

**SOLUCIONES****Parte I: PREGUNTAS TIPO TEST. 30%.**

- | | | | | |
|-------|-------|-------|--------|--------|
| 1. b) | 4. b) | 7. a) | 10. c) | 13. a) |
| 2. b) | 5. c) | 8. a) | 11. b) | 14. b) |
| 3. c) | 6. b) | 9. c) | 12. c) | 15. a) |

Parte II: PROBLEMA. 70%.**Apartado 1.**

Obtenemos los conjuntos PRIMERO y SIGUIENTE que solicita el enunciado:

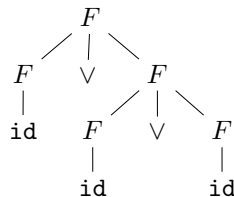
$$\begin{aligned}\text{PRIMERO}(F) &= \{ \vee, \text{id} \} \\ \text{PRIMERO}(L) &= \{ \text{id} \}\end{aligned}$$

$$\begin{aligned}\text{SIGUIENTE}(F) &= \{ \$, \vee,) \} \\ \text{SIGUIENTE}(L) &= \{) \}\end{aligned}$$

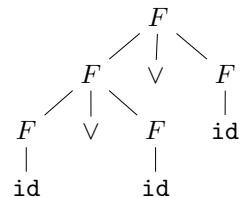
La gramática del enunciado no puede ser una gramática LL(1) por tres razones:

- Es ambigua por la regla $F \rightarrow F \vee F$, ya que esta regla no fija la asociatividad del operador \vee . Podemos verlo obteniendo dos derivaciones más a la derecha (o dos árboles) de la entrada $\text{id} \vee \text{id} \vee \text{id}$:

$$\begin{aligned}F &\Rightarrow F \vee F \\ &\Rightarrow F \vee F \vee F \\ &\Rightarrow F \vee F \vee \text{id} \\ &\Rightarrow F \vee \text{id} \vee \text{id} \\ &\Rightarrow \text{id} \vee \text{id} \vee \text{id}\end{aligned}$$



$$\begin{aligned}F &\Rightarrow F \vee F \\ &\Rightarrow F \vee \text{id} \\ &\Rightarrow F \vee F \vee \text{id} \\ &\Rightarrow F \vee \text{id} \vee \text{id} \\ &\Rightarrow \text{id} \vee \text{id} \vee \text{id}\end{aligned}$$



- Es recursiva por la izquierda con el no terminal F y la regla $F \rightarrow F \vee F$.
- No está factorizada, ya que las reglas $L \rightarrow \text{id} \mid \text{id} , L$ tienen el factor común id .

Apartado 2.

Comenzamos eliminando la ambigüedad de la gramática. Como el enunciado indica que el operador \vee es asociativo por la izquierda, creamos un nuevo no terminal T para representar al operando derecho del operador \vee , o bien a una fórmula sin el \vee como operador de primer nivel:

$$\begin{aligned}F &\rightarrow F \vee T \mid T \\ T &\rightarrow \vee \text{id} (F) \mid \text{id} (L) \\ L &\rightarrow \text{id} \mid \text{id} , L\end{aligned}$$

El siguiente paso consiste en la eliminación de la recursión por la izquierda inmediata del no terminal F :

$$\begin{aligned}F &\rightarrow T \mid T F' \\ F' &\rightarrow \vee T \mid \vee T F' \\ T &\rightarrow \vee \text{id} (F) \mid \text{id} (L) \\ L &\rightarrow \text{id} \mid \text{id} , L\end{aligned}$$

Ahora se pueden eliminar los factores comunes de F , F' y L :

$$\begin{aligned} F &\rightarrow T F'' \\ F' &\rightarrow \vee T F'' \\ F'' &\rightarrow \lambda \mid F' \\ T &\rightarrow \forall \text{id} (F) \mid \text{id} (L) \\ L &\rightarrow \text{id} L' \\ L' &\rightarrow \lambda \mid , L \end{aligned}$$

Aunque no es un paso necesario, se puede simplificar un poco la gramática para obtener G' :

$$\begin{aligned} F &\rightarrow T F' \\ F' &\rightarrow \lambda \mid \vee T F' \\ T &\rightarrow \forall \text{id} (F) \mid \text{id} (L) \\ L &\rightarrow \text{id} L' \\ L' &\rightarrow \lambda \mid , L \end{aligned}$$

