

Diseño de Compiladores I

Análisis Sintáctico
Parsing Ascendente

Parsing Ascendente Predictivo

LR(1)

- ▶ Va del programa a la hipótesis
- ▶ El programa se lee de izquierda a derecha (**L**).
- ▶ Las reglas se leen de derecha a izquierda (**R**): el lado derecho se reemplaza por el izquierdo.



Parsing Ascendente Predictivo

LR(1)

- ▶ No impone restricciones de recursividad a la gramática (si fuera a derecha, la consecuencia será una pila más grande).
- ▶ **La única restricción es que la regla de inicio tenga una sola definición.**

- Ejemplo:

$$E \rightarrow E + T$$

$$E \rightarrow E - T$$

- Solución: $E' \rightarrow E$



Parsing Ascendente Predictivo

LR(1)

- ▶ La estrategia de parsing se basa en el grado de cumplimiento de las reglas.

- ▶ Dada la regla:

- ❖ $E \rightarrow E + T$

Existen 4 posibles grados de cumplimiento:

- ❖ $E \rightarrow .E + T$

- ❖ $E \rightarrow E. + T$

- ❖ $E \rightarrow E + .T$

- ❖ $E \rightarrow E + T.$



Parsing Ascendente Predictivo

LR(1)

Gramática

- 1) $A \rightarrow \text{id} := E$
- 2) $E \rightarrow E + T$
- 3) $E \rightarrow T$
- 4) $T \rightarrow T * F$
- 5) $T \rightarrow F$
- 6) $F \rightarrow \text{id}$
- 7) $F \rightarrow \text{cte}$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

- ▶ Se construyen conjuntos de items.
Cada conjunto representa un estado.
- ▶ El primer conjunto (conjunto o estado 0) será la primera regla con el punto al comienzo:

0) $A \rightarrow .id := E$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

- ▶ Se arman nuevos conjuntos, explorando todos los posibles caminos desde el punto, :

0) $A \rightarrow .id := E$

1) 0 y id

$A \rightarrow id . := E$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

1) 0 y id

$A \rightarrow id . := E$

2) 1 y :=

$A \rightarrow id := . E$

- ***Cuando el punto está delante de un no terminal, se agregan todas las definiciones de ese no terminal.***

