



SOLUCIONES

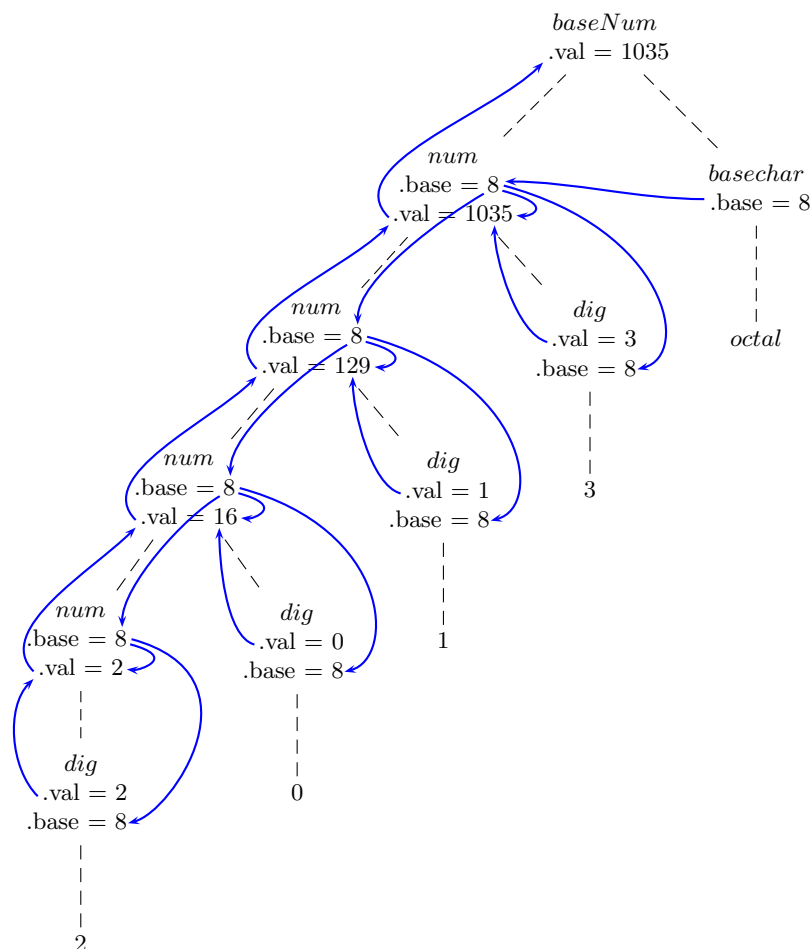
Parte I: PREGUNTAS TIPO TEST. 30%.

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

Parte II: PREGUNTAS CORTAS. 10%.

Apartado 1.

El grafo de dependencias de la entrada 2013 octal se puede representar sobre el árbol de análisis de dicha entrada:



Apartado 2.

Los atributos *base* de los no terminales *num* y *dig* son heredados. Los atributos *val* de los no terminales *num* y *baseNum*, así como el atributo *base* del no terminal *basechar*, son sintetizados.

Apartado 3.

Al tener atributos heredados, la gramática no es S-atribuida. Además, el atributo *base* del no terminal *num* es heredado por la derecha en la acción semántica asociada a la regla de producción $baseNum \rightarrow num\ basechar$, y por tanto la gramática no es L-atribuida, ya que para serlo los atributos heredados tienen que calcularse en función de atributos del padre o de hermanos izquierdos.

Parte III: PROBLEMA. 60%

Apartado 1.

La gramática no es LL(1) porque tiene factores comunes en las reglas del no terminal L . Si aplicamos el algoritmo de eliminación de factores comunes, obtenemos la siguiente gramática con un nuevo símbolo no terminal L' :

- (1) $P \rightarrow [L] (num)$
- (2) $L \rightarrow E L'$
- (3) $L' \rightarrow \lambda$
- (4) $L' \rightarrow , L$
- (5) $E \rightarrow num$
- (6) $E \rightarrow P$

Los conjuntos PRIMERO y SIGUIENTE de la gramática son los siguientes (la coma en los conjuntos representa al terminal de la gramática):

$$\begin{array}{llll} \text{PRIMERO}(P) = \{ [] \} & \text{PRIMERO}(L') = \{ \lambda , \} & \text{SIGUIENTE}(P) = \{ \$,] \} & \text{SIGUIENTE}(L') = \{] \} \\ \text{PRIMERO}(L) = \{ num [] \} & \text{PRIMERO}(E) = \{ num [] \} & \text{SIGUIENTE}(L) = \{] \} & \text{SIGUIENTE}(E) = \{ ,] \} \end{array}$$

