

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

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

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

La gramática tiene factores comunes en las reglas de producción de los no terminales E y L . Esta es la única razón aparente para que no sea LL(1), puesto que no es ambigua (por la forma de las reglas de producción, está definida la asociatividad de los operadores \cup , \cap y coma para que sea derecha) ni es recursiva por la izquierda.

Eliminamos los factores comunes de E y L , obteniendo la gramática equivalente:

$$\begin{aligned}
 E &\rightarrow C E' \\
 E' &\rightarrow \cup E \mid \cap E \mid \lambda \\
 C &\rightarrow \langle L \rangle \mid (E) \mid \emptyset \\
 L &\rightarrow el L' \\
 L' &\rightarrow \lambda \mid , L
 \end{aligned}$$

Para evitar confusión con la notación de los conjuntos PRIMERO, SIGUIENTE y PREDICT, se han sustituido los símbolos $\{$ y $\}$ por \langle y \rangle . Los conjuntos PRIMERO y SIGUIENTE de los no terminales de la gramática modificada son:

$$\begin{aligned}
 \text{PRIMERO}(E) &= \{ \langle (\emptyset \} \\
 \text{PRIMERO}(E') &= \{ \cup \cap \lambda \} \\
 \text{PRIMERO}(C) &= \{ \langle (\emptyset \} \\
 \text{PRIMERO}(L) &= \{ el \} \\
 \text{PRIMERO}(L') &= \{ \lambda , \} \\
 \text{SIGUIENTE}(E) &= \{) \$ \} \\
 \text{SIGUIENTE}(E') &= \{) \$ \} \\
 \text{SIGUIENTE}(C) &= \{ \cup \cap) \$ \} \\
 \text{SIGUIENTE}(L) &= \{ \rangle \} \\
 \text{SIGUIENTE}(L') &= \{ \rangle \}
 \end{aligned}$$

Los conjuntos PREDICT de las reglas de producción de la gramática modificada son:

$$\begin{aligned}
 \text{PREDICT}(E \rightarrow C E') &= \{ \langle (\emptyset \} \\
 \text{PREDICT}(E' \rightarrow \cup E) &= \{ \cup \} \\
 \text{PREDICT}(E' \rightarrow \cap E) &= \{ \cap \} \\
 \text{PREDICT}(E' \rightarrow \lambda) &= \{) \$ \} \\
 \text{PREDICT}(C \rightarrow \langle L \rangle) &= \{ \langle \} \\
 \text{PREDICT}(C \rightarrow (E)) &= \{ (\} \\
 \text{PREDICT}(C \rightarrow \emptyset) &= \{ \emptyset \} \\
 \text{PREDICT}(L \rightarrow el L') &= \{ el \} \\
 \text{PREDICT}(L' \rightarrow \lambda) &= \{ \rangle \} \\
 \text{PREDICT}(L' \rightarrow , L) &= \{ , \}
 \end{aligned}$$

De acuerdo con los conjuntos PREDICT anteriores, la tabla de análisis LL(1) es la siguiente:

NO TERM	TERMINAL									
	\cup	\cap	\langle	\rangle	$($	$)$	\emptyset	el	$,$	$\$$
E			$E \rightarrow C E'$		$E \rightarrow C E'$		$E \rightarrow C E'$			
E'	$E' \rightarrow \cup E$	$E' \rightarrow \cap E$				$E' \rightarrow \lambda$				$E' \rightarrow \lambda$
C			$C \rightarrow \langle L \rangle$		$C \rightarrow (E)$		$C \rightarrow \emptyset$			
L								$L \rightarrow el L'$		
L'				$L' \rightarrow \lambda$					$L' \rightarrow , L$	

No hay conflictos y, por tanto, la gramática es LL(1).

Apartado 2.

Partiendo de la gramática inicial, calculamos la colección LR(0):

