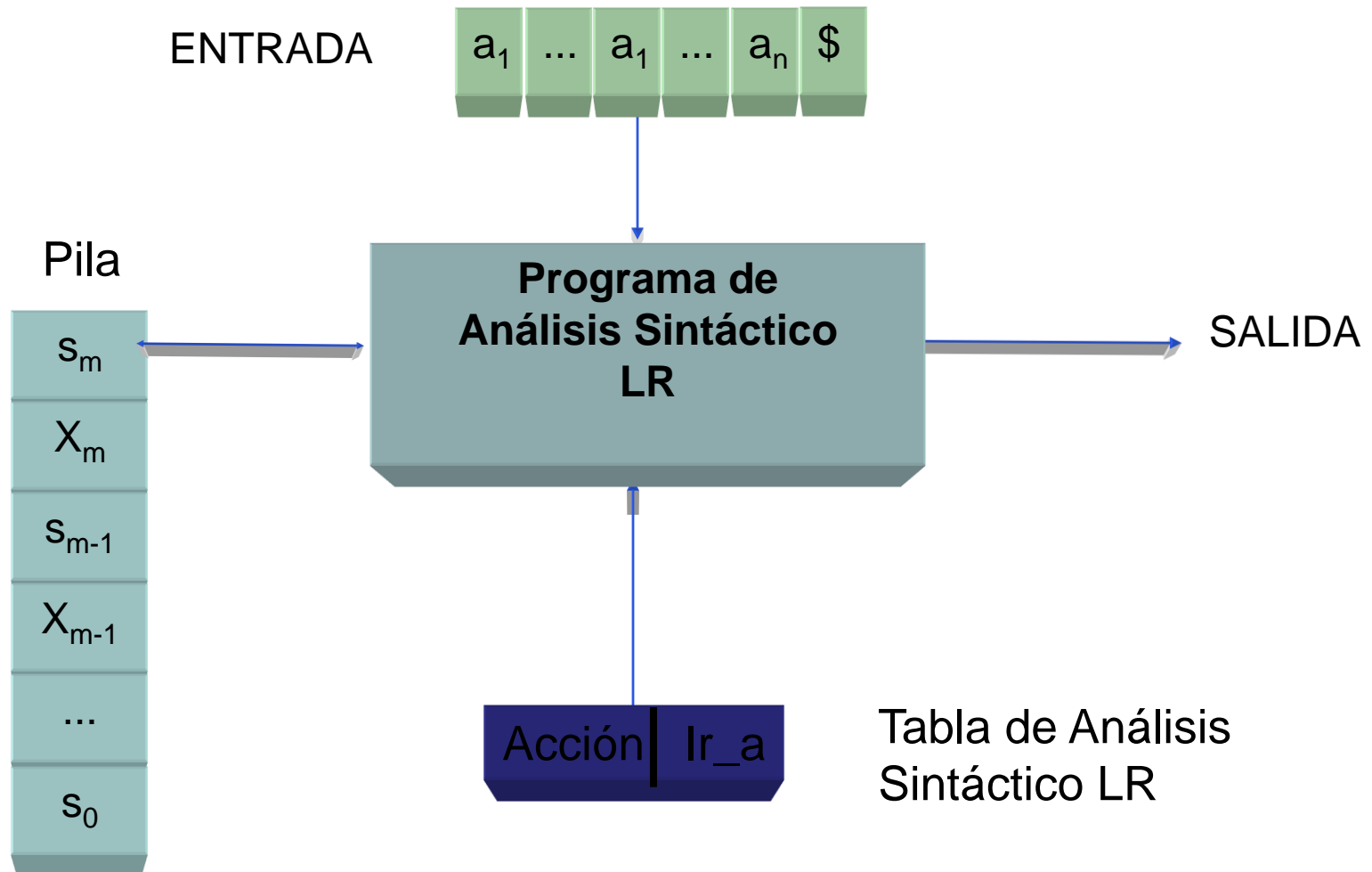
Análisis Ascendente LR

- LR(k): *Left-to-right, rightmost derivation*, k símbolos de entrada son necesarios para tomar las decisiones de análisis sintáctico
 - Ventajas
 - Es el método de análisis por desplazamiento y reducción sin retroceso más general, a pesar de esto es igual de eficiente
 - La clase de gramáticas que pueden analizarse es un supraconjunto de la clase de gramáticas que pueden analizarse con analizadores sintácticos predictivos
 - Detectan los errores sintácticos tan pronto como es posible en un examen de izquierda a derecha de la entrada
 - Se pueden reconocer prácticamente todas las construcciones de los lenguajes de programación descritos por una gramática G2
 - Inconvenientes
 - La construcción “a mano” requiere mucho trabajo

Tipos de Analizadores LR

- LR simple (SLR)
 - Fácil de implementar
 - Menos poderoso, hay algunas gramáticas que los otros métodos pueden analizar y este no puede
- LR(1)
 - Es muy costoso de implementar
 - El más potente
- LALR (LR con examen por anticipado)
 - Intermedio entre los dos métodos anteriores

Modelo de un Analizador LR



Modelo de Analizador LR

- El programa es el mismo para todos los analizadores LR
- X_i es un símbolo gramatical y cada s_i es un símbolo llamado estado
- Se utiliza el símbolo de estado y el símbolo de la entrada para indexar la tabla y determinar la acción siguiente
- La tabla de análisis sintácticos tiene dos partes:
 - $Acción[s_m, a_i]=$
 - Error: error de sintaxis
 - Aceptar: acepta la entrada, el análisis sintáctico finaliza
 - Desplazar: introduce en la pila el símbolo a_i y el estado s_m
 - Reducción: extrae símbolos de la pila, ejecuta la acción semántica correspondiente a una producción
 - $lr_a[s_m, X_i]= s_k$

Modelo de Analizador LR

- Configuración de un analizador sintáctico LR
 - Tupla con el contenido de la pila y la entrada que resta por procesar
 $(s_0 X_1 s_1 X_2 s_2 \dots X_m s_m, a_i a_{i+1} \dots a_n \$)$
- $Acción[s_m, a_i] = \text{desplazar } s$
 - $(s_0 X_1 s_1 X_2 s_2 \dots X_m s_m \mathbf{a_i s}, a_{i+1} \dots a_n \$)$
- $Acción[s_m, a_i] = \text{reducir } A \rightarrow \beta$
 - $(s_0 X_1 s_1 X_2 s_2 \dots X_{m-r} s_{m-r} \mathbf{A s}, a_i a_{i+1} \dots a_n \$)$ donde $s = lr_a[s_{m-r}, A]$
 y $r = |\beta|$ (se extraen r símbolos no terminales y r símbolos de estados de la pila)

