

WUOLAH



Adrian_Lopez7
www.wuolah.com/student/Adrian_Lopez7



ExamenPL-Junio2014.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 2013 - 2014



Examen de la convocatoria de junio
20 de junio de 2014

Teoría

1. Fases y componentes auxiliares del proceso de compilación

- Descripción
- Diagrama

2 puntos

2. Modo de pánico de recuperación de errores sintácticos

- Descripción general
- Implementación en el análisis sintáctico **descendente**
- Implementación en el análisis sintáctico **ascendente**

1 punto

Problemas

3. Componentes léxicos

- La siguiente expresión regular denota números:

$d d^* (p d d^* + \epsilon)$

donde d = dígito y p = punto (.)

- 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:

$d d p d$

2 puntos

4. Análisis descendente predictivo

- La siguiente gramática permite generar algunos **prototipos de funciones** el lenguaje de programación C:

$P = \{$

- $S \rightarrow S D$
- $S \rightarrow D$
- $D \rightarrow T \text{ id } (L) ;$
- $T \rightarrow T^*$
- $T \rightarrow \text{int}$
- $L \rightarrow L, I$



Ingeniería Informática

Procesadores de Lenguajes

7) $L \rightarrow I$
8) $I \rightarrow T \text{ id}$
}

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

2,5 puntos

?

5. Análisis sintáctico ascendente SLR

- Considera la siguiente gramática de contexto libre

$P = \{$
1) $S \rightarrow SD$
2) $S \rightarrow \epsilon$
3) $D \rightarrow \text{enum } \text{identificador } \{L\};$
4) $L \rightarrow \text{identificador}$
5) $L \rightarrow L, \text{identificador}$
 $\}$

Esta gramática puede generar algunas "enumeraciones" del lenguaje de programación C:

enum color {blanco, negro, amarillo};

donde color, blanco, negro y amarillo son identificadores.

- Construye la colección canónica de **LR(0)-elementos**
- Dibuja el **autómata** que reconoce los prefijos viables.
- Construye la tabla de análisis sintáctico **SLR**
- Utiliza el método recuperación de errores de "**nivel de frase**" para completar la tabla SLR.
- Utiliza la tabla SLR para analizar la siguiente declaración errónea:
enum enum { blanco amarillo , ;

2,5 puntos