

WUOLAH



Adrian_Lopez7
www.wuolah.com/student/Adrian_Lopez7



ExamenPL-Junio2016.pdf

Exámenes



3º Procesadores de Lenguajes



Grado en Ingeniería Informática



**Escuela Politécnica Superior de Córdoba
UCO - Universidad de Córdoba**

 escuela
de negocios
CÁMARA DE SEVILLA

MÁSTER EN DIRECCIÓN Y GESTIÓN DE RECURSOS HUMANOS

www.mastersevilla.com

Inscríbete



BECAS



PROCESADORES DE LENGUAJES

Ingeniería Informática
Especialidad de Computación
Tercer curso, segundo cuatrimestre



Departamento de Informática y Análisis Numérico
Escuela Politécnica Superior de Córdoba
Universidad de Córdoba
Curso académico 2015 - 2016
Examen de la convocatoria de junio
17 de junio de 2016

Teoría

1. Generación del código intermedio.
 - Descripción y características1 punto
2. Bootstrapping
 - Descripción
 - Ejemplo1 punto
3. Criterios “generales” para el tratamiento de errores léxicos1 punto

Problemas

4. Componentes léxicos
 - La siguiente expresión regular denota números naturales: $c + n(c + n)^*$ donde
 - c es el número 0
 - n es un dígito del 1 al 9
 - Utiliza el Algoritmo de Thompson para construir un AFN equivalente a la expresión regular.
 - Utiliza el Algoritmo de Construcción de Subconjuntos para obtener un AFD equivalente al AFN obtenido en el apartado “a”.
 - Minimiza, si es posible, el AFD obtenido en el apartado anterior.
 - Comprueba si el último autómata obtenido reconoce a la siguiente secuencia:
 $n \ n \ c \ c$ 2 puntos

5. Análisis descendente predictivo

- Considérese la siguiente gramática de contexto libre
 $P = \{$
 - 1) $S \rightarrow S D$
 - 2) $S \rightarrow D$
 - 3) $D \rightarrow T \text{ identificador } (L) ;$
 - 4) $T \rightarrow \text{entero}$
 - 5) $T \rightarrow \text{real}$
 - 6) $L \rightarrow \varepsilon$
 - 7) $L \rightarrow A$
 - 8) $A \rightarrow A , \text{ identificador}$
 - 9) $A \rightarrow \text{identificador}$ $\}$
- Esta gramática permite generar declaraciones de prototipos de funciones como, por ejemplo:
real media (entero a, entero b);
donde “media”, “a” y “b” son identificadores



Ingeniería Informática

Procesadores de Lenguajes

- a) Elimina la recursividad por la izquierda y factoriza la gramática por la izquierda.
- b) A partir de la gramática obtenida en el apartado "a":
 - o Construye los conjuntos "primero" y "siguiente" de los símbolos no terminales.
 - o Construye la tabla de análisis descendente predictivo.
 - o Utiliza el método recuperación de errores de "nivel de frase" para completar la tabla predictiva.
 - o Utiliza la tabla predictiva para realizar un análisis descendente no recursivo de la siguiente declaración errónea:
real (entero entero a, entero b ;

2 puntos

6. Análisis sintáctico ascendente SLR

- Considérese la siguiente gramática de contexto libre
 $P = \{$
 - 1) $S \rightarrow S D$
 - 2) $S \rightarrow D$
 - 3) $D \rightarrow \text{Var } L : T ;$
 - 4) $L \rightarrow L , \text{identificador}$
 - 5) $L \rightarrow \text{identificador}$
 - 6) $T \rightarrow \text{Integer}$
 - 7) $T \rightarrow \text{Real}$ $\}$
- Esta gramática puede generar declaraciones de variables en el lenguaje de programación Pascal, como, por ejemplo:
 $\text{Var } a, b : \text{Integer} ;$
 $\text{Var } x, y, z : \text{Real} ;$
 donde "a", "b", "x", "y" y "z" son identificadores.
 - a) Construye la colección canónica de LR(0)-elementos del análisis SLR
 - b) Dibuja el autómata que reconoce los prefijos viables.
 - c) Construye los conjuntos "primero" y "siguiente" de los símbolos no terminales.
 - d) Construye la tabla de análisis sintáctico SLR: partes acción e ir_a
 - e) Utiliza el método recuperación de errores de "nivel de frase" para completar la tabla de análisis SLR
 - f) Utiliza la tabla SLR para analizar la siguiente declaración errónea:
 $a , b : \text{Integer Integer} ;$

2 puntos

7. Diseño de una gramática de contexto libre

- Diseña una gramática que permita generar declaraciones de tipos de registros usando la sintaxis del lenguaje de programación Pascal
- Por ejemplo:
 $\text{Type TipoPersona} = \text{Record}$
 $\quad \text{Nombre, Apellidos} : \text{string} ;$
 $\quad \text{Direccion} : \text{array } [1..3] \text{ of string} ;$
 $\quad \text{Telefono} : \text{Integer} ;$
 $\text{End};$

1 punto