Lenguajes de Programación y Procesadores de Lenguajes

(1º parcial) **10 de noviembre de 2017**

1. **(2 ptos.)** Dada la siguiente gramática:

$$
\begin{array}{lll@{\qquad}lll@{\qquad}lll}
A & \rightarrow & B\ C & C & \rightarrow & z\ D & E & \rightarrow & y \mid \epsilon \\
B & \rightarrow & w\ D \mid E\ D & D & \rightarrow & x\ D \mid \epsilon & & &
\end{array}
$$

   a) **(1,25 ptos.)** Construid la tabla de análisis LL(1).
   b) **(0,75 ptos.)** Proporcionad la traza LL(1) para la cadena: `x z x`

---

a)

SIGUIENTES(A)={\$}; SIGUIENTES(B)={z};
SIGUIENTES(C)={\$}; SIGUIENTES(D)={\$, z};
SIGUIENTES(E)={x, z};

(PRIMEROS(B C · SIGUIENTES(A))={w, x, y, z})
(PRIMEROS(w D · SIGUIENTES(B))={w})
(PRIMEROS(E D · SIGUIENTES(B))={x, y, z})
(PRIMEROS(z D · SIGUIENTES(C))={z})
(PRIMEROS(x D · SIGUIENTES(D))={x})
(PRIMEROS(SIGUIENTES(D))={z, \$})
(PRIMEROS(y · SIGUIENTES(E))={y})
(PRIMEROS(SIGUIENTES(E))={x, z})

	w	**x**	**y**	**z**	**\$**
A	r1	r1	r1	r1	
B	r2	r3	r3	r3	
C				r4	
D		r5		r6	r6
E		r8	r7	r8	

b)

A \$	x z x \$	–
B C \$	x z z \$	1
E D C \$	x z x \$	1 3
D C \$	x z x \$	1 3 8
x D C \$	x z x \$	1 3 8 5
D C \$	z x \$	1 3 8 5
C \$	z x \$	1 3 8 5 6
z D \$	z x \$	1 3 8 5 6 4
D \$	x \$	1 3 8 5 6 4
x D \$	x \$	1 3 8 5 6 4 5
D \$	\$	1 3 8 5 6 4 5
\$	\$	1 3 8 5 6 4 5 6

2. **(1 pto.)** Dada la siguiente gramática:

$$P \;\rightarrow\; P\,x \;|\; y\,P\,y \;|\; y\,Q \qquad\qquad Q \;\rightarrow\; x \;|\; z$$

Obtened una gramática LL(1) equivalente y comprobad que la nueva gramática es LL(1).

La gramática tiene recursividad a izquierdas y problemas de factorización. Podemos proceder de dos maneras equivalentes:

a) Factorizando primero y eliminando después la recursividad a izquierdas.

$$\begin{array}{lclcl}
P\rightarrow P\ x & & P\rightarrow P\ x & & P\rightarrow y\ P^1\ P^2 \\
P\rightarrow y\ P\ y & & P\rightarrow y\ P^1 & & P^2\rightarrow x\ P^2 \\
P\rightarrow y\ Q & & P^1\rightarrow P\ y & & P^2\rightarrow \epsilon \\
Q\rightarrow x & \Longrightarrow & P^1\rightarrow Q & \Longrightarrow & P^1\rightarrow P\ y \\
Q\rightarrow z & & Q\rightarrow x & & P^1\rightarrow Q \\
 & & Q\rightarrow z & & Q\rightarrow x \\
 & & & & Q\rightarrow z
\end{array}$$

$\textsc{siguientes}(P) = \textsc{siguientes}(P^2) = \{y, \$\}$;
$\textsc{siguientes}(P^1) = \textsc{siguientes}(Q) = \{x, y, \$\}$;

$\textsc{primeros}(P\ y \cdot \textsc{siguientes}(P^1)) = \{y\} \cap \textsc{primeros}(Q \cdot \textsc{siguientes}(P^1)) = \{x, y\} = \emptyset$
$\textsc{primeros}(x\ P^2 \cdot \textsc{siguientes}(P^2)) = \{x\} \cap \textsc{primeros}(\textsc{siguientes}(P^2)) = \{y, \$\} = \emptyset$
$\textsc{primeros}(x \cdot \textsc{siguientes}(Q)) = \{x\} \cap \textsc{primeros}(z \cdot \textsc{siguientes}(Q)) = \{z\} = \emptyset$

La gramática resultante es LL(1)

b) Eliminando la recursividad primero y factorizando después.

$$\begin{array}{lclcl}
P\rightarrow P\ x & & P\rightarrow y\ P\ y\ P^a & & P\rightarrow y\ P^b \\
P\rightarrow y\ P\ y & & P\rightarrow y\ Q\ P^a & & P^b\rightarrow P\ y\ P^a \\
P\rightarrow y\ Q & & P^a\rightarrow x\ P^a & & P^b\rightarrow Q\ P^a \\
Q\rightarrow x & \Longrightarrow & P^a\rightarrow \epsilon & \Longrightarrow & P^a\rightarrow x\ P^a \\
Q\rightarrow z & & Q\rightarrow x & & P^a\rightarrow \epsilon \\
 & & Q\rightarrow z & & Q\rightarrow x \\
 & & & & Q\rightarrow z
\end{array}$$