Algoritmo de Análisis LR

apuntar ae al primer símbolo de $w\$$ (s está en la cima y ae apunta al símbolo a)

repetir

caso $Acción[s, a]$

Desplazar s'

push a

push s'

leer en la entrada

Reducir $A \rightarrow \beta$

pop $2*|\beta|$ símbolos

s' símbolo en la cima de la pila

$s = lr_a[s', A]$

push A

push s

Aceptar

Error

fincaso

hasta Aceptar o Error

Ejemplo de Análisis LR

- Gramática
- Tabla de análisis sintáctico

1. $E ::= E + T$

2. $E ::= T$

3. $T ::= T * F$

4. $T ::= F$

5. $F ::= (E)$

6. $F ::= Id$

Estado	Acción						lr_a		
	Id	+	*	()	\$	E	T	F
0	d5			d4			1	2	3
1		d6				ACP			
2		r2	d7		r2	r2			
3		r4	r4		r4	r4			
4	d5			d4			8	2	3
5		r6	r6		r6	r6			
6	d5			d4				9	3
7	d5			d4					10
8		d6			d11				
9		r1	d7		r1	r1			
10		r3	r3		r3	r3			
11		r5	r5		r5	r5			

Ejemplo de Análisis LR

Pila	Entrada	Acción
0	Id.*.Id.+.Id.\$	d5
0 Id 5	*.Id.+.Id.\$	r6
0 F 3	*.Id.+.Id.\$	r4
0 T 2	*.Id.+.Id.\$	d7
0 T 2 * 7	Id.+.Id.\$	d5
0 T 2 * 7 Id 5 +.Id.\$		r6
0 T 2 * 7 F 10	+.Id.\$	r3
0 T 2	+.Id.\$	r2
0 E 1	+.Id.\$	d6
0 E 1 + 6	Id.\$	d5
0 E 1 + 6 Id 5	\$	r6
0 E 1 + 6 F 3\$		r4
0 E 1 + 6 T 9\$		r1
0 E 1	\$	ACP

Construcción de Tabla de Análisis SLR



- Definiciones:
 - Elemento del análisis sintáctico LR(0) de una gramática G (elemento, *ítem*)
 - Producción de G con un “**punto**” en alguna posición del lado derecho
 - El punto indica hasta donde se ha visto la producción en un momento del análisis

Ejemplo: La producción $A \rightarrow XYZ$ tiene cuatro elementos:

$A \rightarrow \bullet XYZ$

$A \rightarrow X \bullet YZ$

$A \rightarrow XY \bullet Z$

$A \rightarrow XYZ \bullet$

Pregunta:

¿Qué elemento genera la producción $A \rightarrow \lambda$?

Construcción de Tabla de Análisis SLR

- Definiciones:
 - **Item válido en LR(0) de prefijo viable:**

$$A \rightarrow \beta_1 \bullet \beta_2 \text{ es item válido de } \alpha \beta_1 \text{ sii:}$$

$$\underset{\text{rm}}{S} \xrightarrow{*} \underset{\text{rm}}{\alpha A w} \xrightarrow{*} \alpha \beta_1 \beta_2 w \quad (A \in \Sigma_N, \alpha, \beta_1, \beta_2 \in \Sigma^*, w \in \Sigma_T^*)$$
 - **Estado**
 - Conjunto de *ítems*
 - Dan lugar a los estados del analizador sintáctico SLR
 - El conjunto de estados: ***colección canónica LR(0)***
- Los *ítems* son los estados de un AF que reconoce los prefijos viables
- Los estados son las agrupaciones de estados, (¡La minimización del AF!)

Algoritmo: crear conjunto canónico LR(0)

- Entrada:
 - Gramática ampliada G'
 - Función **cierre**(I) (*clausura* o *cerradura*), $I \equiv \text{ítem}^+$
 - Función **goto**(I, X) (*ir_a*), $X \in (\Sigma_T \cup \Sigma_N)$
- Salida
 - Conjunto canónico LR(0)
- Gramática ampliada G' de G
 - Añadir S' , $\Sigma_N = (\Sigma_N \cup S' \mid S' \text{ es el axioma})$
 - Añadir $S' \rightarrow S$, $P = (P \cup S' \rightarrow S)$

Construir el conjunto canónico LR(0)

- Función ***cierre***(*I*)

function ***cierre***(*I*);

begin

J := *I*;

repeat

for $\forall J_i (A \rightarrow \alpha \bullet B \beta) \in J, \forall p (B \rightarrow \gamma) \in P \mid (B \rightarrow \bullet) \notin J$

do $J := J \cup (B \rightarrow \bullet) ;$

until no pueden añadirse *ítems* a *J*;

return *J*

end

- Ejemplo: $A \rightarrow B$
 $B \rightarrow id \mid C \text{ num} \mid (D)$
 $C \rightarrow + D$
 $D \rightarrow id \mid num$

$\hat{J} \text{ cierre}(A \rightarrow \bullet B) ?$

$A \rightarrow \bullet B$

$B \rightarrow \bullet id \mid \bullet C \text{ num} \mid \bullet (D)$

$C \rightarrow \bullet + D$

Construir el conjunto canónico LR(0)

- Función **goto**(I, X)
 - Si I son items válidos de γ , goto(I, X) son los items válidos de γX

function goto(I, X);

begin

$J := \emptyset$;

$\forall I_i \mid (B \rightarrow \alpha \bullet X \beta) \in I, J := J \cup \text{cierre}(B \rightarrow \alpha X \bullet \beta) ;$

return J

end

- Ejemplo:

$$\begin{aligned}
 A &\rightarrow B \\
 B &\rightarrow id \mid C \text{ num} \mid (D) \\
 C &\rightarrow + D \\
 D &\rightarrow id \mid num
 \end{aligned}$$

$I = \{ B \rightarrow \bullet id, B \rightarrow \bullet (D) \}$
 \hookrightarrow goto $\{I, (\}$?

$B \rightarrow (\bullet D)$

$D \rightarrow \bullet id$

$D \rightarrow \bullet num$

Algoritmo: crear conjunto canónico LR(0)

- La función **elementos** proporciona la colección canónica de conjuntos de elementos de LR(0)

function *elementos*(G');

begin

$C := \text{cierre}(S' \rightarrow \bullet S);$

repeat

for $\forall I \in C, \forall X \mid \text{goto}(I, X) \neq \emptyset, \text{goto}(I, X) \notin C$

do $C := C \cup \text{goto}(I, X);$

until no pueden añadirse más conjuntos de *ítems* a C ;

return C

end

Construir el conjunto canónico LR(0)

Ejemplo:

• G

1. $S \rightarrow A B \text{ end}$
2. $A \rightarrow \text{tipo}$
3. $A \rightarrow \text{id } A$
4. $B \rightarrow \text{begin } C$
5. $C \rightarrow \text{código}$

• G'

1. $S' \rightarrow S$
2. $S \rightarrow A B \text{ end}$
3. $A \rightarrow \text{tipo}$
4. $A \rightarrow \text{id } A$
5. $B \rightarrow \text{begin } C$
6. $C \rightarrow \text{código}$

items LR(0):

$I_0:$ $S' \rightarrow \bullet S$
 $S \rightarrow \bullet A B \text{ end}$
 $A \rightarrow \bullet \text{tipo}$
 $A \rightarrow \bullet \text{id } A$

$I_1:$ $S' \rightarrow S \bullet$

$I_2:$ $S \rightarrow A \bullet B \text{ end}$
 $B \rightarrow \bullet \text{begin } C$

$I_3:$ $A \rightarrow \text{tipo} \bullet$

$I_4:$ $A \rightarrow \text{id} \bullet A$
 $A \rightarrow \bullet \text{tipo}$
 $A \rightarrow \bullet \text{id } A$