Los conjuntos *Predict* de cada regla de la gramática transformada son:

$$\begin{array}{lll} \text{REDICT}(1) = \{ [] \} & \text{PREDICT}(3) = \{] \} & \text{PREDICT}(5) = \{ num \} \\ \text{PREDICT}(2) = \{ num [] \} & \text{PREDICT}(4) = \{ , \} & \text{PREDICT}(6) = \{ [] \} \end{array}$$

Se puede comprobar si la gramática es LL(1) observando si los conjuntos PREDICT de las reglas de un mismo no terminal tienen intersección vacía. Para la gramática del ejercicio, esto sólo es necesario comprobarlo con las reglas de L' (3 y 4), y las reglas de E (5 y 6):

$$\begin{array}{l} \text{PREDICT}(3) \cap \text{PREDICT}(4) = \emptyset \\ \text{PREDICT}(5) \cap \text{PREDICT}(6) = \emptyset \end{array}$$

Por tanto, la gramática es LL(1).

Apartado 2.

Volviendo a la gramática inicial, los conjuntos de ítems de la colección LR(0) son los siguientes:

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

Los conjuntos PRIMERO y SIGUIENTE de los no terminales de la gramática original coinciden con los calculados para la gramática factorizada. Por tanto, la tabla de análisis es la siguiente:

ESTADO	ACCIÓN					IR-A		
	()	,	[]	<i>num</i>	\$	
0					d2			1
1							acc	
2					d2	d5		4 3 6
3					d7			
4					d8	r2		
5					r4	r4		
6					r5	r5		
7	d9							
8					d2	d5		4 10 6
9						d11		
10						r3		
11		d12						
12			r1	r1		r1		

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

Apartado 3.

A continuación se muestra la simulación de la cadena de entrada $[2(3)](1)$ con tratamiento de errores en modo pánico:

PILA	ENTRADA	ACCIÓN
0	[<i>num</i> (<i>num</i>)] (<i>num</i>) \$	d2
0 [2	<i>num</i> (<i>num</i>)] (<i>num</i>) \$	d5
0 [2 <i>num</i> 5	(<i>num</i>)] (<i>num</i>) \$	Error: desapilar hasta un estado con IR-A definido. El estado 2 cumple esa condición. Apilamos E y eliminamos tokens hasta un seguidor de E.
0 [2 E 4] (<i>num</i>) \$	r2 $L \rightarrow E$
0 [2 L 3] (<i>num</i>) \$	d7
0 [2 L 3] 7	(<i>num</i>) \$	d9
0 [2 L 3] 7 (9	<i>num</i>) \$	d11
0 [2 L 3] 7 (9 <i>num</i> 11) \$	d12
0 [2 L 3] 7 (9 <i>num</i> 11) 12	\$	r1 $P \rightarrow [L](num)$
0 P 1	\$	Acep.

Apartado 4.

Al ser SLR, la gramática es también LALR y LR-Canónica.

Apartado 5.

Para la resolución de este apartado emplearemos los siguientes atributos de los símbolos de la gramática:

Símbolo	Atributo	Tipo	Comentario
<i>P</i>	val	float	Valor que toma el polinomio al evaluarlo.
<i>L</i>	coefs	list	Lista de coeficientes del polinomio.
<i>E</i>	val	float	Valor que toma la expresión al evaluarla.
num	val	float	Valor numérico del lexema asociado al token.

Asumiremos que el tipo de dato **list** permite las siguientes operaciones:

- Crear una lista a partir de un valor. Para ello, se rodea el valor de llaves. Por ejemplo: {x} representa una lista formada por un único elemento x.
- Acceso a un valor de la lista. Se puede extraer el valor de la posición i-ésima de una lista L con la notación L[i]. El primer elemento de la lista es el situado más a la izquierda, y su índice es 1.
- Concatenar una lista con otra lista. Asumiremos que la operación de concatenación de listas ya está implementada, y se representa con +. Por tanto, L1+L2 representa la concatenación de las listas L1 y L2.

- Consultar la longitud de una lista. Se puede consultar la longitud de una lista L con la operación $|L|$.

La definición dirigida por la sintaxis (DDS) para evaluar polinomios se puede definir de la siguiente forma:

Regla de producción	Acción
$P \rightarrow [L](num)$	<pre>int exp = 0; P.val = 0; do { P.val += L.coefs[L.coefs -exp]*(num.val)^exp; exp++; } while (exp < L.coefs);</pre>
$L \rightarrow E$	$L.coefs = \{E.val\};$
$L \rightarrow E, L_1$	$L.coefs = \{E.val\} + L_1.coefs;$
$E \rightarrow num$	$E.val = num.val;$
$E \rightarrow P$	$E.val = P.val;$

En la DDS sólo hay atributos sintetizados, de modo que es S-atribuida y L-atribuida.

Para la entrada $[1, 3, [2, 1] (2)] (1)$, el árbol anotado es el siguiente:

