

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

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

Parte II: PREGUNTAS CORTAS. 70%.

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

Símbolo	Atributo	Descripción
S	res	Valor boolean que indica si se ha encontrado el número en el árbol
B	res	Valor boolean que indica si el subárbol B contiene $B.v$
B	v	Valor numérico que se tiene que buscar en B
num	v	Valor numérico del lexema representado por num

La siguiente DDS permite implementar la operación de búsqueda, dejando en $S.res$ el resultado de la misma.

Regla de producción	Regla semántica
$S \rightarrow \text{num} \in B$	$B.v = \text{num}.v; S.res = B.res$
$B \rightarrow (B_1 \text{ num } B_2)$	$B_1.v = B.v; B_2.v = B.v;$ if (num.v == $B.v$) $B.res = \text{true};$ else if (num.v < $B.v$) $B.res = B_1.res;$ else $B.res = B_2.res;$
$B \rightarrow \lambda$	$B.res = \text{false};$

Por el modo de evaluar los atributos, todos ellos son sintetizados salvo $B.v$, que es heredado de un hermano izquierdo. Por tanto, la gramática es L-atribuida.

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

La gramática no es LL(1) porque:

- Es ambigua. Por ejemplo, la regla $R \rightarrow R R$ permite dos árboles de derivación para la cadena aaa .
- Es recursiva por la izquierda, por las reglas $R \rightarrow R R$, $R \rightarrow R /$ y $R \rightarrow R *$.
- Tiene factores comunes en las tres reglas anteriores.

Intentamos obtener una gramática equivalente que sea LL(1). Comenzamos eliminando la ambigüedad, teniendo en cuenta la precedencia de los operadores de las expresiones regulares. La mayor precedencia es para la clausura, seguida de la concatenación y, por último, la unión. Consideramos también la asociatividad izquierda en los operadores. Resolvemos el problema de la ambigüedad usando una estrategia similar a la que aparece en la gramática de expresiones aritméticas:

$$\begin{aligned}
E &\rightarrow E / T \mid T \\
T &\rightarrow T F \mid F \\
F &\rightarrow F * \mid a \mid b
\end{aligned}$$

El siguiente paso consiste en la aplicación del algoritmo de eliminación de la recursividad izquierda:

$$\begin{aligned}
E &\rightarrow T \mid T E' \\
E' &\rightarrow / T \mid / T E' \\
T &\rightarrow F \mid F T' \\
T' &\rightarrow F \mid F T' \\
F &\rightarrow a \mid b \mid a F' \mid b F' \\
F' &\rightarrow * \mid * F'
\end{aligned}$$

Es posible una simplificación de las reglas de T y T' , puesto que ambos terminales derivan lo mismo:

$$\begin{aligned}
E &\rightarrow T \mid T E' \\
E' &\rightarrow / T \mid / T E' \\
T &\rightarrow F \mid F T \\
F &\rightarrow a \mid b \mid a F' \mid b F' \\
F' &\rightarrow * \mid * F'
\end{aligned}$$

Pasamos ahora a la eliminación de los factores comunes:

$$\begin{aligned}
E &\rightarrow T E'' \\
E' &\rightarrow / T E'' \\
E'' &\rightarrow \lambda \mid E' \\
T &\rightarrow F T' \\
T' &\rightarrow \lambda \mid T \\
F &\rightarrow a F'' \mid b F'' \\
F' &\rightarrow * F'' \\
F'' &\rightarrow \lambda \mid F'
\end{aligned}$$

Simplificamos la gramática para evitar no terminales con una única opción y que sólo se emplean en un lado derecho:

$$\begin{aligned}
E &\rightarrow T E' \\
E' &\rightarrow \lambda \mid / T E' \\
T &\rightarrow F T' \\
T' &\rightarrow \lambda \mid T \\
F &\rightarrow a F' \mid b F' \\
F' &\rightarrow \lambda \mid * F'
\end{aligned}$$

Los conjuntos PRIMERO y SIGUIENTE son:

PRIMERO(E) = { a, b }	SIGUIENTE(E) = { $\$$ }
PRIMERO(E') = { $\lambda, /$ }	SIGUIENTE(E') = { $\$$ }
PRIMERO(T) = { a, b }	SIGUIENTE(T) = { $/, \$$ }
PRIMERO(T') = { λ, a, b }	SIGUIENTE(T') = { $/, \$$ }
PRIMERO(F) = { a, b }	SIGUIENTE(F) = { $a, b, /, \$$ }
PRIMERO(F') = { $\lambda, *$ }	SIGUIENTE(F') = { $a, b, /, \$$ }

