



UCA

Universidad
de Cádiz

TEORÍA DE AUTÓMATAS Y LENGUAJES FORMALES

ALEJANDRO SERRANO FERNÁNDEZ

ANÁLISIS SINTÁCTICO. PRÁCTICA EVALUABLE

Dada la gramática:

$$S \rightarrow S + a \mid a * B \mid a$$

$$A \rightarrow Ab \mid cd$$

$$B \rightarrow Ac \mid Cde$$

$$C \rightarrow Sd \mid \epsilon$$

1. Eliminar recursividades por la izquierda, aplicar sustituciones y reemplazos

$$S \rightarrow a * B S' \mid a S'$$

$$S' \rightarrow '+' a S' \mid \epsilon$$

$$A \rightarrow cdA'$$

$$A' \rightarrow bA'$$

$$B \rightarrow Ac \mid Cd$$

$$C \rightarrow Sd \mid \epsilon$$

$$A \rightarrow A\alpha \mid B$$



$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A'$$

Reemplazamos C

$$S \rightarrow a * B S' \mid a S'$$

$$S' \rightarrow + a S' \mid \epsilon$$

$$A \rightarrow cdA'$$

$$A' \rightarrow bA' \mid \epsilon$$

$$B \rightarrow Ac \mid Sd d e \mid \epsilon d e \rightarrow Ac \mid Sd d e \mid d e$$

Factorización

Podemos aplicar factorización sobre S

$$S \rightarrow \underline{a} * BS' \mid \underline{a} S'$$

$\left\{ \begin{array}{l} S \rightarrow aD \\ D \rightarrow *BS' \mid S' \end{array} \right.$

$$A \rightarrow \alpha \beta \mid \alpha \gamma$$



$$A \rightarrow \alpha B$$

$$B \rightarrow \beta \mid \gamma$$

Resultado:

$$S \rightarrow aD$$

$$S' \rightarrow +aS' \mid \varepsilon$$

$$D \rightarrow *BS' \mid S'$$

$$A \rightarrow cdA'$$

$$A' \rightarrow bA' \mid \varepsilon$$

$$B \rightarrow Ac \mid Sdde \mid de$$

2. Eliminar símbolos inútiles

Eliminaremos aquellos símbolos que nos lleven a estados terminales.

Reglas $A \rightarrow w$ (nos llevan a estados finales)

$$N' = \emptyset$$

$$S' \rightarrow \varepsilon \Rightarrow N' = \{S'\}$$

$$A' \rightarrow \varepsilon \Rightarrow N' = \{S', A'\}$$

$$B \rightarrow de \Rightarrow N' = \{S', A', B\}$$

Reglas $A \rightarrow \alpha$ (contienen símbolos del conjunto N')

1ª Iteración

$$D \rightarrow * B \underline{S'} \Rightarrow N' = \{ \underline{S'}, A', B, D \}$$

$$A \rightarrow cd \underline{A'} \Rightarrow N' = \{ S', \underline{A'}, B, D, A \}$$

2ª Iteración

$$S \rightarrow a \underline{D} \Rightarrow N' = \{ S', A', B, \underline{D}, A \}$$

Todas las variables están contenidas en N' , luego no eliminamos ninguna regla.

3. Eliminar símbolos inalcanzables

$$J = \{ S \} ; N' = \{ S \} ; T = \emptyset$$

$$S \rightarrow aD \Rightarrow N' = \{ S, D \} ; T = \{ a, \epsilon \} ; J = \{ D \}$$

$$D \rightarrow * B \underline{S'} \Rightarrow N' = \{ S, D, B, \underline{S'} \} ; T = \{ a, \epsilon, * \}$$
$$| S' \quad J = \{ B, S' \}$$

$$B \rightarrow AC \Rightarrow N' = \{ S, D, B, S', A \} ; T = \{ a, \epsilon, *, c, d, e \} ;$$
$$| Sdde \quad J = \{ S', A \}$$
$$| de$$

$$S' \rightarrow + a S' | \epsilon \Rightarrow N' = \{ S, D, B, S', A \} ; T = \{ a, \epsilon, *, c, d, e, + \}$$
$$J = \{ A \}$$