$I_5:$ $S \rightarrow A B \bullet \text{end}$

$I_6:$ $B \rightarrow \text{begin} \bullet C$
 $C \rightarrow \bullet \text{código}$

$I_7:$ $A \rightarrow \text{id } A \bullet$

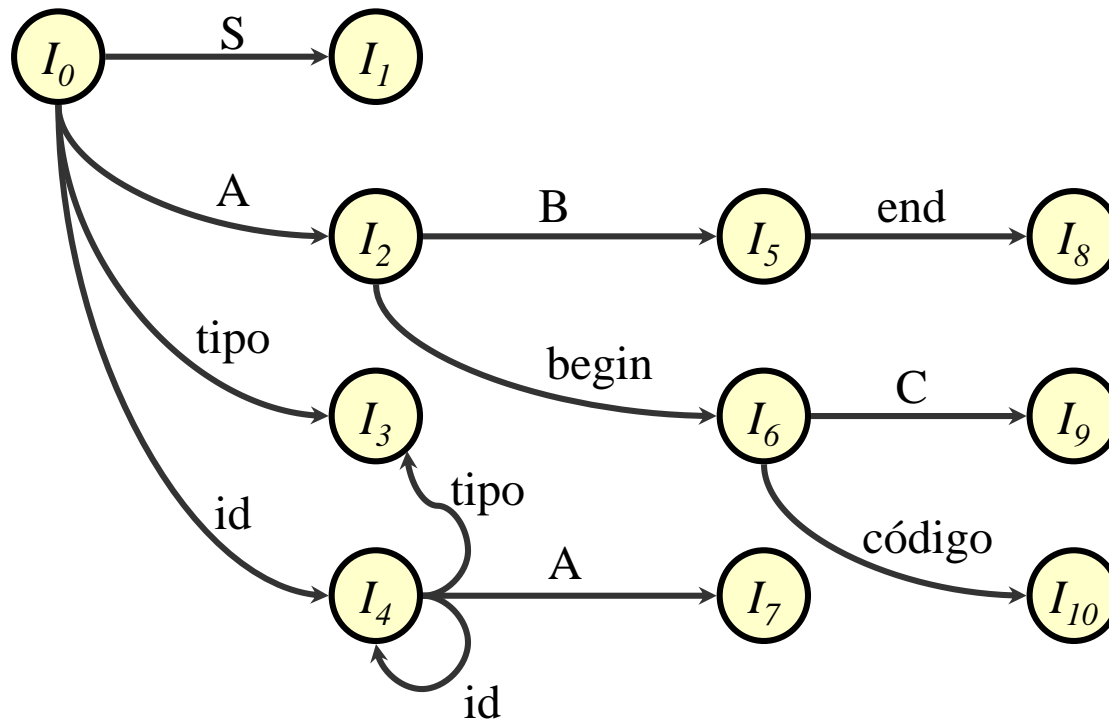
$I_8:$ $S \rightarrow A B \text{ end} \bullet$

$I_9:$ $B \rightarrow \text{begin } C \bullet$

$I_{10}:$ $C \rightarrow \text{código} \bullet$

Construir el conjunto canónico LR(0)

- El conjunto canónico define un AFD que reconoce los prefijos viables de G , con I_0 estado inicial e $I_j \forall j \neq 0$ estados finales



- Para cada prefijo alcanzado, I_i define sus prefijos viables!

Construir la tabla SLR (por fin)

- Entrada:
 - Gramática aumentada G'
- Salida
 - Tabla SLR, funciones de acción e ir_a
- Algoritmo
 1. Construir $C = \{I_0, I_1, \dots, I_n\}$ LR(0) de G'
 2. El estado i se construye a partir de I_i , poner las operaciones adecuadas del analizador sintáctico
 3. Las entradas no definidas son ERROR
 4. El estado inicial es el del conjunto con $[S' \rightarrow \bullet S]$

Construir la tabla SLR

- Operaciones asociadas a los *ítems*
 - Desplazar
 - Si $[A \rightarrow \alpha \bullet a \beta] \in I_i$, $a \in \Sigma_T$, $\text{goto}(I_i, a) = I_j \Rightarrow \text{acción}[i, a] = \text{desplazar } j$
 - Reducir
 - Si $[A \rightarrow \alpha \bullet] \in I_i$, $A \neq S' \Rightarrow \forall a \in \text{SIGUIENTE}(A)$, $\text{acción}[i, a] = \text{reducir } A \rightarrow \alpha$
 - Aceptar
 - Si $[S' \rightarrow S \bullet] \in I_i \Rightarrow \text{acción}[i, \$] = \text{aceptar}$
 - Ir_a
 - Si $\text{goto}(I_i, A) = I_j$, $A \in \Sigma_N \Rightarrow \text{ir_a}[i, A] = j$

Construir la tabla SLR

ccLR(0)		I_5 : $S \rightarrow A B \bullet \text{end}$ acción[5,end]=d8											
I_0 : $S' \rightarrow \bullet S$ ir_a[0,S]=1 $S \rightarrow \bullet A B \text{ end}$ ir_a[0,A]=2 $A \rightarrow \bullet \text{tipo}$ acción[0,tipo]=d3 $A \rightarrow \bullet \text{id } A$ acción[0,id]=d4		I_6 : $B \rightarrow \text{begin} \bullet C$ ir_a[6,C]=9 $C \rightarrow \bullet \text{código}$ acción[6,código]=d10											
		I_7 : $A \rightarrow \text{id } A \bullet$ acción[7,begin]=r(A \rightarrow id)											
		I_8 : $S \rightarrow A B \text{ end} \bullet$ acción[8,\$]=r(S \rightarrow A B end)											
		I_9 : $B \rightarrow \text{begin } C \bullet$ acción[9,end]=r(B \rightarrow begin C)											
acción[1,\$]=aceptar		I_{10} : $C \rightarrow \text{código} \bullet$ acción[10,end]=r(C \rightarrow código)											
I_2 : $S \rightarrow A \bullet B \text{ end}$ ir_a[2,B]=5		<table><tr><th>Σ_N</th><th>Siguiente</th></tr><tr><td>S</td><td>\$</td></tr><tr><td>A</td><td>begin</td></tr><tr><td>B</td><td>end</td></tr><tr><td>C</td><td>end</td></tr></table>		Σ_N	Siguiente	S	\$	A	begin	B	end	C	end
Σ_N	Siguiente												
S	\$												
A	begin												
B	end												
C	end												
$B \rightarrow \bullet \text{begin } C$ acción[2,begin]=d6													
I_3 : $A \rightarrow \text{tipo} \bullet$ acción[3,begin]=r(A \rightarrow tipo)													
I_4 : $A \rightarrow \text{id} \bullet A$ ir_a[4,A]=7 $A \rightarrow \bullet \text{tipo}$ acción[4,tipo]=d3 $A \rightarrow \bullet \text{id } A$ acción[4,id]=d4													

Tabla SLR Resultante

acción							ir_a			
	tipo	id	begin	código	\$	end	S	A	B	C
0	d3	d4					1	2		
1					ACP					
2			d6						5	
3			r(A → tipo)							
4	d3	d4						7		
5						d8				
6				d10						9
7			r(A → id)							
8					r(S → AB end)					
9						r(B → begin C)				
10						r(C → código)				