$E \rightarrow . E + T$

$E \rightarrow . T$

$T \rightarrow . T * F$

$T \rightarrow . F$

$F \rightarrow . id$

$F \rightarrow . cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

3) 2 y E

$A \rightarrow id := E .$

$E \rightarrow E . + T$

4) 2 y T

$E \rightarrow T .$

$T \rightarrow T . * F$

5) 2 y F

$T \rightarrow F .$

6) 2 y id

$F \rightarrow id .$

7) 2 y cte

$F \rightarrow cte .$

2) 1 y :=

$A \rightarrow id := . E$

$E \rightarrow . E + T$

$E \rightarrow . T$

$T \rightarrow . T * F$

$T \rightarrow . F$

$F \rightarrow . id$

$F \rightarrow . cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

8) 3 y +

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

9) 4 y *

$T \rightarrow T * \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

3) 2 y E

$A \rightarrow id := E \cdot$

$E \rightarrow E \cdot + T$

4) 2 y T

$E \rightarrow T \cdot$

$T \rightarrow T \cdot * F$

5) 2 y F

$T \rightarrow F \cdot$

6) 2 y id

$F \rightarrow id \cdot$

7) 2 y cte

$F \rightarrow cte \cdot$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

I0) 8 y T

$E \rightarrow E + T \cdot$

$T \rightarrow T \cdot * F$

II) 8 y F

$T \rightarrow F \cdot$

8) 3 y +

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

3) 2 y E

$A \rightarrow id := E .$

$E \rightarrow E . + T$

4) 2 y T

$E \rightarrow T .$

$T \rightarrow T . * F$

5) 2 y F / 8 y F

$T \rightarrow F .$

6) 2 y id

$F \rightarrow id .$

7) 2 y cte

$F \rightarrow cte .$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

10) 8 y T

$E \rightarrow E + T \cdot$

$T \rightarrow T \cdot * F$

~~11) 8 y F~~

~~$T \rightarrow F \cdot$~~

11) 8 y id

$F \rightarrow id \cdot$

8) 3 y +

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

3) 2 y E

$A \rightarrow id := E .$

$E \rightarrow E . + T$

4) 2 y T

$E \rightarrow T .$

$T \rightarrow T . * F$

5) 2 y F / 8 y F

$T \rightarrow F .$

6) 2 y id / 8 y id

$F \rightarrow id .$

7) 2 y cte

$F \rightarrow cte .$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

10) 8 y T

$E \rightarrow E + T \cdot$

$T \rightarrow T \cdot * F$

~~11) 8 y F~~

~~$T \rightarrow F \cdot$~~

~~11) 8 y id~~

~~$F \rightarrow id \cdot$~~

11) 8 y cte

$F \rightarrow cte \cdot$

8) 3 y +

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

3) 2 y E

$A \rightarrow id := E .$

$E \rightarrow E . + T$

4) 2 y T

$E \rightarrow T .$

$T \rightarrow T . * F$

5) 2 y F / 8 y F

$T \rightarrow F .$

6) 2 y id / 8 y id

$F \rightarrow id .$

7) 2 y cte / 8 y cte

$F \rightarrow cte .$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

10) 8 y T

$E \rightarrow E + T \cdot$

$T \rightarrow T \cdot * F$

~~11) 8 y F~~

~~$T \rightarrow F \cdot$~~

~~11) 8 y id~~

~~$F \rightarrow id \cdot$~~

~~11) 8 y cte~~

