



I.T. Informática Gestión – Teoría de Autómatas y Lenguajes

Examen convocatoria ordinaria. Parte Practica. 17 de junio de 2008.

Apellidos _____

DNI y Firma _____

Nombre _____ Grupo _____

--	--	--	--	--

Resolver los problemas en una página diferente cada uno de ellos.
Poner nombre y apellidos en todas las páginas.
Las hojas del enunciado también deben entregarse.

Se valorará la presentación y la claridad en la exposición.
No se calificarán las respuestas escritas a lápiz.
La nota de cada ejercicio aparece en su enunciado.
Resolver el ejercicio 3 en esta misma hoja.

1. (1,25 pto) Dado el siguiente fichero *y.output* obtenido por Yacc:

<pre> 0 \$accept : S \$end 1 S : a A 2 c 3 A : B a 4 B : a C 5 C : S a state 0 \$accept: . S \$end (0) a shift 1 c shift 2 . error S goto 3 state 1 S : a . A (1) a shift 4 . error A goto 5 B goto 6 state 2 S : c . (2) . reduce 2 </pre>	<pre> state 3 \$accept : S . \$end (0) \$end accept state 4 B : a . C (4) a shift 1 c shift 2 . error S goto 7 C goto 8 state 5 S : a A . (1) . reduce 1 state 6 A : B . a (3) a shift 9 . error state 7 C : S . a (5) a shift 10 . error </pre>	<pre> state 8 B : a C . (4) . reduce 4 state 9 A : B a . (3) . reduce 3 state 10 C : S a . (5) . reduce 5 4 terminals, 5 nonterminals 6 grammar rules, 11 states </pre>
---	--	--

Se pide:

- 1) Obtener la tabla de análisis por desplazamiento reducción.
- 2) Simular el análisis para la cadena *aacaa*, especificando los estados que se van almacenando en la pila y mostrar la derivación y el árbol de análisis obtenidos.

2. (0,75 pts) Calcúlese la TASP para la siguiente gramática, especificando primeros y siguientes, en los espacios reservados para ello (completando las columnas que falten).

$$G \left\{ \begin{array}{l} S \rightarrow a A b d \\ A \rightarrow B A \\ \quad | \varepsilon \\ B \rightarrow f d C g \\ \quad | h e \\ C \rightarrow d C \\ \quad | \varepsilon \end{array} \right.$$

Primeros	S	A	B	C

Siguientes	S	A	B	C

TASP	a	b	d	e	f
S					
A					
B					
C					

3. (1,5 pts) Dado un reconocedor finito determinista escrito en forma de tabla, con un cierto formato, se pide construir un programa (utilizando LEX y YACC) que determine si la entrada es correcta e informe del número de estados finales del autómata.

La tabla del RFD contendrá una línea de cabecera (con los símbolos del alfabeto de entrada separados por el carácter punto_y_coma “;”) y una línea por cada estado.

Cada línea de estado contendrá en primer lugar el nombre o número de estado, entre paréntesis si es final y precedido por los caracteres “->” si es el inicial, y a continuación los estados siguientes para cada entrada separados por el carácter punto_y_coma “;”. Para nombra estados se usaran cadenas de letras minúsculas y dígitos.

Por ejemplo, dada la entrada:

```

a ; c
->p ; q ; p
(r) ; q ; r
q ; p ; r
    
```

La salida debería ser:

```

AFD correcto
Tiene 1 estado final
    
```