Resumen

- El análisis sintáctico LR
 - El análisis ascendente sin retroceso más general
 - Utiliza una pila y una tabla de análisis
 - Desplazamiento/Reducción
 - La tabla SLR se obtiene haciendo el conjunto canónico LR(0)

Analizador canónico LR

- Motivación
 - Tabla SLR tiene info insuficiente y pueden aparecer conflictos desplazamiento/reducción
 - Si el estado I_i contiene $[A \rightarrow \beta \bullet]$ No es adecuada siempre la reducción $A \rightarrow a$ para todos los terminales en $\text{Siguiente}(A)$
 - Dado prefijo viable $\alpha\beta$ con items válidos en I_i puede ser que no existan formas sentenciales αAa , con $a \in \text{Siguiente}(A)$

Ejemplo:

$S \rightarrow L=R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow \text{id}$

$R \rightarrow L$

Conjunto LR(0)

Ejemplo:

• G

1. $S \rightarrow L=R$
2. $S \rightarrow R$
3. $L \rightarrow *R$
4. $L \rightarrow id$
5. $R \rightarrow L$

• G'

1. $S \rightarrow L=R$
2. $S \rightarrow R$
3. $L \rightarrow *R$
4. $L \rightarrow id$
5. $R \rightarrow L$
6. $S' \rightarrow S$

items LR(0):

$I_0:$ $S' \rightarrow \bullet S$
 $S \rightarrow \bullet L=R$
 $S \rightarrow \bullet R$
 $R \rightarrow \bullet L$
 $L \rightarrow \bullet *R$
 $L \rightarrow \bullet id$

$I_1:$ $S' \rightarrow S \bullet$

$I_2:$ $S \rightarrow L \bullet =R$
 $R \rightarrow L \bullet$

$I_3:$ $S \rightarrow R \bullet$

$I_4:$ $L \rightarrow * \bullet R$
 $R \rightarrow \bullet L$
 $L \rightarrow \bullet *R$
 $L \rightarrow \bullet id$

$I_5:$ $L \rightarrow id \bullet$

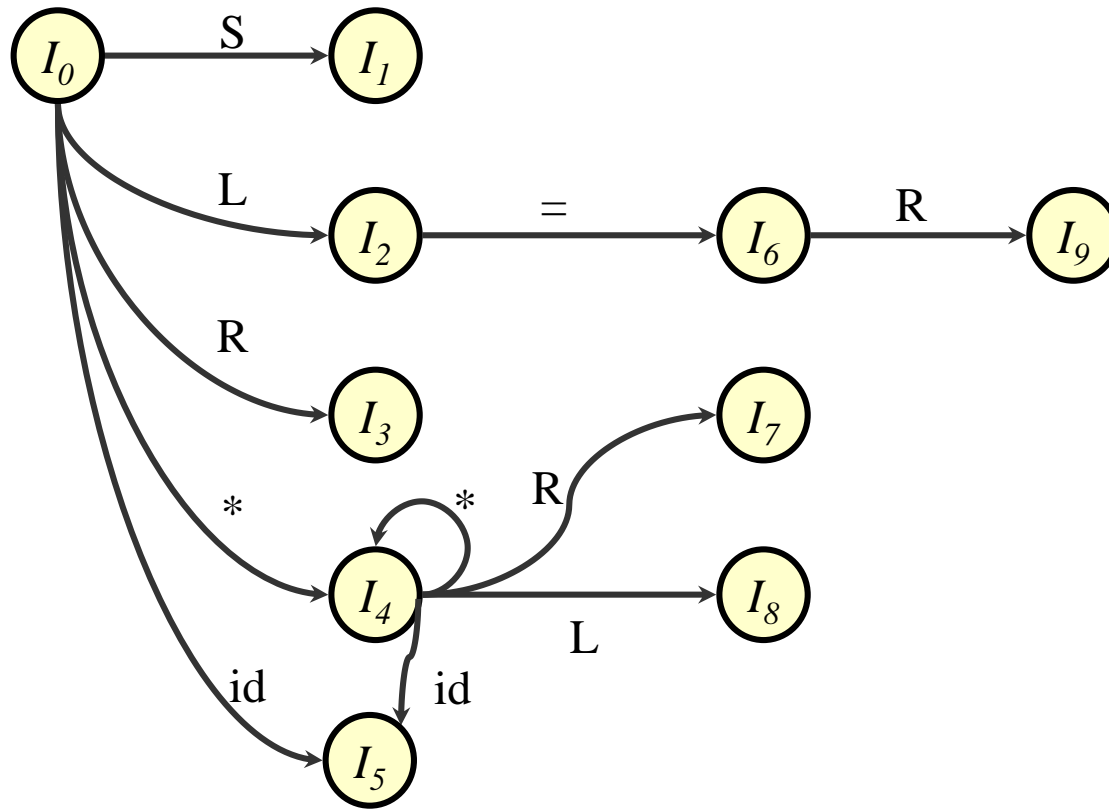
$I_6:$ $S \rightarrow L = \bullet R$
 $R \rightarrow \bullet L$
 $L \rightarrow \bullet *R$
 $L \rightarrow \bullet id$

$I_7:$ $L \rightarrow * R \bullet$

$I_8:$ $R \rightarrow L \bullet$

$I_9:$ $S \rightarrow L = R \bullet$

Tabla de análisis LR(0)



Conflicto en la tabla SLR!

ccLR(0)	acción
....	
$I_2: S \rightarrow L \bullet =R$	$\text{acción}[2,]=d6$
$R \rightarrow L \bullet$	$\text{acción}[2,]=r5$
...	

- \$ está en $\text{Siguiente}(R)$ pero no hay ninguna derivación donde reducir L por R seguido de \$
- Esta información de contexto no se usa en el analizador SLR

Σ_N	Siguiente
S	\$
L	\$, =
R	\$, =

Items en LR(1)

- *Ítems*: estados de AF que reconoce los prefijos viables junto a símbolos posibles tras pivote
 - **Elementos con más información:**
 $[A \rightarrow \beta_1 \bullet \beta_2, a]$ con $A \rightarrow \beta_1 \bullet \beta_2 \in P$, $a \in \Sigma_T$
 - **Definición:**
 $[A \rightarrow \beta_1 \bullet \beta_2, a]$ es ítem válido de $\gamma = \alpha \beta_1$ sii:
 - $S \xrightarrow{*} \underset{rm}{\alpha} A \underset{rm}{w} \rightarrow \alpha \beta_1 \beta_2 w$
 - a es el primer símbolo de w (o si w es λ , a es $\$$)
 - **Estado:**
 - Conjunto de estados: ***colección canónica LR(1)***
 - Se aumentan los estados LR(0)

Items en LR(1)

- Construcción: varía la función cierre(I)
 - $[A \rightarrow \alpha \bullet B \beta, a] \in I$
 $[A \rightarrow \alpha \bullet B \beta, a]$ es item válido de $\gamma = \delta \alpha$:

$$S \xrightarrow{*}_{rm} \delta A a x \rightarrow \delta \alpha B \beta a x = \gamma B \beta a x$$
 - **Para cada item $B \rightarrow \eta$:**

$$S \xrightarrow{*}_{rm} \gamma \eta b y, \quad \forall b | b \in \text{Primero}(\beta a)$$

$$[B \rightarrow \bullet \eta, b] \in \text{Primero}(I)$$

Construir el conjunto canónico LR(1)

- Función **cierre**(*I*)

function **cierre**(*I*);

begin

J := *I*;

repeat

for $\forall J_i [A \rightarrow \alpha \bullet B \beta, a] \in J, \forall p (B \rightarrow \gamma) \in P, \forall b \in \text{Primero}(\beta a) \mid$
 $[B \rightarrow \bullet \gamma, b] \notin J$

do $J := J \cup [B \rightarrow \bullet \gamma, b];$

until no pueden añadirse ítems a *J*;

return *J*

end

Ejemplo:

$S' \rightarrow S$

$S \rightarrow CC$

$C \rightarrow cC \mid d$

¿ **cierre**($[S' \rightarrow \bullet S, \$]$)?