~~$F \rightarrow cte \cdot$~~

8) 3 y +

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

9) 4 y *

$T \rightarrow T * \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

11) 9 y F

$T \rightarrow T * F .$

12) 9 y id

$F \rightarrow id .$

9) 4 y *

$T \rightarrow T * . F$

$F \rightarrow . id$

$F \rightarrow . cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

3) 2 y E

$A \rightarrow id := E .$

$E \rightarrow E . + T$

4) 2 y T

$E \rightarrow T .$

$T \rightarrow T . * F$

5) 2 y F / 8 y F

$T \rightarrow F .$

6) 2 y id / 8 y id / 9 y id

$F \rightarrow id .$

7) 2 y cte / 8 y cte

$F \rightarrow cte .$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

11) 9 y F

$T \rightarrow T * F .$

~~12) 9 y id~~

~~$F \rightarrow id .$~~

12) 9 y cte

$F \rightarrow cte .$

9) 4 y *

$T \rightarrow T * . F$

$F \rightarrow . id$

$F \rightarrow . cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

3) 2 y E

$A \rightarrow id := E .$

$E \rightarrow E . + T$

4) 2 y T

$E \rightarrow T .$

$T \rightarrow T . * F$

5) 2 y F / 8 y F

$T \rightarrow F .$

6) 2 y id / 8 y id / 9 y id

$F \rightarrow id .$

7) 2 y cte / 8 y cte / 9 y cte

$F \rightarrow cte .$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

11) 9 y F

$T \rightarrow T * F \cdot$

~~12) 9 y id~~

~~$F \rightarrow id \cdot$~~

~~12) 9 y cte~~

~~$F \rightarrow cte \cdot$~~

9) 4 y *

$T \rightarrow T * \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

10) 8 y T

$E \rightarrow E + T \cdot$

$T \rightarrow T \cdot * F$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

12) 10 y *

$T \rightarrow T * . F$

$F \rightarrow . id$

$F \rightarrow . cte$

10) 8 y T

$E \rightarrow E + T .$

$T \rightarrow T . * F$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

8) 3 y +

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

9) 4 y * / 10 y *

$T \rightarrow T * \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

~~12) 10 y *~~

$T \rightarrow T * \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

10) 8 y T

$E \rightarrow E + T \cdot$

$T \rightarrow T \cdot * F$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

0) $A \rightarrow \cdot id := E$

1) 0 y id

$A \rightarrow id \cdot := E$

2) 1 y :=

$A \rightarrow id := \cdot E$

$E \rightarrow \cdot E + T$

$E \rightarrow \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

3) 2 y E

$A \rightarrow id := E \cdot$

$E \rightarrow E \cdot + T$

4) 2 y T

$E \rightarrow T \cdot$

$T \rightarrow T \cdot * F$

5) 2 y F / 8 y F

$T \rightarrow F \cdot$

6) 2 y id / 8 y id / 9 y id

$F \rightarrow id \cdot$

7) 2 y cte / 8 y cte / 9 y cte

$F \rightarrow cte \cdot$

8) 3 y +

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

9) 4 y * / 10 y *

$T \rightarrow T * \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