Los conjuntos PREDICT son:

PREDICT($E \rightarrow T E'$) = { a, b }	PREDICT($T' \rightarrow T$) = { a, b }
PREDICT($E' \rightarrow \lambda$) = { $\$$ }	PREDICT($F \rightarrow a F'$) = { a }
PREDICT($E' \rightarrow / T E'$) = { $/$ }	PREDICT($F \rightarrow b F'$) = { b }
PREDICT($T \rightarrow F T'$) = { a, b }	PREDICT($F' \rightarrow \lambda$) = { $a, b, /, \$$ }
PREDICT($T' \rightarrow \lambda$) = { $/, \$$ }	PREDICT($F' \rightarrow * F'$) = { $*$ }

Pasamos la información de los conjuntos PREDICT a la tabla LL(1):

NO TERM	TERMINAL				
	a	b	$/$	$*$	$\$$
E	$E \rightarrow T E'$	$E \rightarrow T E'$			
E'			$E' \rightarrow / T E'$		$E' \rightarrow \lambda$
T	$T \rightarrow F T'$	$T \rightarrow F T'$			
T'	$T' \rightarrow T$	$T' \rightarrow T$	$T' \rightarrow \lambda$		$T' \rightarrow \lambda$
F	$F \rightarrow a F'$	$F \rightarrow b F'$			
F'	$F' \rightarrow \lambda$	$F' \rightarrow \lambda$	$F' \rightarrow \lambda$	$F' \rightarrow * F'$	$F' \rightarrow \lambda$

No hay conflictos y, por tanto, podemos concluir que **la gramática modificada es LL(1)**.

Apartado 2.

Calculamos la colección LR(0) de la gramática G es la siguiente:

$$\begin{aligned}
I_0 &= \{ R' \rightarrow \cdot R \\
&\quad R \rightarrow \cdot R R \\
&\quad R \rightarrow \cdot R / R \\
&\quad R \rightarrow \cdot R * \\
&\quad R \rightarrow \cdot a \\
&\quad R \rightarrow \cdot b \} \\
I_1 &= \text{GOTO}(I_0, R) = \\
&\{ R' \rightarrow R \cdot \\
&\quad R \rightarrow R \cdot R \\
&\quad R \rightarrow R \cdot / R \\
&\quad R \rightarrow R \cdot * \\
&\quad R \rightarrow \cdot R R \\
&\quad R \rightarrow \cdot R / R \\
&\quad R \rightarrow \cdot R * \\
&\quad R \rightarrow \cdot a \\
&\quad R \rightarrow \cdot b \} \\
I_2 &= \text{GOTO}(I_0, a) = \\
&\{ R \rightarrow a \cdot \} \\
I_3 &= \text{GOTO}(I_0, b) = \\
&\{ R \rightarrow b \cdot \} \\
I_4 &= \text{GOTO}(I_1, R) = \\
&\{ R \rightarrow R R \cdot \\
&\quad R \rightarrow R \cdot R \\
&\quad R \rightarrow R \cdot / R \\
&\quad R \rightarrow R \cdot * \\
&\quad R \rightarrow \cdot R R \\
&\quad R \rightarrow \cdot R / R \\
&\quad R \rightarrow \cdot R * \\
&\quad R \rightarrow \cdot a \\
&\quad R \rightarrow \cdot b \} \\
I_5 &= \text{GOTO}(I_1, /) = \\
&\{ R \rightarrow R / \cdot R \\
&\quad R \rightarrow \cdot R R \\
&\quad R \rightarrow \cdot R / R \\
&\quad R \rightarrow \cdot R * \\
&\quad R \rightarrow \cdot a \\
&\quad R \rightarrow \cdot b \} \\
I_6 &= \text{GOTO}(I_1, *) = \\
&\{ R \rightarrow R * \cdot \} \\
I_7 &= \text{GOTO}(I_5, R) = \\
&\{ R \rightarrow R / R \cdot \\
&\quad R \rightarrow R \cdot R \\
&\quad R \rightarrow R \cdot / R \\
&\quad R \rightarrow R \cdot * \\
&\quad R \rightarrow \cdot R R \\
&\quad R \rightarrow \cdot R / R \\
&\quad R \rightarrow \cdot R * \\
&\quad R \rightarrow \cdot a \\
&\quad R \rightarrow \cdot b \} \\
\text{GOTO}(I_4, *) &= I_6 \\
\text{GOTO}(I_4, a) &= I_2 \\
\text{GOTO}(I_4, b) &= I_3 \\
\text{GOTO}(I_5, a) &= I_2 \\
\text{GOTO}(I_5, b) &= I_3 \\
\text{GOTO}(I_7, R) &= I_4 \\
\text{GOTO}(I_7, /) &= I_5 \\
\text{GOTO}(I_7, *) &= I_6 \\
\text{GOTO}(I_7, a) &= I_2 \\
\text{GOTO}(I_7, b) &= I_3 \\
\text{GOTO}(I_4, /) &= I_5
\end{aligned}$$