$[S' \rightarrow \bullet S, \$]$

$[S \rightarrow \bullet CC, \$]$

$[C \rightarrow \bullet cC, c/d]$

$[C \rightarrow \bullet d, c/d]$

	Primero
\$	\$
C\$	c, d

Construir el conjunto canónico LR(1)

- Función **goto**(I, X)
 - Si I son items válidos de γ , goto(I, X) son los items válidos de γX

function goto(I, X);

begin

$J := \emptyset$;

$\forall I_i \mid [B \rightarrow \alpha \bullet X \beta, a] \in I, J := J \cup \mathbf{cierre}([B \rightarrow \alpha X \bullet \beta, a])$;

return J

end

Ejemplo:

$S' \rightarrow S$
 $S \rightarrow CC$
 $C \rightarrow cC \mid d$

i goto($[C \rightarrow \bullet cC, c/d], c$)?

$[C \rightarrow c \bullet C, c/d]$

$[C \rightarrow \bullet cC, c/d]$

$[C \rightarrow \bullet d, c/d]$

	Primero
$(\lambda) c$	c
$(\lambda) d$	d

Algoritmo: crear conjunto canónico LR(1)

- La función **elementos** proporciona la colección canónica de conjuntos de elementos de LR(1)

function *elementos*(G');

begin

$C := \text{cierre}([S' \rightarrow \bullet S, \$]);$

repeat

for $\forall I \in C, \forall X \mid \text{goto}(I, X) \neq \emptyset, \text{goto}(I, X) \notin C$

do $C := C \cup \text{goto}(I, X);$

until no pueden añadirse más conjuntos de *ítems* a C ;

return C

end

Conj. LR(1)

• G'

1. $S \rightarrow L=R$
2. $S \rightarrow R$
3. $L \rightarrow *R$
4. $L \rightarrow id$
5. $R \rightarrow L$
6. $S' \rightarrow S$

items LR(1):

$I_0:$ $[S' \rightarrow \bullet S, \$]$
 $[S \rightarrow \bullet L=R, \$]$
 $[S \rightarrow \bullet R, \$]$
 $[L \rightarrow \bullet *R, =]$
 $[L \rightarrow \bullet id, =]$
 $[R \rightarrow \bullet L, \$]$

$I_1:$ $[S' \rightarrow S \bullet, \$]$

$I_2:$ $[S \rightarrow L \bullet =R, \$]$
 $[R \rightarrow L \bullet, \$]$

$I_3:$ $[S \rightarrow R \bullet, \$]$

$I_4:$ $[L \rightarrow * \bullet R, =]$
 $[R \rightarrow \bullet L, =]$
 $[L \rightarrow \bullet *R, =]$
 $[L \rightarrow \bullet id, =]$

$I_5:$ $[L \rightarrow id \bullet, =]$

$I_6:$ $[S \rightarrow L = \bullet R, \$]$
 $[R \rightarrow \bullet L, \$]$
 $[L \rightarrow \bullet *R, \$]$
 $[L \rightarrow \bullet id, \$]$

$I_7:$ $[L \rightarrow *R \bullet, =]$

$I_8:$ $[R \rightarrow L \bullet, =]$

$I_9:$ $[S \rightarrow L = R \bullet, \$]$

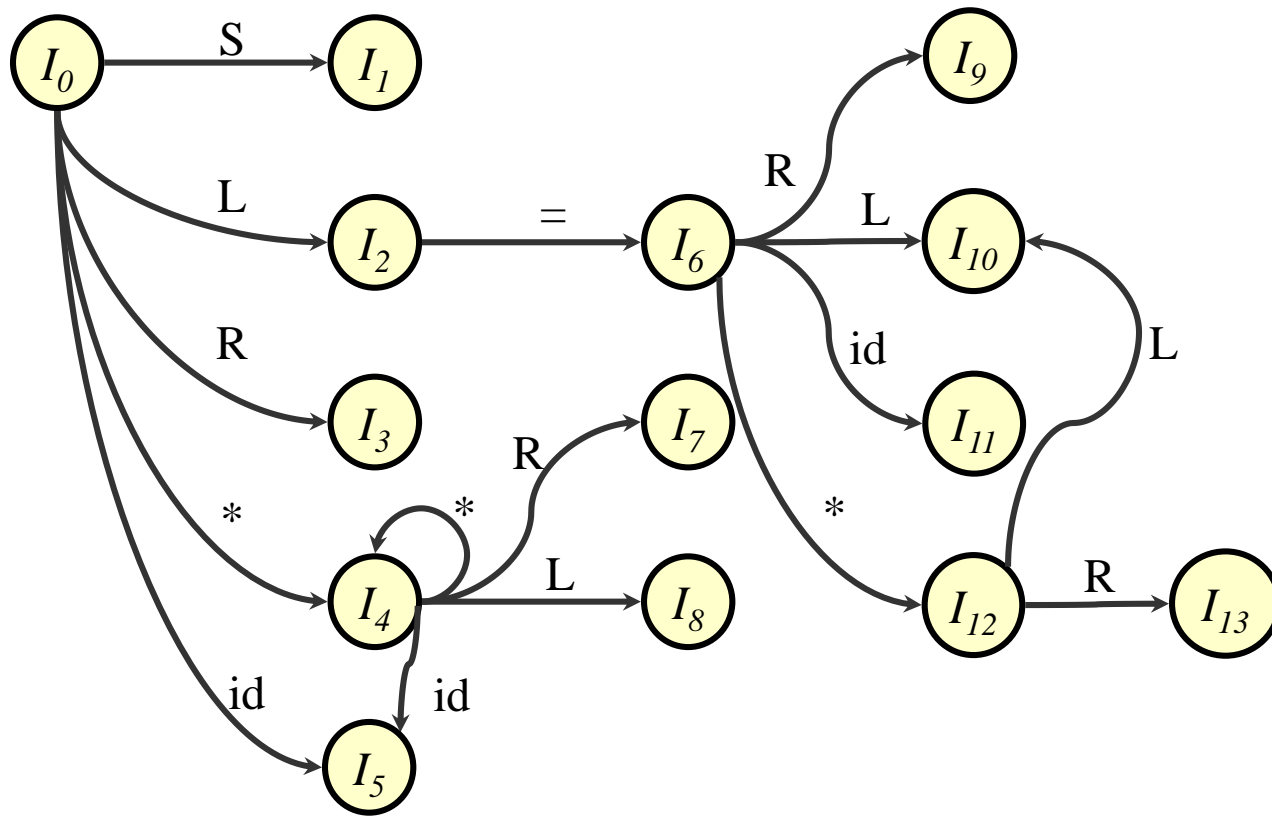
$I_{10}:$ $[R \rightarrow L \bullet, \$]$

