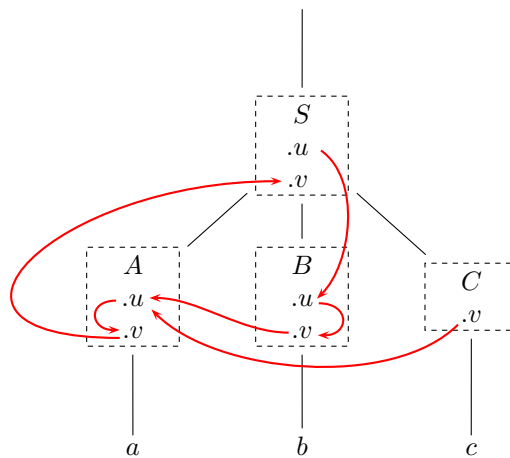


**SOLUCIONES****Parte I: PREGUNTAS TIPO TEST. 30%.**

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

Parte II: PREGUNTAS CORTAS. 10%.

1.

2. El valor de $S.v$ al terminar la evaluación sería 8.**Parte III: PROBLEMA. 60%.****Apartado 1.**

La gramática G

$$\begin{aligned}
 S &\rightarrow id = E \\
 E &\rightarrow id \\
 &\quad | \quad id(E, E, E)
 \end{aligned}$$

no puede ser LL(1), puesto que no está *factorizada*: las reglas segunda y tercera tienen factor común. Por otro lado, la gramática no es *recursiva por la izquierda* ni *ambigua*.

Para intentar conseguir una gramática equivalente que sea LL(1), tendremos que factorizar, quedando la siguiente gramática G':

$$\begin{aligned}
 S &\rightarrow id = E \\
 E &\rightarrow id E' \\
 E' &\rightarrow \lambda \\
 &\quad | \quad (E, E, E)
 \end{aligned}$$

Veamos si G' es LL(1). Calculamos los conjuntos PRIMERO y SIGUIENTE para los no terminales:

$$\begin{aligned}
 \text{PRIMERO}(S) &= \{ id \} & \text{SIGUIENTE}(S) &= \{ \$ \} \\
 \text{PRIMERO}(E) &= \{ id \} & \text{SIGUIENTE}(E) &= \{ \$, , \} \\
 \text{PRIMERO}(E') &= \{ \lambda \} & \text{SIGUIENTE}(E') &= \{ \$, , \}
 \end{aligned}$$

Y ahora, los conjuntos *predict* para cada regla:

$$\begin{aligned} \text{predict}(1) &= \{ \text{id} \} & \text{predict}(3) &= \{ \$,) \} \\ \text{predict}(2) &= \{ \text{id} \} & \text{predict}(4) &= \{ (\} \end{aligned}$$

Puesto que $\text{predict}(3) \cap \text{predict}(4) = \emptyset$, la gramática G' es LL(1).

Apartado 2.

Para demostrar que G es SLR(1), se debe aplicar dicho método a la gramática para comprobar que la tabla de análisis no tiene conflictos. Numeramos las reglas de producción de la siguiente forma:

- (1) $S \rightarrow \text{id} = E$
- (2) $E \rightarrow \text{id}$
- (3) $E \rightarrow \text{id} (E, E, E)$