A partir de la última gramática, calculamos los conjuntos PRIMERO y SIGUIENTE:

$$\begin{aligned} \text{PRIMERO}(F) &= \{ \forall, \text{id} \} & \text{SIGUIENTE}(F) &= \{ \$,) \} \\ \text{PRIMERO}(F') &= \{ \lambda, \vee \} & \text{SIGUIENTE}(F') &= \{ \$,) \} \\ \text{PRIMERO}(T) &= \{ \forall, \text{id} \} & \text{SIGUIENTE}(T) &= \{ \vee, \$,) \} \\ \text{PRIMERO}(L) &= \{ \text{id} \} & \text{SIGUIENTE}(L) &= \{) \} \\ \text{PRIMERO}(L') &= \{ \lambda, , \} & \text{SIGUIENTE}(L') &= \{) \} \end{aligned}$$

Con los conjuntos anteriores podemos calcular los PREDICT de las reglas de la gramática G' :

$$\begin{aligned} \text{PREDICT}(F \rightarrow T F') &= \{ \forall, \text{id} \} & \text{PREDICT}(T \rightarrow \text{id} (L)) &= \{ \text{id} \} \\ \text{PREDICT}(F' \rightarrow \lambda) &= \{ \$,) \} & \text{PREDICT}(L \rightarrow \text{id} L') &= \{ \text{id} \} \\ \text{PREDICT}(F' \rightarrow \vee T F') &= \{ \vee \} & \text{PREDICT}(L' \rightarrow \lambda) &= \{) \} \\ \text{PREDICT}(T \rightarrow \forall \text{id} (F)) &= \{ \forall \} & \text{PREDICT}(L' \rightarrow , L) &= \{ , \} \end{aligned}$$

Se puede observar que la intersección de los conjuntos PREDICT de cualquier par de reglas de un mismo no terminal es siempre vacía. Por tanto, la gramática equivalente G' es LL(1).

Apartado 3.

Los conjuntos de ítems de la colección LR(0) son los siguientes:

$$\begin{aligned} I_0 &= \{ F' \rightarrow \cdot F \\ &\quad F \rightarrow \cdot F \vee F \\ &\quad F \rightarrow \cdot \forall \text{id} (F) \\ &\quad F \rightarrow \cdot \text{id} (L) \} \\ I_1 &= \text{GOTO}(I_0, F) = \{ F' \rightarrow F \cdot \\ &\quad F \rightarrow F \cdot \vee F \} \\ I_2 &= \text{GOTO}(I_0, \forall) = \{ F \rightarrow \forall \cdot \text{id} (F) \} \\ I_3 &= \text{GOTO}(I_0, \text{id}) = \{ F \rightarrow \text{id} \cdot (L) \} \\ I_4 &= \text{GOTO}(I_1, \vee) = \{ F \rightarrow F \vee \cdot F \\ &\quad F \rightarrow \cdot F \vee F \\ &\quad F \rightarrow \cdot \forall \text{id} (F) \\ &\quad F \rightarrow \cdot \text{id} (L) \} \\ I_5 &= \text{GOTO}(I_2, \text{id}) = \{ F \rightarrow \forall \text{id} \cdot (F) \} \\ I_6 &= \text{GOTO}(I_3, () = \{ F \rightarrow \text{id} (\cdot L) \\ &\quad L \rightarrow \cdot \text{id} \\ &\quad L \rightarrow \cdot \text{id} , L \} \\ I_7 &= \text{GOTO}(I_4, F) = \{ F \rightarrow F \vee F \cdot \\ &\quad F \rightarrow F \cdot \vee F \} \\ \text{GOTO}(I_4, \forall) &= I_2 \\ \text{GOTO}(I_4, \text{id}) &= I_3 \\ I_8 &= \text{GOTO}(I_5, () = \{ F \rightarrow \forall \text{id} (\cdot F) \\ &\quad F \rightarrow \cdot F \vee F \\ &\quad F \rightarrow \cdot \forall \text{id} (F) \\ &\quad F \rightarrow \cdot \text{id} (L) \} \\ I_9 &= \text{GOTO}(I_6, L) = \{ F \rightarrow \text{id} (L \cdot) \} \\ I_{10} &= \text{GOTO}(I_6, \text{id}) = \{ L \rightarrow \text{id} \cdot \\ &\quad L \rightarrow \text{id} \cdot , L \} \end{aligned}$$