$I_{11}:$ $[L \rightarrow id \bullet, \$]$

$I_{12}:$ $[L \rightarrow * \bullet R, \$]$
 $[R \rightarrow \bullet L, \$]$
 $[L \rightarrow \bullet *R, \$]$
 $[L \rightarrow \bullet id, \$]$

$I_{13}:$ $[L \rightarrow *R \bullet, \$]$

Estados de analizador canónico LR(1)



Construir la tabla canónica LR

- Entrada:
 - Gramática aumentada G'
- Salida
 - Tabla LR, funciones de acción e ir_a
- Algoritmo
 1. Construir $C = \{I_0, I_1, \dots, I_n\}$ LR(1) de G'
 2. El estado i se construye a partir de I_i , poner las operaciones adecuadas del analizador sintáctico
 3. Las entradas no definidas son ERROR
 4. El estado inicial es el del conjunto con $[S' \rightarrow \bullet S, \$]$

Construir la tabla canónica LR

accLR(1)			
$I_0: [S' \rightarrow \bullet S, \$]$ $[S \rightarrow \bullet L = R, \$]$ $[S \rightarrow \bullet R, \$]$ $[L \rightarrow \bullet * R, =]$ $[L \rightarrow \bullet id, =]$ $[R \rightarrow \bullet L, \$]$	$ir_a(0, S) = 1$ $ir_a(0, L) = 2$ $ir_a(0, R) = 3$ $acc(0, *) = d4$ $acc(0, id) = d5$	$I_6: [S \rightarrow L = \bullet R, \$]$ $[R \rightarrow \bullet L, \$]$ $[L \rightarrow \bullet * R, \$]$ $[L \rightarrow \bullet id, \$]$	$ir_a(6, R) = 9$ $ir_a(6, L) = 10$ $acc(6, *) = d12$ $acc(6, id) = d11$
		$I_7: [L \rightarrow * R \bullet, =]$	$acc(7, =) = r(L \rightarrow * R)$
		$I_8: [R \rightarrow L \bullet, =]$	$acc(8, =) = r(R \rightarrow L)$
		$I_9: [S \rightarrow L = R \bullet, \$]$	$acc(9, \$) = r(S \rightarrow L = R)$
		$I_{10}: [R \rightarrow L \bullet, \$]$	$acc(10, \$) = r(R \rightarrow L)$
		$I_{11}: [L \rightarrow id \bullet, \$]$	$acc(11, \$) = r(L \rightarrow id)$
$I_1: [S' \rightarrow S \bullet, \$]$ $I_2: [S \rightarrow L \bullet = R, \$]$ $[R \rightarrow L \bullet, \$]$	$acc(1, \$) = ACP$ $acc(2, =) = d6$ $acc(2, \$) = r(R \rightarrow L)$	$I_{12}: [L \rightarrow * \bullet R, \$]$ $[R \rightarrow \bullet L, \$]$ $[L \rightarrow \bullet * R, \$]$ $[L \rightarrow \bullet id, \$]$	$ir_a(12, R) = 13$ $ir_a(12, L) = 10$ $acc(12, *) = d12$ $acc(12, id) = d11$
$I_3: [S \rightarrow R \bullet, \$]$	$acc(3, \$) = r(S \rightarrow R)$	$I_{13}: [L \rightarrow * R \bullet, \$]$	$acc(13, \$) = r(L \rightarrow * R)$
$I_4: [L \rightarrow * \bullet R, =]$ $[R \rightarrow \bullet L, =]$ $[L \rightarrow \bullet * R, =]$ $[L \rightarrow \bullet id, =]$	$ir_a(4, R) = 7$ $ir_a(4, L) = 8$ $acc(4, *) = d4$ $acc(4, id) = d5$		
$I_5: [L \rightarrow id \bullet, =]$	$acc(5, =) = r(L \rightarrow id)$		

Analizador LALR

- Motivación
 - Utilizado en la práctica al reducir el número de estados del analizador canónico LR (mismo número que SLR)
- Forma estados a partir de la unión de estados de LR(1) con el mismo núcleo
 - Si estado I_i contiene $[A \rightarrow \alpha \cdot \beta, \alpha]$, estado I_j contiene $[A \rightarrow \alpha \cdot \beta, b]$ forma estado unión I_{ij} con $[A \rightarrow \alpha \cdot \beta, a/b]$
- Las gramáticas LALR(1) son un subconjunto de LR(1).
 - Pueden inducirse conflictos reducción/reducción pero nunca desplazamiento/reducción
 - Algunos errores se manifiestan más tarde

Construir la tabla LALR

- Algoritmo
 1. Construir $C = \{I_0, I_1, \dots, I_n\}$ LR(1) de G'
 2. Para cada núcleo en el conjunto de items LR(1), buscar todos los conjuntos con ese núcleo y reemplazarlos por su estado unión
 3. $C' = \{J_0, J_1, \dots, J_m\}$ pasan a ser los nuevos estados. Las acciones se generan igual que en el analizador LR. Si hay conflictos, la gramática no es LALR(1)
 4. La tabla ir_a se forma sobre los conjuntos resultantes de efectuar las uniones de elementos LR(1) con el mismo núcleo

Ejemplo

Unir estados:

- 4,12
- 8,10
- 5,11
- 7,13

estados LALR(1):

$I_0:$ $[S' \rightarrow \bullet S, \$]$
 $[S \rightarrow \bullet L = R, \$]$
 $[S \rightarrow \bullet R, \$]$
 $[L \rightarrow \bullet * R, =]$
 $[L \rightarrow \bullet id, =]$
 $[R \rightarrow \bullet L, \$]$

$I_1:$ $[S' \rightarrow S \bullet, \$]$

$I_2:$ $[S \rightarrow L \bullet = R, \$]$
 $[R \rightarrow L \bullet, \$]$

$I_3:$ $[S \rightarrow R \bullet, \$]$

$I_{4_12}:$

$[L \rightarrow * \bullet R, =/\$]$
 $[R \rightarrow \bullet L, =/\$]$
 $[L \rightarrow \bullet * R, =/\$]$
 $[L \rightarrow \bullet id, =/\$]$

$I_{5_11}:$

$[L \rightarrow id \bullet, =/\$]$

$I_6:$ $[S \rightarrow L = \bullet R, \$]$

$[R \rightarrow \bullet L, \$]$
 $[L \rightarrow \bullet * R, \$]$
 $[L \rightarrow \bullet id, \$]$