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

$$\begin{aligned} I_0 &= \{ [S' \rightarrow \cdot S] \\ &\quad [S \rightarrow \cdot \text{id} = E] \} \\ I_1 &= \text{GOTO}(I_0, S) = \{ [S' \rightarrow S \cdot] \} \\ I_2 &= \text{GOTO}(I_0, \text{id}) = \{ [S \rightarrow \text{id} \cdot = E] \} \\ I_3 &= \text{GOTO}(I_2, =) = \{ [S \rightarrow \text{id} = \cdot E] \\ &\quad [E \rightarrow \cdot \text{id}] \\ &\quad [E \rightarrow \cdot \text{id} (E, E, E)] \} \\ I_4 &= \text{GOTO}(I_3, E) = \{ [S \rightarrow \text{id} = E \cdot] \} \\ I_5 &= \text{GOTO}(I_3, \text{id}) = \{ [E \rightarrow \text{id} \cdot] \\ &\quad [E \rightarrow \text{id} \cdot (E, E, E)] \} \\ I_6 &= \text{GOTO}(I_5, () = \{ [E \rightarrow \text{id} (\cdot E, E, E)] \\ &\quad [E \rightarrow \cdot \text{id}] \\ &\quad [E \rightarrow \cdot \text{id} (E, E, E)] \} \\ I_7 &= \text{GOTO}(I_6, E) = \{ [E \rightarrow \text{id} (E \cdot, E, E)] \} \\ \text{GOTO}(I_6, \text{id}) &= I_5 \\ I_8 &= \text{GOTO}(I_7,) = \{ [E \rightarrow \text{id} (E, \cdot E, E)] \\ &\quad [E \rightarrow \cdot \text{id}] \\ &\quad [E \rightarrow \cdot \text{id} (E, E, E)] \} \\ I_9 &= \text{GOTO}(I_8, E) = \{ [E \rightarrow \text{id} (E, E \cdot, E)] \} \\ \text{GOTO}(I_8, \text{id}) &= I_5 \\ I_{10} &= \text{GOTO}(I_9,) = \{ [E \rightarrow \text{id} (E, E, \cdot E)] \\ &\quad [E \rightarrow \cdot \text{id}] \\ &\quad [E \rightarrow \cdot \text{id} (E, E, E)] \} \\ I_{11} &= \text{GOTO}(I_{10}, E) = \{ [E \rightarrow \text{id} (E, E, E \cdot)] \} \\ \text{GOTO}(I_{10}, \text{id}) &= I_5 \\ I_{12} &= \text{GOTO}(I_{11},) = \{ [E \rightarrow \text{id} (E, E, E) \cdot] \} \end{aligned}$$

Calculamos los conjuntos SIGUIENTE para los no terminales:

$$\text{SIGUIENTE}(S) = \{ \$ \}$$

$$\text{SIGUIENTE}(E) = \{ \$,) \}$$

La tabla de análisis SLR es la siguiente:

ESTADO	ACCIÓN					IR-A	
	()	,	=	id	\$	E S
0					d2		1
1						acc	
2				d3			
3					d5		4
4						r1	
5	d6	r2	r2			r2	
6					d5		7
7				d8			
8					d5		9
9				d10			
10					d5		11
11		d12					
12		r3	r3			r3	

La gramática es SLR, ya que la tabla no contiene ningún conflicto.

Apartado 3.

La simulación de la cadena de entrada $id = id (id , id id ,)$ usando tratamiento de errores en modo pánico es la siguiente:

PILA	ENTRADA	ACCIÓN
0	$id = id (id , id id ,) \$$	$d2$
0 id 2	$= id (id , id id ,) \$$	$d3$
0 id 2 = 3	$id (id , id id ,) \$$	$d5$
0 id 2 = 3 id 5	$(id , id id ,) \$$	$d6$
0 id 2 = 3 id 5 (6	$id , id id ,) \$$	$d5$
0 id 2 = 3 id 5 (6 id 5	$, id id ,) \$$	$r2$
0 id 2 = 3 id 5 (6 E 7	$, id id ,) \$$	$d8$
0 id 2 = 3 id 5 (6 E 7 , 8	$id id ,) \$$	$d5$
0 id 2 = 3 id 5 (6 E 7 , 8 id 5	$id ,) \$$	Error: desapila hasta estado 8 y apila E Eliminar tokens hasta , $\in SIG(E)$
0 id 2 = 3 id 5 (6 E 7 , 8 E 9	$,) \$$	$d10$
0 id 2 = 3 id 5 (6 E 7 , 8 E 9 , 10	$) \$$	Error: apila E
0 id 2 = 3 id 5 (6 E 7 , 8 E 9 , 10 E 11	$) \$$	$d12$
0 id 2 = 3 id 5 (6 E 7 , 8 E 9 , 10 E 11) 12	$\$$	$r3$
0 id 2 = 3 E 4	$\$$	$r1$
0 S 1	$\$$	Fin: entrada con errores

Apartado 4.