$$\begin{aligned}
I_0 &= \{ E' \rightarrow \cdot E \\
&\quad E \rightarrow \cdot C \\
&\quad E \rightarrow \cdot C \cup E \\
&\quad E \rightarrow \cdot C \cap E \\
&\quad C \rightarrow \cdot \langle L \rangle \\
&\quad C \rightarrow \cdot (E) \\
&\quad C \rightarrow \cdot \emptyset \} \\
I_1 &= \text{GOTO}(I_0, E) = \{ E' \rightarrow E \cdot \} \\
I_2 &= \text{GOTO}(I_0, C) = \{ E \rightarrow C \cdot \\
&\quad E \rightarrow C \cdot \cup E \\
&\quad E \rightarrow C \cdot \cap E \} \\
I_3 &= \text{GOTO}(I_0, \langle \rangle) = \{ C \rightarrow \langle \cdot L \rangle \\
&\quad L \rightarrow \cdot el \\
&\quad L \rightarrow \cdot el, L \} \\
I_4 &= \text{GOTO}(I_0, (\)) = \{ C \rightarrow (\cdot E) \\
&\quad E \rightarrow \cdot C \\
&\quad E \rightarrow \cdot C \cup E \\
&\quad E \rightarrow \cdot C \cap E \\
&\quad C \rightarrow \cdot \langle L \rangle \\
&\quad C \rightarrow \cdot (E) \\
&\quad C \rightarrow \cdot \emptyset \} \\
I_5 &= \text{GOTO}(I_0, \emptyset) = \{ C \rightarrow \emptyset \cdot \} \\
I_6 &= \text{GOTO}(I_2, \cup) = \{ E \rightarrow C \cup \cdot E \\
&\quad E \rightarrow \cdot C \\
&\quad E \rightarrow \cdot C \cup E \\
&\quad E \rightarrow \cdot C \cap E \\
&\quad C \rightarrow \cdot \langle L \rangle \\
&\quad C \rightarrow \cdot (E) \\
&\quad C \rightarrow \cdot \emptyset \} \\
I_7 &= \text{GOTO}(I_2, \cap) = \{ E \rightarrow C \cap \cdot E \\
&\quad E \rightarrow \cdot C \\
&\quad E \rightarrow \cdot C \cup E \\
&\quad E \rightarrow \cdot C \cap E \\
&\quad C \rightarrow \cdot \langle L \rangle \\
&\quad C \rightarrow \cdot (E) \\
&\quad C \rightarrow \cdot \emptyset \} \\
I_8 &= \text{GOTO}(I_3, L) = \{ C \rightarrow \langle L \cdot \rangle \} \\
I_9 &= \text{GOTO}(I_3, el) = \{ L \rightarrow el \cdot \\
&\quad L \rightarrow el \cdot, L \} \\
I_{10} &= \text{GOTO}(I_4, E) = \{ C \rightarrow (E \cdot) \} \\
\text{GOTO}(I_4, C) &= I_2 \\
\text{GOTO}(I_4, \langle \rangle) &= I_3 \\
\text{GOTO}(I_4, (\)) &= I_4 \\
\text{GOTO}(I_4, \emptyset) &= I_5 \\
I_{11} &= \text{GOTO}(I_6, E) = \{ E \rightarrow C \cup E \cdot \} \\
\text{GOTO}(I_6, C) &= I_2 \\
\text{GOTO}(I_6, \langle \rangle) &= I_3 \\
\text{GOTO}(I_6, (\)) &= I_4 \\
\text{GOTO}(I_6, \emptyset) &= I_5 \\
I_{12} &= \text{GOTO}(I_7, E) = \{ E \rightarrow C \cap E \cdot \} \\
\text{GOTO}(I_7, C) &= I_2 \\
\text{GOTO}(I_7, \langle \rangle) &= I_3 \\
\text{GOTO}(I_7, (\)) &= I_4 \\
\text{GOTO}(I_7, \emptyset) &= I_5 \\
I_{13} &= \text{GOTO}(I_8, \rangle) = \{ C \rightarrow \langle L \rangle \cdot \} \\
I_{14} &= \text{GOTO}(I_9, ,) = \{ L \rightarrow el, L \\
&\quad L \rightarrow \cdot el \\
&\quad L \rightarrow \cdot el, L \} \\
I_{15} &= \text{GOTO}(I_{10},) = \{ C \rightarrow (E) \cdot \} \\
I_{16} &= \text{GOTO}(I_{14}, L) = \{ L \rightarrow el, L \cdot \} \\
\text{GOTO}(I_{14}, el) &= I_9
\end{aligned}$$

Para determinar las acciones de reducción, es necesario obtener previamente los conjuntos SIGUIENTE de los no terminales de la gramática inicial:

$$\begin{aligned}
\text{PRIMERO}(E) &= \{ \langle (\emptyset \} \\
\text{PRIMERO}(C) &= \{ \langle (\emptyset \} \\
\text{PRIMERO}(L) &= \{ el \} \\
\text{SIGUIENTE}(E) &= \{) \$ \} \\
\text{SIGUIENTE}(C) &= \{ \cup \cap) \$ \} \\
\text{SIGUIENTE}(L) &= \{ \rangle \}
\end{aligned}$$

A partir de los cálculos anteriores, podemos construir la siguiente tabla de análisis SLR:

ESTADO	ACCIÓN										IR-A		
	\cup	\cap	\langle	\rangle	$($	$)$	\emptyset	el	$,$	$\$$	E	C	L
0			d3		d4		d5				1	2	
1										acc			
2	d6	d7					r1			r1			
3								d9					8
4			d3		d4		d5				10	2	
5	r6	r6					r6			r6			
6			d3		d4		d5				11	2	
7			d3		d4		d5				12	2	
8				d13									
9				r7					d14				
10							d15						
11							r2			r2			
12							r3			r3			
13	r4	r4					r4			r4			
14								d9					16
15	r5	r5					r5			r5			
16				r8									

