

Procesadores de Lenguaje. Curso 2000-2001. Septiembre de 2001. Tiempo: 2 hora 15 min.

1. (2 puntos) Contesta a las siguientes preguntas razonando tu respuesta:

- ¿Puede una gramática ambigua ser LL(1)?
- Demuestra que una gramática recursiva a izquierdas no puede ser LL(1).
- ¿Puede haber una gramática SLR(1) que no sea LALR(1)?

• Dada la producción

$S \rightarrow \text{if } E \text{ then } S \mid \text{if } E \text{ then } S \text{ else } S$

comenta el tipo de conflicto que aparece y que acción se debe tomar para que la parte del **else** esté asociada al **if** más cercano.

2. (2 puntos) Diseña un ETDS que traduzca consultas sencillas en SQL escritas en inglés a castellano. La gramática que genera este tipo de consultas en SQL es:

$\text{Consult} \rightarrow \text{ClausulaSelect ClausulaFrom ClausulaWhere}$

$\text{ClausulaSelect} \rightarrow \text{select id } (, \text{id})^+$

$\text{ClausulaFrom} \rightarrow \text{from id } (, \text{id})^+$

$\text{ClausulaWhere} \rightarrow \text{where } (\text{id} = \text{id} \mid \text{id} > \text{id})$

De manera que ante la entrada

select Nombre, DNI

from Estudiantes

where Edad>20

se convierta en:

seleccionar Nombre, DNI

desde Estudiantes

donde Edad>20

3. (2 puntos) Dada la siguiente gramática para generar expresiones booleanas

$E \rightarrow E \text{ and } E \mid E \text{ or } E \mid \text{not } E \mid \text{id}$

Demuestra que la gramática es ambigua usando como entrada **id and id or id**. Transforma la gramática de manera que el operador **not** tenga mayor precedencia que el **and** y éste mayor precedencia que el **or**. Se desea también que el operador **not** sea asociativo por la derecha y los operadores **and** y **or** por la izquierda.

4. (2 puntos) Considera la siguiente gramática para generar árboles binarios de enteros:

$\text{btree} \rightarrow (\text{number btree btree}) \mid \text{null}$

Diseña una gramática de atributos para comprobar si un árbol binario está ordenado, es decir, que en cada nodo el valor de los enteros en el subárbol de la izquierda sean menores o iguales que el valor en el nodo y que los valores del subárbol de la derecha sean mayores que el valor del nodo. Por ejemplo, el árbol (2 (1 nil nil) (3 nil nil)) está ordenado, pero no el árbol (1 (2 nil nil) (3 nil nil)).

5. (2 puntos) Considera la siguiente gramática simplificada la generación de sentencias:
`stmt-sequence` \rightarrow `stmt-sequence ; stmt | stmt`
`stmt` \rightarrow `s`

Se pide:

- Construye el AFD de elementos LR(0).
- Construye la tabla de análisis sintáctico SLR(1).
- Muestra el contenido de la pila de análisis y de acciones para la entrada `s;s;s`.
- ¿Es esta gramática LR(0)? Si no lo es, describe el tipo de conflicto. Y si lo es, construye la tabla de análisis sintáctico LR(0) y describe de que modo difieren los dos tipos de análisis.