$I_{7_13}:$

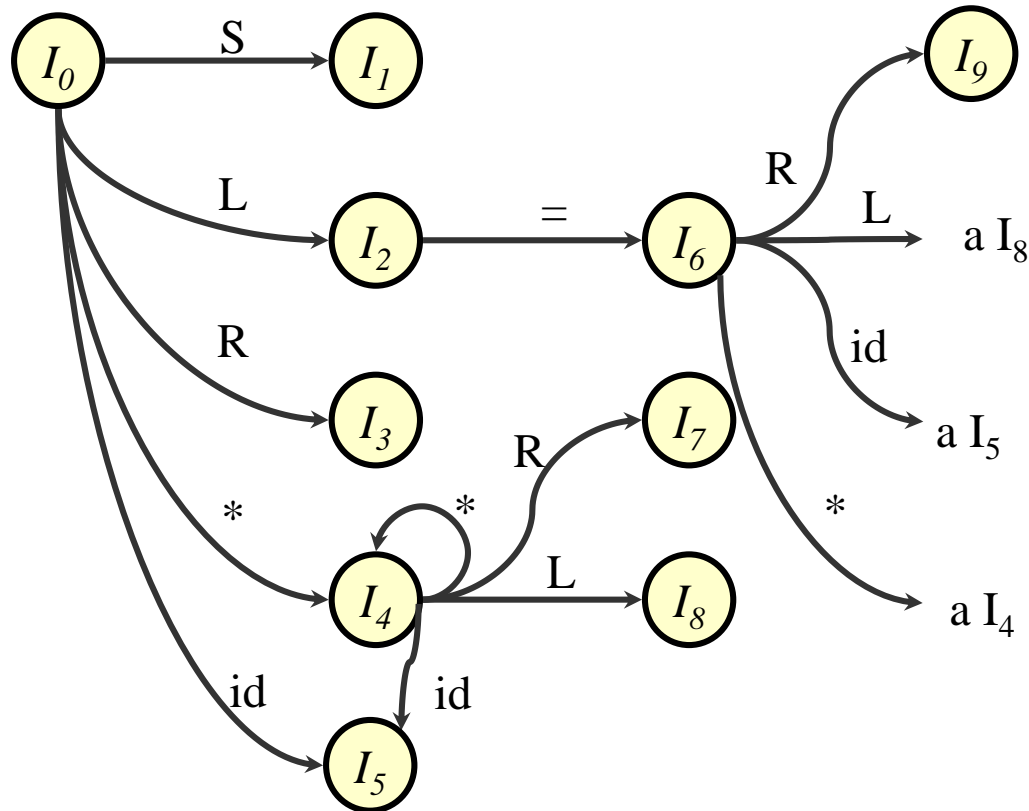
$[L \rightarrow * R \bullet, =/\$]$

$I_{8_10}:$

$[R \rightarrow L \bullet, =/\$]$

$I_9:$ $[S \rightarrow L = R \bullet, \$]$

Estados de analizador LALR(1)



◆ 10 estados, igual que SLR(1)

Construire la tabla LALR

accLR(1)

$I_0: [S' \rightarrow \bullet S, \$]$ $ir_a(0, S)=1$
 $[S \rightarrow \bullet L=R, \$]$ $ir_a(0, L)=2$
 $[S \rightarrow \bullet R, \$]$ $ir_a(0, R)=3$
 $[L \rightarrow \bullet *R, =]$ $acc(0, *) = d4$
 $[L \rightarrow \bullet id, =]$ $acc(0, id)=d5$
 $[R \rightarrow \bullet L, \$]$

$I_1: [S' \rightarrow S \bullet, \$]$ $acc(1, \$)=ACP$

$I_2: [S \rightarrow L \bullet =R, \$]$ $acc(2, =)=d6$
 $[R \rightarrow L \bullet, \$]$ $acc(2, \$)=r(R \rightarrow L)$

$I_3: [S \rightarrow R \bullet, \$]$ $acc(3, \$)=r(S \rightarrow R)$

$I_4: [L \rightarrow * \bullet R, =/\$]$ $ir_a(4, R)=7$
 $[R \rightarrow \bullet L, =/\$]$ $ir_a(4, L)=8$
 $[L \rightarrow \bullet *R, =/\$]$ $acc(4, *) = d4$
 $[L \rightarrow \bullet id, =/\$]$ $acc(4, id)=d5$

$I_5: [L \rightarrow id \bullet, =/\$]$ $acc(5, =)=r(L \rightarrow id)$
 $acc(5, \$)=r(L \rightarrow id)$

$I_6: [S \rightarrow L = \bullet R, \$]$ $ir_a(6, R)=9$
 $[R \rightarrow \bullet L, \$]$ $ir_a(6, L)=10$
 $[L \rightarrow \bullet *R, \$]$ $acc(6, *) = d12$
 $[L \rightarrow \bullet id, \$]$ $acc(6, id)=d11$

$I_7: [L \rightarrow *R \bullet, =/\$]$ $acc(7, =)=r(L \rightarrow R)$
 $acc(7, \$)=r(L \rightarrow R)$

$I_8: [R \rightarrow L \bullet, =/\$]$ $acc(8, =)=r(R \rightarrow L)$
 $acc(8, \$)=r(R \rightarrow L)$

$I_9: [S \rightarrow L = R \bullet, \$]$ $acc(9, \$)=r(S \rightarrow L=R)$

Uso de gramáticas ambiguas

- Una gramática ambigua nunca puede ser LR
- A veces es útil emplear una gramática ambigua:
 - Construcciones más naturales y concisas
 - Aislar casos particulares que puede ser útil tenerlos separados
- Una gramática ambigua puede generar lenguaje con reglas para “deshacer la ambigüedad”
 - Idea similar a las reglas de prioridad en gramáticas de operador
 - Las gramáticas ambiguas sólo deben usarse de forma escasa y controlada, para asegurar qué lenguaje se reconoce

Ejemplo 1ª gramática ambigua

- Sintaxis de condicional:
 - $S \rightarrow \text{if } E \text{ then } S \text{ else } S$
 - $S \rightarrow \text{if } E \text{ then } S$
 - $S \rightarrow \text{other}$
- Gramática ambigua. Versión no Ambigua:
 - $S \rightarrow S_emparejada \mid S_no_emparejada$
 - $S_emparejada \rightarrow \text{if } E \text{ then } S_emparejada \text{ else } S_emparejada$
 - $S_emparejada \rightarrow \text{other}$
 - $S_no_emparejada \rightarrow \text{if } E \text{ then } S$
 - $S_no_emparejada \rightarrow \text{if } E \text{ then } S_emparejada \text{ else } S_no_emparejada$

Conj. LR(0)

- G'

1. $S \rightarrow iSeS$
2. $S \rightarrow iS$
3. $S \rightarrow a$
4. $S' \rightarrow S$

items LR(0):

I_0 : $[S' \rightarrow \bullet S]$

$[S \rightarrow \bullet iSeS]$

$[S \rightarrow \bullet iS]$

$[S \rightarrow \bullet a]$

I_1 : $[S' \rightarrow S \bullet]$

I_2 : $[S \rightarrow i \bullet SeS]$

$[S \rightarrow i \bullet S]$

$[S \rightarrow \bullet iSeS]$

$[S \rightarrow \bullet iS]$

$[S \rightarrow \bullet a]$

I_3 : $[S \rightarrow a \bullet]$

I_4 : $[S \rightarrow iS \bullet eS]$

$[S \rightarrow iS \bullet]$

I_5 : $[S \rightarrow iSe \bullet S]$

$[S \rightarrow \bullet iSeS]$

$[S \rightarrow \bullet iS]$

$[S \rightarrow \bullet a]$

I_6 : $[S \rightarrow iSeS \bullet]$

Acciones analizador SLR(1)