A partir de los cálculos anteriores, y teniendo en cuenta que:

$$\text{PRIMERO}(R) = \{ a, b \} \quad \text{SIGUIENTE}(R) = \{ \$, /, *, a, b \}$$

podemos construir la siguiente tabla de análisis SLR:

ESTADO	ACCIÓN					IR-A
	/	*	a	b	\$	
0			d2	d3		1
1	d5	d6	d2	d3	acep.	4
2	r4	r4	r4	r4	r4	
3	r5	r5	r5	r5	r5	
4	d5/r1	d6/r1	d2/r1	d3/r1	r1	4
5			d2	d3		7
6	r3	r3	r3	r3	r3	
7	d5/r2	d6/r2	d2/r2	d3/r2	r2	4

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

1. $R \rightarrow R R$
2. $R \rightarrow R / R$
3. $R \rightarrow R *$
4. $R \rightarrow a$
5. $R \rightarrow b$

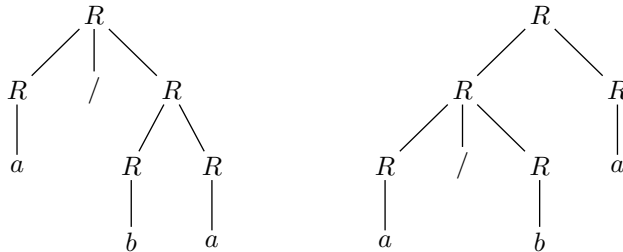
La gramática **no es SLR**. Al ser ambigua, **tampoco es LR-canónica ni LALR**.

Apartado 3.

La resolución de los conflictos, de acuerdo con la precedencia y asociatividad de los operadores de expresiones regulares, es la siguiente:

- En la pila hay una concatenación, por el ítem $R \rightarrow R R \cdot$ del estado 4.
 - $T[4,/] = r1$. Precedencia de la concatenación sobre la unión.
 - $T[4,*] = d6$. Precedencia de la clausura sobre la concatenación.
 - $T[4,a] = r1$. Asociatividad izquierda de la concatenación.
 - $T[4,b] = r1$. Asociatividad izquierda de la concatenación.
- En la pila hay una unión, por el ítem $R \rightarrow R / R \cdot$ del estado 7.
 - $T[7,/] = r2$. Asociatividad izquierda de la unión.
 - $T[7,*] = d6$. Precedencia de la clausura sobre la unión.
 - $T[7,a] = d2$. Precedencia de la concatenación sobre la unión.
 - $T[7,b] = d3$. Precedencia de la concatenación sobre la unión.

Los dos árboles de derivación posibles para la entrada a/ba son:



El árbol correcto es el izquierdo, considerando la precedencia y asociatividad de los operadores de expresiones regulares.

Apartado 4.

La simulación de la cadena a/ba es la siguiente:

PILA	ENTRADA	ACCIÓN
0	$a / b a \$$	d2
0 a 2	$/ b a \$$	r4 $R \rightarrow a$
0 R 1	$/ b a \$$	d5
0 R 1 / 5	$b a \$$	d3
0 R 1 / 5 b 3	$a \$$	r5 $R \rightarrow b$
0 R 1 / 5 R 7	$a \$$	d2
0 R 1 / 5 R 7 a 2	$\$$	r4 $R \rightarrow a$
0 R 1 / 5 R 7 R 4	$\$$	r1 $R \rightarrow R R$
0 R 1 / 5 R 7	$\$$	r2 $R \rightarrow R / R$
0 R 1	$\$$	Aceptar

La secuencia de derivaciones más a la derecha se corresponde con el orden inverso de las reglas de producción empleadas en las reducciones de la simulación:

$$R \Rightarrow R / R \Rightarrow R / R R \Rightarrow R / R a \\ \Rightarrow R / b a \Rightarrow a / b a$$

El árbol de derivación es el primero de los dos indicados en el apartado anterior.