PROCESADORES DE LENGUAJE. *Septiembre-2005*
4º Curso Ingeniería Informática Tiempo: 2:00 horas
No se permiten ni libros ni apuntes.

1. (2.0 p.) Demostrar que una gramática con producciones de la forma $A \rightarrow \alpha_1 \mid \ldots \mid \alpha_n \mid \varepsilon$ con $\alpha_i \in (VT \cup VN)$ de forma que A es un símbolo que puede generar cadenas no vacías nunca puede ser LR(0). ¿Podría ser LR(0) si sólo A puede generar la cadena vacía?

2. (1.0 p.) Dada una gramática de atributos heredados y sintetizados. ¿Se podrían evaluar estos atributos a la vez usando el método de análisis sintáctico LR?

3. (2.0 p) Dada la siguiente gramática modifícala para permitir la operación % ("modulo de un numero"). Se pide que este operador sea asociativo a derechas y tenga mayor prioridad que la suma y la división.

   E → E **+** T | T

   T → T **/** F | F

   F → **(** E **)** | **num**

4. (2.5 p) Dada la siguiente gramática de precedencia de operadores que permite generar paréntesis anidados S → **(** S **)** | **id.** a). Obtén la tabla de precedencia de operadores. b) Analiza la cadena ((id)) indicando el contenido de la pila y de la entrada. c) Implementa un mecanismo de recuperación de errores a nivel de frase.

5. (2.5 p) Dada la sentencia **exit** de los lenguajes de programación que permite de forma inmediata terminar un bucle. Se pide: a). Dibujar el diagrama de flujo de dicha construcción; b) Dibujar el árbol que construirías para traducir a código intermedio dicha construcción.; c) Implementar en pseudocódigo la función genera_código para traducir esta construcción.

   Stmt_exit → **exit** When_Cond **;**

   When_Cond → **when** E | ε