10) 8 y T

$E \rightarrow E + T \cdot$

$T \rightarrow T \cdot * F$

11) 9 y F

$T \rightarrow T * F \cdot$

MATRIZ 1

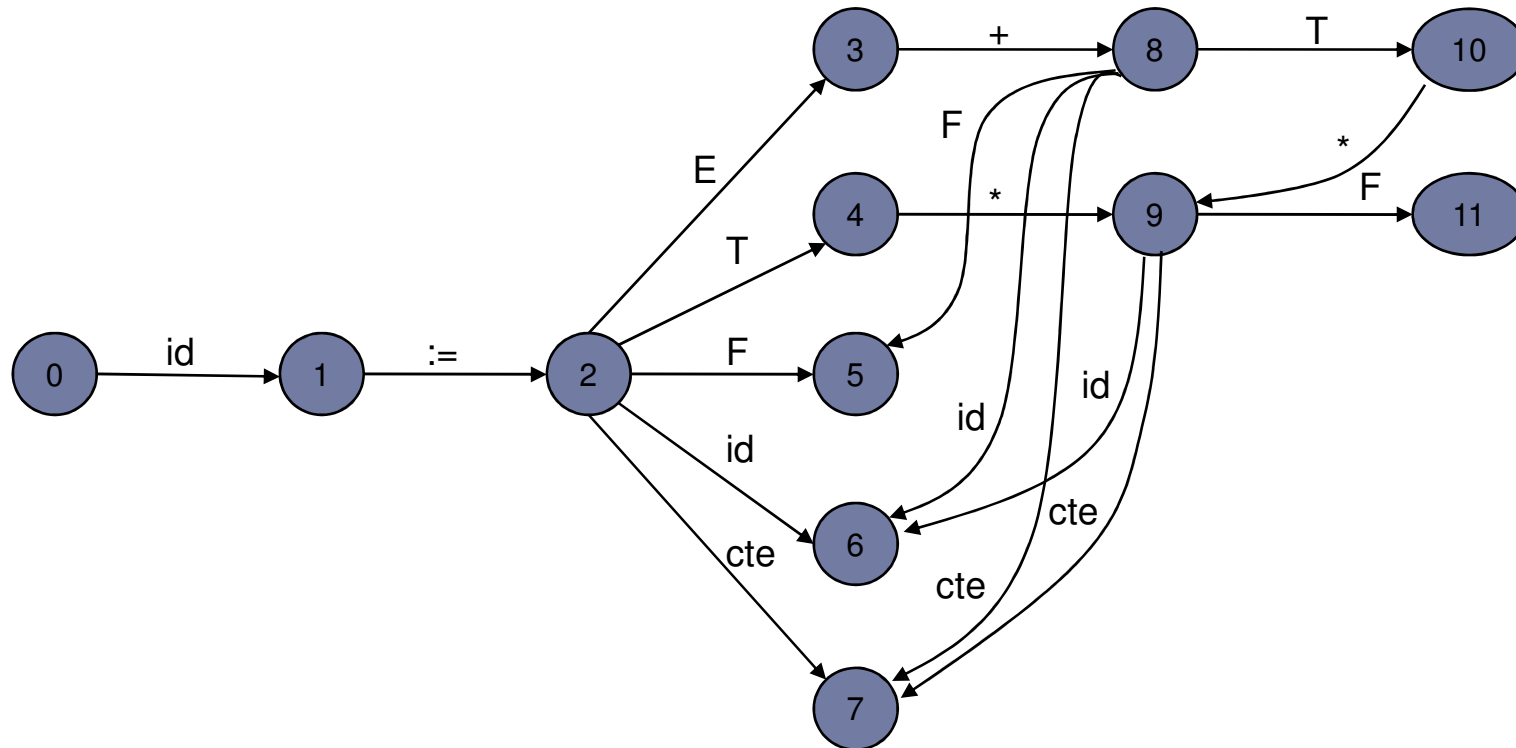
MATRIZ 2



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 1

- ▶ Los conjuntos pueden representarse mediante un diagrama de transición de estados



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 2

- ▶ Es posible construir la matriz de transición de estados equivalente.
- ▶ Se construye la matriz de transición de estados:
 - ▶ Para cada estado y Terminal, se coloca la letra **d** seguida del número del conjunto.
 - ▶ Para cada estado y No Terminal, se coloca el número del conjunto.



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 2

	id	cte	:=	+	*	\$	A	E	T	F
0	<i>d1</i>									
1			<i>d2</i>							
2	<i>d6</i>	<i>d7</i>								
3				<i>d8</i>						
4					<i>d9</i>					
5										
6										
7										
8	<i>d6</i>	<i>d7</i>								
9	<i>d6</i>	<i>d7</i>								
10					<i>d9</i>					
11										

Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 2

- ▶ **Se construye la matriz de transición de estados:**
 - ▶ Para cada estado y No Terminal, se coloca el número del conjunto.



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 2

	id	cte	:=	+	*	\$	A	E	T	F
0	d1									
1			d2							
2	d6	d7						3	4	5
3				d8						
4					d9					
5										
6										
7										
8	d6	d7							10	5
9	d6	d7								11
10					d9					
11										

Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 3

- ▶ Se identifican los estados donde el grado de cumplimiento está al final de las reglas (punto al final).



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 3

0) $A \rightarrow \cdot id := E$

1) 0 y id

$A \rightarrow id \cdot := E$

2) 1 y :=

$A \rightarrow id := \cdot E$

$E \rightarrow \cdot E + T$

$E \rightarrow \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

3) 2 y E

$A \rightarrow id := E \cdot$ Regla 1

$E \rightarrow E \cdot + T$

4) 2 y T

$E \rightarrow T \cdot$ Regla 3

$T \rightarrow T \cdot * F$

5) 2 y F / 8 y F

$T \rightarrow F \cdot$ Regla 5

6) 2 y id / 8 y id / 9 y id

$F \rightarrow id \cdot$ Regla 6

7) 2 y cte / 8 y cte / 9 y cte

$F \rightarrow cte \cdot$ Regla 7

8) 3 y +

$E \rightarrow E + \cdot T$

$T \rightarrow \cdot T * F$

$T \rightarrow \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

9) 4 y * / 10 y *

$T \rightarrow T * \cdot F$

$F \rightarrow \cdot id$

$F \rightarrow \cdot cte$

10) 8 y T

$E \rightarrow E + T \cdot$ Regla 2

$T \rightarrow T \cdot * F$

11) 9 y F

$T \rightarrow T * F \cdot$ Regla 4

MATRIZ

Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 3

- ▶ Se agrega a cada estado la regla con el NT que se genera, a la derecha de la matriz.