$\textsc{siguientes}(P) = \textsc{siguientes}(P^a) = \textsc{siguientes}(P^b) = \{y, \$\}$;
$\textsc{siguientes}(Q) = \{x, y, \$\}$;

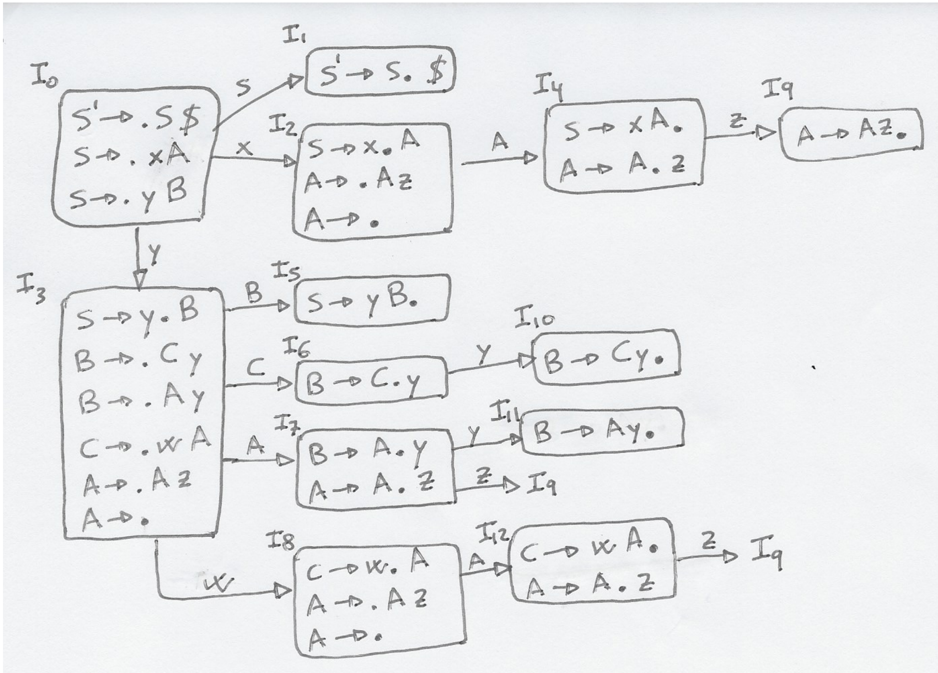
$\textsc{primeros}(P\ y\ P^a \cdot \textsc{siguientes}(P^b)) = \{y\} \cap \textsc{primeros}(Q\ P^a \cdot \textsc{siguientes}(P^b)) = \{x, z\} = \emptyset$
$\textsc{primeros}(x\ P^a \cdot \textsc{siguientes}(P^a)) = \{x\} \cap \textsc{primeros}(\textsc{siguientes}(P^a)) = \{y, \$\} = \emptyset$
$\textsc{primeros}(x \cdot \textsc{siguientes}(Q)) = \{x\} \cap \textsc{primeros}(z \cdot \textsc{siguientes}(Q)) = \{z\} = \emptyset$

La gramática resultante es LL(1)

1.- Dada la siguiente gramática:

S → x A | y B
A → A z | ε
B→ C y | A y
C→ w A

a) (1,25 pto) Construye la colección canónica de conjuntos de elementos LR(0).
b) (1,25 pto) Construye la tabla de análisis SLR(1)
c) (1 pto) Realiza la traza de análisis SLR(1) para la cadena “yzzy”



SIG(S)= { $ }
SIG(A)={ $, z, y}
SIG(B)={$}
SIG(C)={y}

	x	y	z	w	$	S	A	B	C
0	d-2	d-3				1			
1					ACEPTAR				
2		r-4	r-4		r-4		4		
3		r-4	r-4	d-8	r-4		7	5	6
4			d-9		r-1				
5					r-2				
6		d-10							
7		d-11	d-9						
8		r-4	r-4		r-4		12		
9		r-3	r-3		r-3				
10					r-5				
11					r-6				
12		r-7	d-9						

(0 , yzzy$,) |- (0y3 , zzy$,) |- (0y3A7 , zzy$, 4) |- (0y3A7z9 , zy$, 4) |- (0y3A7 , zy$, 4-3) |- (0y3A7z9 , y$, 4-3) |- (0y3A7 , y$, 4-3-3) |- (0y3A7y11 , $, 4-3-3) |- (0y3B5 , $, 4-3-3-6) |- (0S1 , $, 4-3-3-6-2) |- ACEPTAR

