

Examen de Procesadores de Lenguaje. Junio 2008. Teoría. Curso 2007-2008

Duración: 2:30 horas

Segundo Parcial: 2, 3, 4

Final: 1, 2, 4

Pregunta 1 (2.5 pts) Dadas la siguiente gramática indica si es ambigua. Justifica la respuesta.

```
S → if C then S | if C then S else S | id
C → 0 | 1
```

Modifícala para que deje de serlo introduciendo la palabra reservada **endif** en la sintáxis. Además debe de ser LL(1). Construye la tabla de análisis sintáctico descendente. Indica el conjunto de predicción y sincronización para cada símbolo. Haz la traza para la cadena `if then a endif`. Indica el mensaje de error correspondiente y comenta las acciones a realizar para sincronizarnos usando el método de recuperación en modo de pánico.

Pregunta 2 (3.5 pts) Dada la siguiente gramática justifica si es SLR(1). Construye el autómata de ítems LR(0). En el caso de que aparezcan conflictos resuélvelos usando las prioridades habituales. El operador `+` y `*` tienen asociatividad a izquierdas y la potencia a derechas. El operador suma tiene menor prioridad que el producto y la potencia. El operador producto tiene menor prioridad que la potencia. Construye la tabla de análisis sintáctico. Introduce la producción de error $E \rightarrow E \chi E$. Construye de nuevo el autómata y realiza la recuperación mediante producciones de error de la cadena `n*n n n`.

```
E → E * E | E + E | E ^ E | n
```

Pregunta 3 (2.5 pts) Dada la siguiente gramática que permite generar declaraciones de funciones. Se pide: a) diseñar una gramática de atributos que permita obtener para cada parámetro su nombre y tipo de forma que en el axioma podamos tener una lista con el nombre y tipo de cada parámetro. b) Indica de qué tipo de atributos se trata. c) Implementa en pseudocódigo la función que los evalúa. Indica la forma del árbol que has creado. d) Haz la traza para la entrada `function suma (integer a, integer b) : integer;`

```
DeclsFunc → Cabezera Cuerpo
Cabezera → function id ( Param ) : Type ;
Cuerpo → DecliVar begin StmtList end id ;
Param → Type id ParamList | ε
ParamList → , Type id ParamList | ε
Type → char | integer | boolean
```

Pregunta 4 (4.0 pts) Dada la sentencia de selección del lenguaje BASIC

```
S → selectcase id of C* endcase
C → case num : S*
```

Se pide: a) indica el diagrama de flujo; b) indica la forma del árbol que construirías; c) implementa la función genera código; d) haz la traza para la cadena siguiente y e) optimiza el código generado en su caso.

```
a = 2 - 1 ;
selectcase a of
  case 1 :
    b=a+2 ;
  case 2 :
    b=a+3 ;
endcase ;
```