$$\text{GOTO}(I_7, \vee) = I_4$$

$$I_{11} = \text{GOTO}(I_8, F) = \{ F \rightarrow \vee \text{id} (F \cdot) \\ F \rightarrow F \cdot \vee F \}$$

$$\text{GOTO}(I_8, \vee) = I_2$$

$$\text{GOTO}(I_8, \text{id}) = I_3$$

$$I_{12} = \text{GOTO}(I_9,) = \{ F \rightarrow \text{id} (L) \cdot \}$$

$$I_{13} = \text{GOTO}(I_{10},) = \{ L \rightarrow \text{id} , \cdot L \\ L \rightarrow \cdot \text{id} \\ L \rightarrow \cdot \text{id} , L \}$$

$$I_{14} = \text{GOTO}(I_{11},) = \{ F \rightarrow \vee \text{id} (F) \cdot \}$$

$$\text{GOTO}(I_{11}, \vee) = I_4$$

$$I_{15} = \text{GOTO}(I_{13}, L) = \{ L \rightarrow \text{id} , L \cdot \}$$

$$\text{GOTO}(I_{13}, \text{id}) = I_{10}$$

A partir de la colección anterior, se construye la tabla de análisis SLR es la siguiente:

ESTADO	ACCIÓN						IR-A	
	\vee	\forall	id	()	,	\$	F L
0		d2	d3					1
1	d4						acc.	
2			d5					
3				d6				
4		d2	d3					7
5				d8				
6				d10				9
7	r1/d4				r1		r1	
8		d2	d3					11
9				d12				
10				r4	d13			
11	d4				d14			
12	r3				r3		r3	
13				d10				15
14	r2				r2		r2	
15					r5			

La gramática no es SLR(1), ya que la tabla tiene un conflicto desplaza/reduce en el estado 7. El conflicto era previsible, ya que la gramática es ambigua. Por esta misma razón, no puede ser tampoco LALR(1) ni LR(1).

De los ítems del conjunto I_7 se deduce la acción que hay que seleccionar para que el operador \vee sea asociativo por la izquierda. En la forma sentencial $F \vee F \vee \text{id}$ se produce el conflicto al procesar el segundo operador \vee , y se resuelve haciendo la reducción de $F \vee F$ en lugar de desplazar.

Apartado 4.

La simulación de la cadena $\forall \text{id}(\forall \text{id}(\text{id}))$ con tratamiento de errores en modo pánico es la siguiente:

