
Lenguajes de Programación y Procesadores de Lenguajes

(1º parcial)

9 de noviembre de 2022

1. (2,25 ptos.) Dada la siguiente gramática:

$$\begin{array}{ll} A \rightarrow B \ x \ ; \ ; & B \rightarrow C \ D \ * \\ C \rightarrow z \ D \ * \mid \epsilon & D \rightarrow y \ B \mid x \mid \epsilon \end{array}$$

- a) (0,75 ptos.) Calculad los **siguientes** de todos sus no-terminales.
b) (0,75 ptos.) Construid la tabla de análisis LL(1). ¿Es una gramática LL(1)? Justificad la respuesta.
c) (0,75 ptos.) A partir de la tabla anterior, obtened la traza LL(1) para la cadena:
 $z \ * \ x \ * \ x \ ;$

2. (0,75 ptos.) Dada la siguiente gramática,

$$A \rightarrow x \ y \ A \ B \mid x \ y \ ; \qquad B \rightarrow B \ y \mid B \ z \mid x$$

¿Porque no es LL(1)? Utilizando las transformaciones vistas en clase, reescribid la gramática eliminando los posibles problemas detectados.

3. (3,5 ptos.) Dada la siguiente gramática:

$$S \rightarrow b \ A \ B \mid \epsilon \qquad A \rightarrow S \ c \mid a \qquad B \rightarrow a \mid \epsilon$$

- a) (1,5 ptos.) Construid la colección canónica de conjuntos de ítems LR(0).
b) (1 pto.) A partir de dicha colección, construid la tabla de análisis SLR(1). ¿Es una gramática SLR(1)? Justificad brevemente la respuesta.
c) (1 pto.) Empleando la Tabla de Análisis SLR(1), proporcionad la traza de análisis sintáctico ascendente para la cadena: $b \ c \ a$
4. (1,5 ptos.) Construid un ETDS para la siguiente gramática. Donde la instrucción **find** (id, L) debe devolver, en el atributo **S.sol**, la posición de la primera ocurrencia del identificador *id* en la lista ordenada de identificadores *L* (o 0, si el identificador no se encuentra en la lista).

$$S \rightarrow \textbf{find} \ (\ id \ , \ L \) \qquad L \rightarrow id \ R \qquad R \rightarrow \ , \ id \ R \mid \epsilon$$

Para la solución del ejercicio se puede utilizar una función auxiliar **head(1)** que extrae y proporciona el elemento cabecera de la lista *L*.

5. (2 ptos.) Cuestiones teóricas (contestad brevemente):

- a) (0,5 ptos.) Nombra los módulos de un compilador.
- b) (0,5 ptos.) Escribid una expresión regular que reconozca números reales y acepte los siguientes lexemas: `-0.25`, `.4`, `+25.0`.
- c) (0,5 ptos.) Proporcionad las acciones de comprobación de tipos para las siguientes producciones de un lenguaje similar a C. Suponed que no hay coerción de tipos y que solo se permite la suma de enteros y reales. La expresión de asignación tomará el valor asignado a la variable del lado izquierdo de la asignación.

`E -> id = E`

`E -> E + E`

- d) (0,5 ptos.) Indicad cuál de las siguientes afirmaciones es cierta:
 - 1) En un estado del autómata reconocedor de prefijos viables de una gramática SLR(1) no puede haber más de un ítem/elemento que indique reducción.
 - 2) En la tabla de análisis LL(1) no puede haber una fila sin ninguna acción distinta de **error**.
 - 3) Toda gramática LL(1) es SLR(1).
 - 4) Una gramática es L-atribuida si todos los atributos empleados en todas las acciones semánticas son sintetizados.

1.-

a)

$SIG(A) = \{ \$ \}$

$SIG(B) = \{ x, * \}$

$SIG(C) = \{ y, x, * \}$

$SIG(D) = \{ * \}$

b)

	x	y	z	;	*	\$
A	Bx,, 1	Bx,, 1	Bx,, 1	;, 2	Bx,, 1	
B	CD*, 3	CD*, 3	CD*, 3		CD*, 3	
C	ϵ , 5	ϵ , 5	z D *, 4		ϵ , 5	
D	x, 7	yB, 6			ϵ , 8	

c)

$(A \$, z*x*x; \$,) \mid - (Bx; \$, z*x*x; \$, 1) \mid - (CD*x; \$, z*x*x; \$, 1-3) \mid - (zD*D*x; \$, z*x*x; \$, 1-3-4)$
 $\mid - (D*D*x; \$, *x*x; \$, 1-3-4) \mid - (*D*x; \$, *x*x; \$, 1-3-4-8) \mid - (D*x; \$, x*x; \$, 1-3-4-8) \mid -$
 $(x*x; \$, x*x; \$, 1-3-4-8-7) \mid - (*x; \$, *x; \$, 1-3-4-8-7) \mid - (x; \$, x; \$, 1-3-4-8-7) \mid - (; \$, \$, 1-3-4-8-7) \mid - (\$, \$, 1-3-4-8-7) \mid - \text{ACEPTAR}$

2.-

No es LL(1) porque hay dos producciones del no-terminal A que comienzan por el mismo prefijo y porque es recursiva a izquierdas ($B \rightarrow By$)