$$S \rightarrow aD$$
$$S' \rightarrow + a S' | \epsilon$$
$$D \rightarrow * B S' | S'$$
$$A \rightarrow cd A'$$
$$A' \rightarrow b A' | \epsilon$$
$$B \rightarrow Ac | Sdde | de$$

$$A \rightarrow cdA' \Rightarrow N' = \{ S, D, B, S', A \} ; T = \{ a, \epsilon, *, c, d, e, + \};$$

$$J = \{ A' \}$$

$$A' \rightarrow bA' \mid \epsilon \Rightarrow N' = \{ S, D, B, S', A \} ; T = \{ a, \epsilon, *, c, d, e, + \};$$

$$J = \{ \emptyset \}$$

Todos los símbolos, tanto terminales como no terminales, están contenidos en N' y T , luego no podemos eliminar ninguno.

4. Generar la tabla $LL(1)$ y decir si la gramática generada es $LL(1)$ o no

$$S \rightarrow aD$$

$$S' \rightarrow +aS' \mid \epsilon$$

$$D \rightarrow *BS' \mid S'$$

$$A \rightarrow cdA'$$

$$A' \rightarrow bA' \mid \epsilon$$

$$B \rightarrow Ac \mid sdde \mid de$$

$$\text{First}(B) = \{ d \} \cup \text{First}(A) \cup \text{First}(S)$$

$$= \{ d, c, a \}$$

$$\text{First}(A') = \{ b, \epsilon \}$$

$$\text{First}(A) = \{ c \}$$

$$\text{First}(D) = \{ * \} \cup \text{First}(S') = \{ *, +, \epsilon \}$$

$$\text{First}(S') = \{ +, \epsilon \} \cup \text{First}(S') = \{ +, \epsilon \}$$

$$\text{First}(S) = \{ a \}$$

$$\text{Follow}(B) = \text{First}(S') - \{ \epsilon \} \cup \text{Follow}(D) = \{ +, \$, d \}$$