2.- Dada la siguiente gramática para definir objetos en 2 y 3 dimensiones, donde un punto (*P*) está representado por números (*num*) separados por comas:

S → **space** E **begin** LO **end**
E → **2D** | **3D**
LO → **line (** LP **)** LO | **square (** LP **)** LO | ε
LP → P | LP **:** P
P → **num** | P **,** **num**

<u>Ejemplo:</u>

```
space 2D
  begin
    line (5, 2 : 9, 3)
  end
```

Escribe un ETDS que realice las siguientes comprobaciones semánticas:

a) (0.75) Si el espacio es *2D*, todos los puntos deben estar definidos por 2 números, si es *3D* por 3 números.
b) (0.75) Los objetos *line* están definidos por 2 puntos y los objetos *square* por 3 puntos.

S → **space** E **begin** LO **end**	*{ LO.spc = E.spc ; }*
E → **2D**	*{ E.spc = 2 ; }*
| **3D**	*{ E.spc = 3 ; }*
LO → **line (** LP **)** LO_1	*{ LP.spc = LO.spc ; LO_1.spc = LO.spc ; }* *{ if (LP.pto != 2) yyerror("Linea mal definida"); }*
| **square (** LP **)** LO_1	*{ LP.spc = LO.spc ; LO_1.spc = LO.spc ; }* *{ if (LP.pto != 2) yyerror("Cuadrado mal definido"); }*
| ε	
LP → P	*{ LP.pto =1 ; if (P.num != LP.spc) yyerror("Punto mal definido"); }*
| LP_1 **:** P	*{ LP_1.spc = LP.spc ;}* *{ LP.pto = LP_1.pto + 1 ; if (P.num != LP.spc) yyerror("Punto mal definido"); }*
P → **num**	*{ P.num = 1 ; }*
| P_1 **,** **num**	*{ P.num = P_1.num + 1 ; }*

Apartado a en verde y apartado b en azul

3. **(2 ptos.)** Cuestiones teóricas (contestad brevemente):

a) `(0,5 ptos.)` Describid las principales funciones de un Analizador Léxico e indicad cómo se puede tratar el problema de la detección de las palabras reservadas.

b) `(0,5 ptos.)` Dada una derivación a izquierdas, $S \Rightarrow wA\alpha$, siendo a el símbolo actual de la cadena de entrada, $A\alpha$ está en la pila y A en su cima: i) justificad que w debe ser una cadena de símbolos terminales ($w \in T^*$); y ii) ¿qué condición se debe cumplir para que se derive el no-terminal A (p.ej. con la regla $A \rightarrow \beta$) ?

c) `(0,5 ptos.)` Dada la Colección Canónica de Conjuntos de ítems LR(0) del ejercicio **3.a**, `''y z''`, ¿puede ser un prefijo viable? Si `''y A''` es un prefijo viable, ¿cuales son sus ítems válidos? Justificad ambas respuestas.

d) `(0,5` **ptos.)** Considerando que se ha completado la fase de declaración de los objetos, diseñad un ETDS para la comprobación semántica de tipos en el par de reglas:

`E → * id` y `E → id`

---

a) Las principales funciones de un Analizador Léxico (AL) son la detección de los símbolos del lenguaje y la realización de las acciones semánticas asociadas con la detección de un símbolo, así como la emisión de las componentes léxicas ("tokens") detectadas.

Además el AL también realiza otras funciones: 1) Tratamiento de errores léxicos; 2) eliminación de cadenas inútiles (comentarios, tabuladores, saltos de línea, etc.); 3) lectura eficiente del fichero de entrada; 4) relación de los mensajes de error con las líneas del programa fuente; y 4) reconocimiento y ejecución de las directivas de compilación.

Las palabras reservadas pueden tratarse, o bien con expresiones regulares independientes, o bien, como identificadores especiales (tabla de palabras reservadas).

b) Si es una secuencia de derivación a izquierdas, significa que derivamos siempre el no-terminal de la izquierda. Si A está en la cima de la pila, implica que debe derivarse por tanto todos los símbolos a su izquierda solo pueden ser terminales.

$$a \in \text{PRIMEROS}(\alpha \cdot \text{SIGUIENTES}(\text{A}))$$

c) `''y z''`, no es un prefijo viable ya que no existe un camino desde el origen en la Colección Canónica de Conjuntos de ítems LR(0) del ejercicio **3.a** con ese prefijo.

Para el prefijo viable `''y A''`, sus ítems válidos son:

$$\{\ [\text{B} \rightarrow \text{A} \cdot \text{y}],\ [\text{A} \rightarrow \text{A} \cdot \text{z}]\ \}$$

d)

E ⇒ id	<u>Si</u> ¬[ObtenerTDS(id.nom,E.t)] { E.t=terror; MenError(.); }
E ⇒ * id	<u>Si</u> ¬[ObtenerTDS(id.nom,id.t) ∧ id.t=tpuntero(E.t)] { E.t=terror; MenError(.); }