$A \rightarrow x y A'$

$A' \rightarrow A B \mid ;$

$B \rightarrow x B'$

$B' \rightarrow y B' \mid z B' \mid \epsilon$

3. (3,5 ptos.)

a)

I_0	$\begin{array}{l} S' \rightarrow \cdot S \\ S \rightarrow \cdot b A B \\ S \rightarrow \cdot \end{array}$	I_1	$\begin{array}{l} S' \rightarrow S \cdot \end{array}$	I_2	$\begin{array}{l} S \rightarrow b \cdot A B \\ A \rightarrow \cdot S c \\ A \rightarrow \cdot a \\ S \rightarrow \cdot b A B \\ S \rightarrow \cdot \end{array}$	I_3	$\begin{array}{l} S \rightarrow b A \cdot B \\ B \rightarrow \cdot a \\ B \rightarrow \cdot \end{array}$
I_4	$\begin{array}{l} A \rightarrow S \cdot c \end{array}$	I_5	$\begin{array}{l} A \rightarrow a \cdot \end{array}$	I_6	$\begin{array}{l} S \rightarrow b A B \cdot \end{array}$	I_7	$\begin{array}{l} B \rightarrow a \cdot \end{array}$
I_8	$\begin{array}{l} A \rightarrow S c \cdot \end{array}$						

$$\begin{array}{ccccc}
 I_0 \xrightarrow{S} I_1 & I_0 \xrightarrow{b} I_2 & I_2 \xrightarrow{A} I_3 & I_2 \xrightarrow{S} I_4 & I_2 \xrightarrow{a} I_5 \\
 I_2 \xrightarrow{b} I_2 & I_3 \xrightarrow{B} I_6 & I_3 \xrightarrow{a} I_7 & I_4 \xrightarrow{c} I_8 &
 \end{array}$$

b) Tabla de Análisis SLR(1)

	ACC				SUCC		
	a	b	c	\$	S	A	B
0		D-2	R-2	R-2	1		
1				AC			
2	D-5	D-2	R-2	R-2	4	3	
3	D-7		R-6	R-6			6
4			D-8				
5	R-4		R-4	R-4			
6			R-1	R-1			
7			R-5	R-5			
8	R-3		R-3	R-3			

$$\text{SIG}(S') = \{ \$ \}; \quad \text{SIG}(S) = \text{SIG}(B) = \{ \$, c \}; \quad \text{SIG}(A) = \{ \$, c, a \};$$

La gramática es SLR(1) ya que no tiene entradas múltiples en su tabla de análisis.

c) La traza para la cadena b c a

0		b c a \$	—
0 b 2		c a \$	2
0 b 2 S 4		c a \$	—
0 b 2 S 4 c 8		a \$	3
0 b 2 A 3		a \$	—
0 b 2 A 3 a 7		\$	5
0 b 2 A 3 B 6		\$	1
0 S 1		\$	Ac

4. (1,5 ptos.)

Utilizando atributos heredados

S \Rightarrow find (id , L	L.id = id.n; S.sol = L.sol;
L \Rightarrow id R	R.id = L.id; R.pos = 2; if (L.id == id.n) L.sol = 1; else L.sol = R.sol;
R \Rightarrow , id R	R'.id = R.id; R'.pos = R.pos + 1; if (R.id == id.n) R.sol = R.pos; else R.sol = R'.sol;
$\Rightarrow \epsilon$	R.sol = 0;

Utilizando atributos sintetizados y la función head.

S \Rightarrow find (id , L)	aux = head(L.l); i = 1; ok = true; while ((aux \neq nil) && ok) if (id.n == aux) { S.sol = i; ok = false; } else { aux = head(L.l); i++; } if (aux == nil) S.sol = 0;
L \Rightarrow id R	L.l = id.n x R.l;
R \Rightarrow , id R	R.l = id.n x R'.l;
$\Rightarrow \epsilon$	R.l = nil

5.

a)

Analizador léxico, analizador sintáctico, analizador semántico, generador de código intermedio, optimizador de código intermedio, generador de código, tabla de símbolos, manejador de errores.

b)

$(+|-|\epsilon)(0-9)^* \cdot (0-9)^+$

c)

$E \rightarrow id = E_1$	<code>id.tipo = BuscaTipo(id.nom) ; if (id.tipo != E₁.tipo) { yyerror ("Error de tipo en la asignación"); E.tipo = terror ; } else E.tipo = id.tipo ;</code>
$E \rightarrow E_1 + E_2$	<code>if (E₁.tipo != E₂.tipo (E₁.tipo != entero && E₁.tipo != real)) { yyerror("Error de tipos en la suma"); E.tipo = terror} else E.tipo = E₁.tipo ;</code>

d)

1) En un estado del autómata reconocedor de prefijos viables de una gramática SLR(1) no puede haber más de un ítem/elemento que indique reducción.

2) En la tabla de análisis LL(1) no puede haber una fila sin ninguna acción distinta de error.

3) Toda gramática LL(1) es SLR(1).

4) Una gramática es L-atribuida si todos los atributos empleados en todas las acciones semánticas son sintetizados.