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 3

	id	cte	:=	+	*	\$	A	E	T	F	
0	d1										
1			d2								
2	d6	d7						3	4	5	
3				d8							r1 → A
4					d9						r3 → E
5											r5 → T
6											r6 → F
7											r7 → F
8	d6	d7							10	5	
9	d6	d7								11	
10					d9						r2 → E
11											r4 → T

Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 4

- ▶ Se obtienen los SIGUIENTES de los NT



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 4

- ▶ Para obtener los SIGUIENTES, se necesitan los PRIMEROS

PRIMEROS

1. Si x es Terminal \rightarrow $\text{PRIMEROS}(x) = \{x\}$
2. Si $X \rightarrow \lambda$ \rightarrow $\lambda \in \text{PRIMEROS}(X)$
3. Si $X \rightarrow A B C D \dots$ \rightarrow $\text{PRIMEROS}(X) =$
 $\text{PRIMEROS}(A)$
 $\cup \text{PRIMEROS}(B)$
 $\cup \text{PRIMEROS}(C)$
 $\cup \dots$
 Si $\lambda \in \text{PRIMEROS}(A)$
 Si $\lambda \in \text{PRIMEROS}(A) \wedge \lambda \in \text{PRIMEROS}(B)$

Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 4

PRIMEROS

1. Si x es Terminal \rightarrow $\text{PRIMEROS}(x) = \{x\}$
2. Si $X \rightarrow \lambda$ \rightarrow $\lambda \in \text{PRIMEROS}(X)$
3. Si $X \rightarrow A B C D \dots$ \rightarrow

$$\text{PRIMEROS}(X) = \text{PRIMEROS}(A) \cup \text{PRIMEROS}(B) \cup \text{PRIMEROS}(C) \cup \dots$$

$$\begin{aligned} \text{Si } \lambda \in \text{PRIMEROS}(A) \quad & \text{Si } \lambda \in \text{PRIMEROS}(A) \\ & \wedge \lambda \in \text{PRIMEROS}(B) \end{aligned}$$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 4

- ▶ Se obtienen los SIGUIENTES de los NT

SIGUIENTES

1. Se agrega \$ a SIGUIENTES(A)
(Siendo A la primera regla de la gramática)

2. Si $X \rightarrow \alpha A \beta$ \rightarrow
SIGUIENTES(A) =

$$\text{PRIMEROS}(\beta) \cup \text{SIGUIENTES}(X)$$

Excepto λ

Si $\beta \neq \epsilon$

$\forall \lambda \in \text{PRIMEROS}(\beta)$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 4

► Se obtienen los siguientes de los NT

1) $A \rightarrow id := E$

$$\text{PRIMEROS}(F) = \{ \text{cte}, \text{id} \}$$

2) $E \rightarrow E + T$

$$\text{PRIMEROS}(T) = \text{PRIMEROS}(F) = \{ \text{cte}, \text{id} \}$$

3) $E \rightarrow T$

$$\text{PRIMEROS}(E) = \text{PRIMEROS}(T) = \{ \text{cte}, \text{id} \}$$

4) $T \rightarrow T * F$

$$\text{PRIMEROS}(A) = \{ \text{id} \}$$

5) $T \rightarrow F$

$$\text{SIGUIENTES}(A) = \{ \$ \}$$

6) $F \rightarrow id$

$$\text{SIGUIENTES}(E) = \{ \$, + \}$$

7) $F \rightarrow \text{cte}$

$$\text{SIGUIENTES}(T) = \{ \$, +, * \}$$

$$\text{SIGUIENTES}(F) = \{ \$, +, * \}$$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 4