- La gramática es SLR, por tanto, también LALR y LR-Canónica. Esto se debe a que el conjunto de las gramáticas SLR está incluido en el de las gramáticas LALR, y éste en el de las LR-Canónicas.
- Las tablas SLR y LALR siempre tienen el mismo tamaño para la misma gramática, puesto que el número de conjuntos de items coincide necesariamente. No puede haber dos conjuntos en una colección LALR con la misma parte izquierda y diferenciándose sólo por los tokens de anticipación, igual que ocurre con la colección SLR. Sin embargo, en una colección LR(1) sí es posible que ocurra esto, es decir, que existan dos conjuntos idénticos salvo por los tokens de anticipación de sus items. Cuando se da este caso, la colección LR(1) y la LALR(1) no tendrán el mismo tamaño y, por tanto, tampoco las tablas. En el caso de la gramática del problema:

$$\begin{aligned}
 S &\rightarrow id = E \\
 E &\rightarrow id \\
 &\mid id(E, E, E)
 \end{aligned}$$

ocurrirá esto, puesto que el símbolo no terminal E aparece en contextos diferentes, de manera que aparecerán conjuntos de items iguales con distintos tokens de anticipación. De hecho, la colección LR(1) para G tiene 29 estados. Por ejemplo, aparecerán tres conjuntos:

$$\begin{aligned}
 I_i = \text{GOTO}(I_9,) = \{ &[E \rightarrow id \cdot , \$] \\
 &[E \rightarrow id \cdot (E , E , E) , \$] \}
 \end{aligned}$$

$$\begin{aligned}
 I_j = \text{GOTO}(I_9,) = \{ &[E \rightarrow id \cdot , ,] \\
 &[E \rightarrow id \cdot (E , E , E) , ,] \}
 \end{aligned}$$

$$\begin{aligned}
 I_k = \text{GOTO}(I_9,) = \{ &[E \rightarrow id \cdot ,)] \\
 &[E \rightarrow id \cdot (E , E , E) ,)] \}
 \end{aligned}$$

según que los estados se obtengan a partir de la clausura del símbolo E cuando aparece en la regla de producción 1, cuando aparece antes de la coma en la regla 3, o bien cuando aparece antes del paréntesis en esa misma regla. Estos tres estados se fusionarán en uno al realizar la colección LALR:

$$\begin{aligned}
 I_{ijk} = \text{GOTO}(I_9,) = \{ &[E \rightarrow id \cdot , \$,)] \\
 &[E \rightarrow id \cdot (E , E , E) , \$,)] \}
 \end{aligned}$$

Esto ocurrirá con otros muchos estados.

c) La gramática

$$\begin{array}{lcl} S & \rightarrow & id = E \\ & | & S; S \\ & | & \lambda \\ E & \rightarrow & id \\ & | & id(E, E, E) \end{array}$$

no puede ser LALR ni LR-Canónica, puesto que es ambigua al tener una regla de producción ambigua, la segunda.

Apartado 5.

Suponemos que disponemos de las siguientes funciones:

- `buscaTipo(ap)`, que devuelve el tipo asociado al identificador al que apunta `ap` en la *tabla de símbolos*.
- `buscaLexema(ap)`, que devuelve el lexema asociado con el identificador apuntado por `ap` en la *tabla de símbolos*.

Por otra parte, se definen los siguientes atributos:

- `ap` asociado al símbolo terminal *id*. Apunta a su posición en la *tabla de símbolos*.
- `tipo` asociado a los símbolos no terminales *S* y *E*. Puede tomar los valores `lógico`, `natural`, `correcto` o `error`.

Suponemos que los tipos de los identificadores están almacenados en la *tabla de símbolos* antes de comenzar la comprobación. Teniendo esto en cuenta, el *comprobador de tipos* podría ser el siguiente:

Regla de producción	Regla semántica
$S \rightarrow id = E$	if (buscaTipo(id.ap)==E.tipo) then S.tipo=correcto; else S.tipo=error;
$E \rightarrow id$	E.tipo = buscaTipo(id.ap);
$E \rightarrow id (E_1 , E_2 , E_3)$	if (buscaLexema(id.ap)==potMul) then if (E ₁ .tipo!=error && E ₂ .tipo!=error && E ₃ .tipo!=error) then E.tipo=natural; else E.tipo=error; else if (buscaLexema(id.ap)==dosT) then if (E ₁ .tipo==lógico && E ₂ .tipo==lógico && E ₃ .tipo==lógico) then E.tipo=lógico; else E.tipo=error;