La numeración de las reglas de producción usada en la tabla es la siguiente:

1. $E \rightarrow C$
2. $E \rightarrow C \cup E$
3. $E \rightarrow C \cap E$
4. $C \rightarrow \langle L \rangle$
5. $C \rightarrow (E)$
6. $C \rightarrow \emptyset$
7. $L \rightarrow el$
8. $L \rightarrow el , L$

Al no aparecer ningún conflicto en la tabla, podemos deducir que la gramática es SLR(1).

Apartado 3.

Por construcción de los métodos de análisis, toda gramática SLR(1) es también LALR(1) y LR(1).

Apartado 4.

La simulación de la cadena $\{a, b\}$ con tratamiento de errores en modo pánico es la siguiente:

PILA	ENTRADA	ACCIÓN
0	$\langle el, el \rangle \$$	$d3$
0 $\langle 3$	$el, el \rangle \$$	$d9$
0 $\langle 3 el 9$	$, el \rangle \$$	$d14$
0 $\langle 3 el 9 , 14$	$el \rangle \$$	$d9$
0 $\langle 3 el 9 , 14 el 9$	$\rangle \$$	$r7 L \rightarrow el$
0 $\langle 3 el 9 , 14 L 16$	$\rangle \$$	$r8 L \rightarrow el , L$
0 $\langle 3 L 8$	$\rangle \$$	$d13$
0 $\langle 3 L 8 \rangle 13$	$\rangle \$$	Error: desapilar hasta estado con IR-A definido,
0 $\langle 3$	$\rangle \$$	apilar L y descartar símbolos hasta SIGUIENTE(L)
0 $\langle 3 L 8$	$\rangle \$$	$d13$
0 $\langle 3 L 8 \rangle 13$	$\$$	$r4 C \rightarrow \langle L \rangle$
0 $C 2$	$\$$	$r1 E \rightarrow C$
0 $E 1$	$\$$	Fin: no aceptar la entrada.

Apartado 5.

Para realizar la definición dirigida por la sintaxis que se solicita, emplearemos los siguientes atributos:

Símbolo	Atributo	Descripción
el	tipo	Tipo del elemento
L	tipo	Tipo de la lista de elementos
C	tipo	Tipo del conjunto
E	tipo	Tipo de la expresión

Los atributos **tipo** para los distintos símbolos pueden tomar los siguientes valores:

1. **cadena** : el tipo del símbolo es una cadena o un conjunto de cadenas.
2. **id** : el tipo del símbolo es un identificador o un conjunto de identificadores.
3. **número** : el tipo del símbolo es un número o un conjunto de números.
4. **undef** : el tipo del símbolo no está definido (por el uso de \emptyset , para los no terminales E y C únicamente).
5. **error** : el tipo del símbolo es erróneo (por mezcla de tipos).

Las reglas semánticas para la evaluación del atributo **tipo** de cada símbolo son:

Regla de producción	Regla semántica
$E' \rightarrow E$	if ($E.tipo \neq error$) printf("OK"); else printf("Error de tipos");
$E \rightarrow C$	$E.tipo = C.tipo$;
$E \rightarrow C \cup E_1$	if ($C.tipo == undef$) $E.tipo = E_1.tipo$; else if ($E_1.tipo == undef$) $E.tipo = C.tipo$; else if ($C.tipo == E_1.tipo$) $E.tipo = C.tipo$; else $E.tipo = error$;
$E \rightarrow C \cap E_1$	if ($C.tipo == undef$) $E.tipo = E_1.tipo$; else if ($E_1.tipo == undef$) $E.tipo = C.tipo$; else if ($C.tipo == E_1.tipo$) $E.tipo = C.tipo$; else $E.tipo = error$;
$C \rightarrow \{ L \}$	$C.tipo = L.tipo$;
$C \rightarrow (E)$	$C.tipo = E.tipo$;
$C \rightarrow \emptyset$	$C.tipo = undef$;
$L \rightarrow el$	$L.tipo = el.tipo$;
$L \rightarrow el, L_1$	if ($el.tipo == L_1.tipo$) $L.tipo = el.tipo$; else $L.tipo = error$;

Por el modo de evaluar los atributos **tipo**, todos ellos son sintetizados. Por tanto, la gramática atribuida es S-atribuida. También es L-atribuida, porque por definición las gramáticas L-atribuidas incluyen a las S-atribuidas.

Con esta DDS, para la cadena solicitada en el enunciado, podemos construir un árbol sintáctico decorado como el que se indica en la siguiente página.