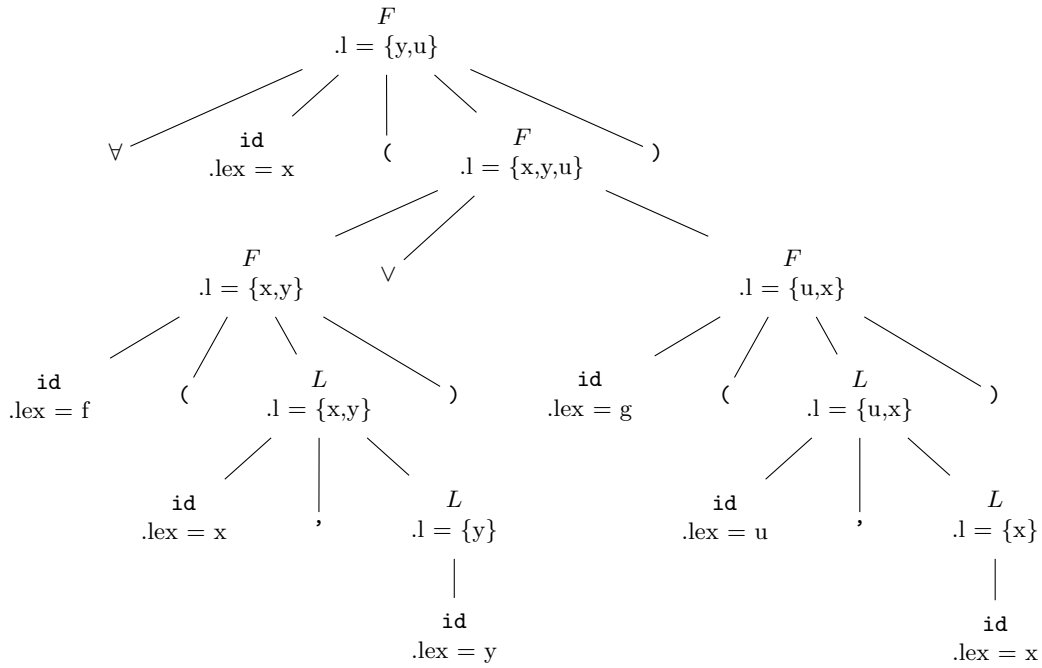
PILA	ENTRADA	ACCIÓN
0	$\forall \text{id}(\forall \text{id}(\text{id})) \$$	d2
0 \forall 2	id($\forall \text{id}(\text{id})) \$$	d5
0 \forall 2 id 5	($\forall \text{id}(\text{id})) \$$	d8
0 \forall 2 id 5 (8	$\forall \text{id}(\text{id})) \$$	Error: apilar F e IR-A[8,F] Como $\vee \in \text{SIG}(F)$, continúa el análisis sin eliminar tokens de la entrada.
0 \forall 2 id 5 (8 F 11	$\forall \text{id}(\text{id})) \$$	d4
0 \forall 2 id 5 (8 F 11 \forall 4	id(id)) \$	d3
0 \forall 2 id 5 (8 F 11 \forall 4 id 3	(id)) \$	d6
0 \forall 2 id 5 (8 F 11 \forall 4 id 3 (6	id)) \$	d10
0 \forall 2 id 5 (8 F 11 \forall 4 id 3 (6 id 10)) \$	r4 $L \rightarrow \text{id}$
0 \forall 2 id 5 (8 F 11 \forall 4 id 3 (6 L 9)) \$	d12
0 \forall 2 id 5 (8 F 11 \forall 4 id 3 (6 L 9) 12) \$	r3 $F \rightarrow \text{id}(L)$
0 \forall 2 id 5 (8 F 11 \forall 4 F 7) \$	r1 $F \rightarrow F \vee F$
0 \forall 2 id 5 (8 F 11) \$	d14
0 \forall 2 id 5 (8 F 11) 14	\$	r2 $F \rightarrow \forall \text{id}(F)$
0 F 1	\$	Fin: no aceptar

Apartado 5.

a) Para realizar la definición dirigida por la sintaxis que se solicita, consideramos que el token `id` tiene un atributo `lex` que contiene su lexema asociado. La definición emplea un atributo `l` asociado a los no terminales F y L , cuyo valor es una lista de variables. La lista de F contiene las variables no ligadas de la fórmula que representa dicho no terminal, mientras que la lista de L contiene las variables que son parámetros de una función. La acción asociada a la regla $F \rightarrow \forall \text{id} (F_1)$ elimina de la lista de F_1 la variable afectada por el cuantificador, y almacena el resultado en la lista de F . En las reglas semánticas, las listas se manejan como conjuntos.

Regla de producción	Regla semántica
$F \rightarrow F_1 \vee F_2$	$F.l = F_1.l \cup F_2.l;$
$F \rightarrow \forall \text{id} (F_1)$	$F.l = F_1.l - \{ \text{id.lex} \};$
$F \rightarrow \text{id} (L)$	$F.l = L.l;$
$L \rightarrow \text{id}$	$L.l = \{ \text{id.lex} \};$
$L \rightarrow \text{id}, L_1$	$L.l = L_1.l \cup \{ \text{id.lex} \};$

b) El árbol anotado para la entrada $\forall x(f(x,y)\vee g(u,x))$ es el siguiente:



c) La DDS anterior define una gramática S-atribuida, porque los atributos que se emplean son todos sintetizados. Y por definición, toda gramática S-atribuida también es L-atribuida.