- ▶ Para todos los estados donde se genera un NT, colocar la letra r seguida del número de regla para todos los siguientes de ese NT

$$\text{SIGUIENTES}(A) = \{ \$ \}$$

$$\text{SIGUIENTES}(E) = \{ \$, + \}$$

$$\text{SIGUIENTES}(T) = \{ \$, + , * \}$$

$$\text{SIGUIENTES}(F) = \{ \$, + , * \}$$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 4

	id	cte	:=	+	*	\$	A	E	T	F	
0	d1										
1			d2								
2	d6	d7						3	4	5	
3				d8		<i>r1</i>					$r1 \rightarrow A$
4				<i>r3</i>	d9	<i>r3</i>					$r3 \rightarrow E$
5				<i>r5</i>	<i>r5</i>	<i>r5</i>					$r5 \rightarrow T$
6				<i>r6</i>	<i>r6</i>	<i>r6</i>					$r6 \rightarrow F$
7				<i>r7</i>	<i>r7</i>	<i>r7</i>					$r7 \rightarrow F$
8	d6	d7							10	5	
9	d6	d7								11	
10				<i>r2</i>	d9	<i>r2</i>					$r2 \rightarrow E$
11				<i>r4</i>	<i>r4</i>	<i>r4</i>					$r4 \rightarrow T$



Parsing Ascendente Predictivo LR(1)

Algoritmo: Paso 5

	id	cte	:=	+	*	\$	A	E	T	F	
0	d1						FIN				
1			d2								
2	d6	d7						3	4	5	
3				d8		r1					r1 → A
4				r3	d9	r3					r3 → E
5				r5	r5	r5					r5 → T
6				r6	r6	r6					r6 → F
7				r7	r7	r7					r7 → F
8	d6	d7							10	5	
9	d6	d7								11	
10				r2	d9	r2					r2 → E
11				r4	r4	r4					r4 → T

Se agrega FIN en la celda: estado 0 - NT hipótesis



Parsing Ascendente Predictivo LR(1)

Tabla de Parsing

	id	cte	:=	+	*	\$	A	E	T	F
0	d1						FIN			
1			d2							
2	d6	d7						3	4	5
3				d8		r1				
4				r3	d9	r3				
5				r5	r5	r5				
6				r6	r6	r6				
7				r7	r7	r7				
8	d6	d7							10	5
9	d6	d7								11
10				r2	d9	r2				
11				r4	r4	r4				

r1 → A

r3 → E

r5 → T

r6 → F

r7 → F

r2 → E

r4 → T



Parsing Ascendente Predictivo LR(1)

Programación

- ▶ Se construye un programa a partir de la tabla.
- ▶ Se utilizará una pila.



Parsing Ascendente Predictivo LR(1)

Programación

Acciones:

- ▶ **dn** Desplazamiento (shift) al estado **n**
 - Apilar el estado actual y el token leído
 - Leer token
 - Ir al estado n
- ▶ **rm** Reducción (reduce) por la regla **m**
 - Reemplazar el lado derecho de la regla m por el lado izquierdo
 - Desapilar hasta llegar al estado en que estaba al entrar a la regla reducida (el nuevo estado lo da Estado/NT del tope de la pila)
- ▶ **j** Ir (go to) al estado **j** (luego de la reducción)
 - Ir al estado j
- ▶ **Aceptación** Se llegó a la hipótesis
 - Terminar
- ▶ **Error** Celda vacía
 - Informar y terminar



Parsing Ascendente Predictivo LR(1)

Conflictos

- ▶ Si la gramática es ambigua, se producen conflictos en el parsing
 - ▶ Si en una celda aparece dn y rm
 - ▶ Conflicto **Shift / Reduce**
 - ▶ Si en una celda aparece rm y rj
 - ▶ Conflicto **Reduce / Reduce**
- ➔ La gramática no se puede compilar con esta técnica
- ▶ En general, se puede re-escribir la gramática para resolver los conflictos.

Parsing Ascendente Predictivo LR(1)

Ejemplo

$z := a * b * 5 + c * 3 \quad \rightarrow \quad id := id * id * cte + id * cte$

Reglas	Estado	Leído	Pila
r7	7	\$	0,id 1, := 2,E 3,+ 8,T 10,* <u>9,cte</u>
r4	11	\$	0,id 1, := 2,E 3,+ <u>8,T 10,* 9,F</u>
r2	10	\$	0,id 1, := <u>2,E 3,+ 8,T</u>
r1	3	\$	<u>0,id 1, := 2,E</u>
FIN			0,A

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99



Parsing Ascendente Predictivo LR(1)

Errores

- ▶ Los errores son determinados por las celdas vacías de la matriz.
- ▶ Se puede escribir el mensaje que corresponde a cada celda vacía.