$$\text{Follow}(A') = \text{Follow}(A') \cup \text{Follow}(A) = \{ c \}$$

$$\text{Follow}(A) = \{ c \}$$

$$\text{Follow}(D) = \text{Follow}(S) = \{ \$, d \}$$

$$\text{Follow}(S') = \text{Follow}(D) \cup \text{Follow}(S') = \{ \$, d \}$$

$$\text{Follow}(S) = \{ \$, d \}$$

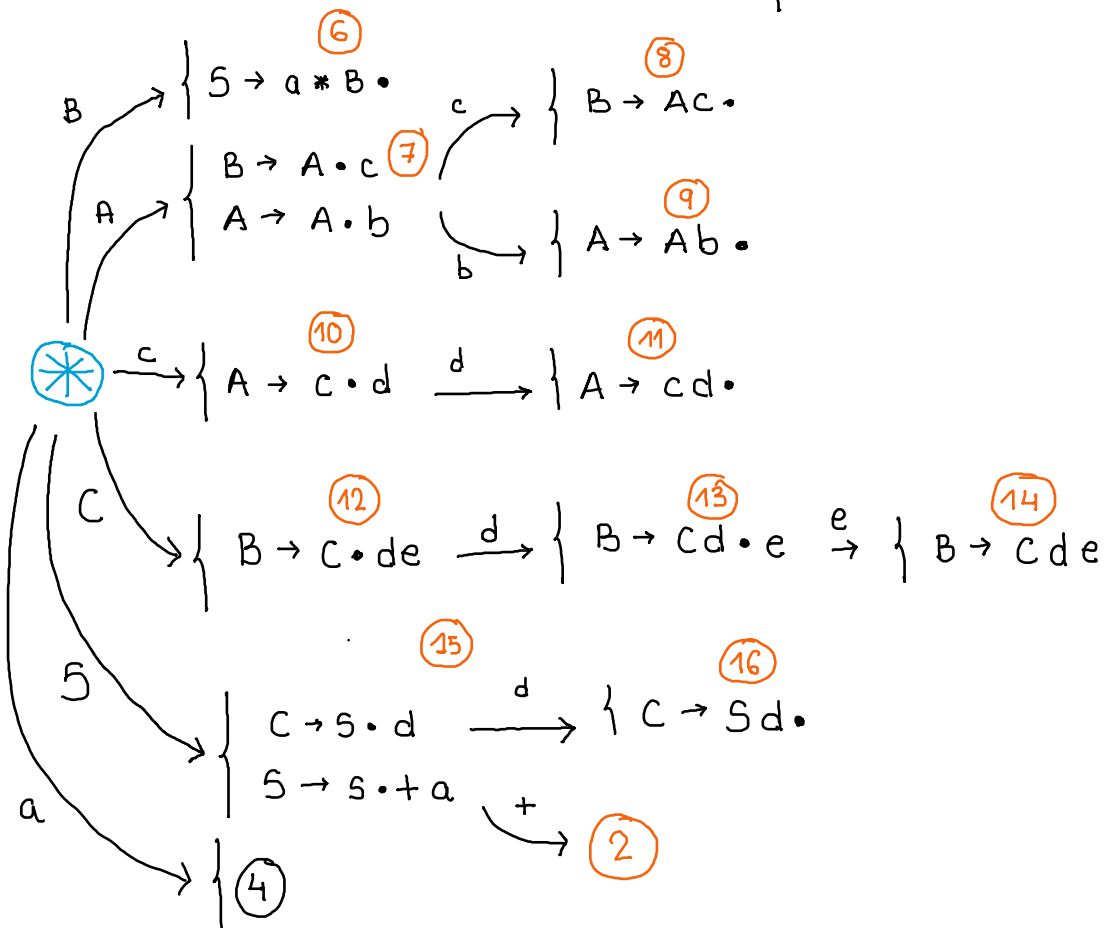
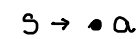
- $A \rightarrow \alpha \implies A \rightarrow \alpha$ in $\text{table}[A, \text{First}(\alpha) - \{\epsilon\}]$
- $A \rightarrow \epsilon \implies A \rightarrow \epsilon$ in $\text{table}[A, \text{Follow}(A)]$
- $A \rightarrow \alpha \ \& \ \alpha \Rightarrow^* \epsilon \implies A \rightarrow \alpha$ in $\text{table}[A, \text{Follow}(A)]$

	a	b	c	d	e	+	*	\$
S	S -> aD							
S'				S' -> E		S' -> +aS'		S' -> E
D				D -> S'		D -> S'	D -> *BS'	D -> S'
A			A -> cdA'					
A'		A' -> bA'	A -> E					
B	B -> Sdde		B -> Ac	S -> de				

Como podemos observar no hay conflictos, es decir, no hay más de una regla en una misma celda, por tanto la gramática es LL(1).

5. Generar la tabla SLR(1) para G y comprobar si es SLR(1) o no.

Primero construiremos el DFA, tal y como se muestra en la siguiente página:

$$C \rightarrow \mathcal{E}$$


	a	b	c	d	e	+	*	\$	S	A	B	C
0	Shift 4								GOTO 1			
1						Shift 2		ACCEPT				
2	Shift 3											
3				R(S->S+a)		R(S->S+a)		R(S->S+a)				
4				R(S->a)		R(S->a)	Shift 5	R(S->a)				
5	Shift 4		Shift 10						GOTO 15	GOTO 7	GOTO 6	GOTO 12
6				R(S->a*B)		R(S->a*B)		R(S->a*B)				
7		Shift 9	Shift 8									
8				R(B->Ac)		R(B->Ac)		R(B->Ac)				
9		R(A->Ab)	R(A->Ab)									
10				Shift 11								
11		R(A->cd)	R(A->cd)									
12				Shift 13								
13					Shift 14							
14				R(B->Cde)		R(B->Cde)		R(B->Cde)				
15				Shift 16		Shift 2						
16				R(C->Sd)								

Follow(S') = {\$}

Follow(S) = Follow(S') \cup {+,d} = {\$,+,d}

Follow(A) = {b,c}

Follow(B) = Follow(S) = {\$,+,d}

Follow(C) = {d}

Como podemos observar, no existe ningún conflicto, por lo tanto la gramática dada es SLR(1).