Items LR(0)-acciones SLR (1):

$I_0:$	$[S' \rightarrow \bullet S]$	
	$[S \rightarrow \bullet iSeS]$	$\text{acc}(0,i)=d2$
	$[S \rightarrow \bullet iS]$	
	$[S \rightarrow \bullet a]$	$\text{acc}(0,a)=d3$
$I_1:$	$[S' \rightarrow S \bullet]$	$\text{acc}(1,\$)=ACP$
$I_2:$	$[S \rightarrow i \bullet SeS]$	
	$[S \rightarrow i \bullet S]$	
	$[S \rightarrow \bullet iSeS]$	$\text{acc}(2,i)=d2$
	$[S \rightarrow \bullet iS]$	
	$[S \rightarrow \bullet a]$	$\text{acc}(2,a)=d3$
$I_3:$	$[S \rightarrow a \bullet]$	$\text{acc}(3,\$)=r3$
		$\text{acc}(3,e)=r3$

$I_4:$	$[S \rightarrow iS \bullet eS]$	$\text{acc}(4,e)=d5$
	$[S \rightarrow iS \bullet]$	$\text{acc}(4,e)=r2$
		$\text{acc}(4,\$)=r2$
$I_5:$	$[S \rightarrow iSe \bullet S]$	
	$[S \rightarrow \bullet iSeS]$	$\text{acc}(5,i)=d2$
	$[S \rightarrow \bullet iS]$	
	$[S \rightarrow \bullet a]$	$\text{acc}(5,i)=d3$
$I_6:$	$[S \rightarrow iSeS \bullet]$	$\text{acc}(6,e)=r1$
		$\text{acc}(6,\$)=r1$

Σ_N	Siguiente
S	\$,e

Tabla SLR sin conflictos

Resolución conflicto: en estado 4, desplazamiento sobre “else” (prioridad para if más anidado)

	acción				ir_a
	i	e	a	\$	
0	d2		d3		1
1				ACP	
2	d2		d3		4
3		r3		r3	
4		d5		r2	
5	d2		d3		
6		r1		r1	6

G:

1. $S \rightarrow iSeS$
2. $S \rightarrow iS$
3. $S \rightarrow a$

Ejemplo 2ª gramática ambigua

- Gramática de expresiones con sumas y productos:
$$E ::= E \cdot + \cdot T \mid T$$
$$T ::= T \cdot * \cdot F \mid F$$
$$F ::= (\cdot E \cdot) \mid \text{Id}$$
- Versión ambigua:
$$E ::= E \cdot + \cdot E \mid E \cdot * \cdot E \mid (\cdot E \cdot) \mid \text{Id}$$
- Dos ventajas:
 - Más intuitiva
 - Analizador más rápido al evitar reducciones $F \rightarrow \text{Id}$, $T \rightarrow F$
- Deshacer la ambigüedad con LR equivale aquí a fijar externamente la tabla de precedencia con analizador de precedencia

Conj. LR(0)

• G'

1. $E \rightarrow E + E$
2. $E \rightarrow E * E$
3. $E \rightarrow (E)$
4. $E \rightarrow \text{Id}$
5. $E' \rightarrow E$

I_0 : $[E' \rightarrow \bullet E]$
 $[E \rightarrow \bullet E + E]$
 $[E \rightarrow \bullet E * E]$
 $[E \rightarrow \bullet (E)]$
 $[E \rightarrow \bullet \text{Id}]$

I_1 : $[E' \rightarrow E \bullet]$
 $[E \rightarrow E \bullet + E]$
 $[E \rightarrow E \bullet * E]$

I_2 : $[E \rightarrow (\bullet E)]$
 $[E \rightarrow \bullet E + E]$
 $[E \rightarrow \bullet E * E]$
 $[E \rightarrow \bullet (E)]$
 $[E \rightarrow \bullet \text{Id}]$

I_3 : $[E \rightarrow \text{Id} \bullet]$

I_4 : $[E \rightarrow E + \bullet E]$
 $[E \rightarrow \bullet E + E]$
 $[E \rightarrow \bullet E * E]$
 $[E \rightarrow \bullet (E)]$
 $[E \rightarrow \bullet \text{Id}]$

I_5 : $[E \rightarrow E * \bullet E]$
 $[E \rightarrow \bullet E + E]$
 $[E \rightarrow \bullet E * E]$
 $[E \rightarrow \bullet (E)]$
 $[E \rightarrow \bullet \text{Id}]$

I_6 : $[E \rightarrow (E \bullet)]$
 $[E \rightarrow E \bullet + E]$
 $[E \rightarrow E \bullet * E]$

I_7 : $[E \rightarrow E + E \bullet]$
 $[E \rightarrow E \bullet + E]$
 $[E \rightarrow E \bullet * E]$

I_8 : $[E \rightarrow E * E \bullet]$
 $[E \rightarrow E \bullet + E]$
 $[E \rightarrow E \bullet * E]$

I_9 : $[E \rightarrow (E) \bullet]$

Conflictos en analizador SLR

• G'

1. $E \rightarrow E + E$
2. $E \rightarrow E * E$
3. $E \rightarrow (E)$
4. $E \rightarrow \text{Id}$
5. $E' \rightarrow E$

Σ_N	Siguiente
S	\$, +, *,)

$I_7: [E \rightarrow E + E \bullet]$

$[E \rightarrow E \bullet + E]$

$[E \rightarrow E \bullet * E]$

$\text{acc}(7, \$) = r1$

$\text{acc}(7, +) = r1$

$\text{acc}(7, *) = r1$

$\text{acc}(7,) = r1$

$\text{acc}(7, +) = d4$

$\text{acc}(7, *) = d5$

$I_8: [E \rightarrow E * E \bullet]$

$[E \rightarrow E \bullet + E]$

$[E \rightarrow E \bullet * E]$

$\text{acc}(8, \$) = r2$

$\text{acc}(8, +) = r2$

$\text{acc}(8, *) = r2$

$\text{acc}(8,) = r2$

$\text{acc}(8, +) = d4$

$\text{acc}(8, *) = d5$

Tabla SLR sin conflictos

Resolución conflictos (desplaz/reducción) sobre +,*

{+,*} asociativos por la izquierda

“*” más prioridad que “+”

• **estado I_7 :**

- **acc(7,+)=r1 (“+” es asociativo por izda)**
- **acc(7,*)=d5 (“*” mayor prioridad que “+”)**

• **estado I_8 :**

- **acc(8,+)=r2 (“*” mayor prioridad que “+”)**
- **acc(8,*)=r2 (“*” es asociativo por izda)**

Tabla SLR sin conflictos

	acción						ir_a
	id	+	*	()	\$	
0	d3			d2			1
1		d4	d5			ACP	
2	d3			d2			6
3		r4	r4			r4 r4	
4	d3			d2			7
5	d3			d2			8
6		d4	d5			d9	
7		r1	d5			r1 r1	
8		r2	r2			r2 r2	
9		r3	r3			r3 r3	

G:

1. $E \rightarrow E + E$
2. $E \rightarrow E * E$
3. $E \rightarrow (E)$
4. $E \rightarrow \text{Id}$