¿Todas las celdas vacías
representan errores?



Parsing Ascendente Predictivo LR(1)

Ejemplo con error

$z := a * + c * 3$



$id := id * + id * cte$

Reglas	Estado	Leído	Pila
	0	id	-
	1	:=	0,id
	2	ld	0,id l,:=
r6	6	*	0,id l,:= <u>2,id</u>
r5	5	*	0,id l,:= <u>2,F</u>
	4	*	0,id l,:= 2,T
	9	+	0,id l,:= 2,T 4,*
ERROR			

id	:=	ld	*	+	cte
0	1	2	3	4	5
6	7	8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35
36	37	38	39	40	41
42	43	44	45	46	47
48	49	50	51	52	53
54	55	56	57	58	59
60	61	62	63	64	65
66	67	68	69	70	71
72	73	74	75	76	77
78	79	80	81	82	83
84	85	86	87	88	89
90	91	92	93	94	95
96	97	98	99	100	101



Parsing Ascendente Predictivo LR(1)

Errores

- ▶ Ante un error, el Análisis Sintáctico se detiene.
- ▶ Para continuar con la compilación, se deben usar técnicas de recuperación:
 - Producciones de error
 - Modo pánico
 - Recuperación a nivel de frase
 - Corrección global



Errores Sintácticos

Técnicas de Recuperación

- ▶ **Producciones de error:**
 - Aumentar la gramática con reglas que generen las construcciones erróneas.
- ▶ **Modo pánico:**
 - Consumir hasta un carácter de sincronización.
- ▶ **Recuperación a nivel de frase:**
 - Utilizar reemplazo / inserción / borrado.
- ▶ **Corrección global:**
 - Muy costosa (sólo teórica)



Ejercicios

- Dada la siguiente gramática y su tabla de Parsing Ascendente Predictivo:

1. $A \rightarrow id := E$
2. $E \rightarrow E + T$
3. $E \rightarrow T$
4. $T \rightarrow T * F$
5. $T \rightarrow F$
6. $F \rightarrow id$
7. $F \rightarrow cte$

	id	cte	:=	+	*	\$	A	E	T	F
0	d1						FIN			
1			d2							
2	d6	d7						3	4	5
3				d8		r1				
4				r3	d9	r3				
5				r5	r5	r5				
6	A			r6	r6	r6				
7		B		r7	r7	r7				
8	d6	d7							10	5
9	d6	d7								11
10				r2	d9	r2				
11				r4	r4	r4				

Se han marcado 2 casilleros, con las letras A, y B, en los que se produce un error.
Se desea crear una gramática de error que detecte esos errores

- ¿Cuáles son esos errores?
- ¿Cuáles serían las reglas de la gramática de error?

Ejercicios

- ▶ Dada la siguiente gramática y su tabla de Parsing Ascendente Predictivo:

- 0) $E' \rightarrow E$
- 1) $E \rightarrow E + T$
- 2) $E \rightarrow T$
- 3) $T \rightarrow T * F$
- 4) $T \rightarrow F$
- 5) $F \rightarrow id$
- 6) $F \rightarrow cte$

	id	cte	+	*	\$		E'	E	T	F
0	d4	d5					FIN	1	2	3
1			d6	D	r0					
2			r2	d7	r2					
3	C		r4	r4	r4					
4	A		r5	r5	r5					
5			r6	r6	r6					
6	d4	d5							8	3
7	d4	d5		B						9
8			r1	d7	r1					
9			r3	r3	r3					

- ▶ En dicha tabla se han marcado con las letras A, B, C y D algunas casillas vacías.
- ▶ Indicar qué situación representa cada celda marcada, con un ejemplo para cada una de ellas.