

Práctica. Segunda Fase

Desarrollo de analizadores sintácticos para Tiny(0) y para Tiny(1)

En esta segunda parte debe realizarse el siguiente trabajo:

- 1) Desarrollo manual de un analizador sintáctico para **Tiny(0)**. Para ello, deberá entregarse:
 - Una memoria con las siguientes secciones:
 - Portada en la que aparezcan los nombres y apellidos de los integrantes del grupo, y el número de grupo.
 - Especificación sintáctica (gramática) para **Tiny(0)**. Dicha especificación deberá desarrollarse utilizando los patrones de escritura de gramáticas explicados en clase.
 - Acondicionamiento de la gramática para permitir la implementación de un analizador sintáctico descendente predictivo recursivo.
 - Directores de cada regla de la gramática acondicionada (para su determinación, se recomienda utilizar una herramienta de análisis de gramáticas, como, por ejemplo, Proletool).
 - Símbolos para el diagnóstico de errores para cada no terminal, junto con una justificación de la elección de dichos símbolos.
 - Una implementación manual, en Java, del analizador sintáctico descendente predictivo recursivo. Dicho analizador deberá funcionar con la implementación manual del analizador léxico para **Tiny(0)** desarrollado en la primer fase. Debe proporcionarse, además, un programa de prueba que acepte como argumento el archivo a procesar, y genere como salida, bien un mensaje legible del primer error (léxico o sintáctico) detectado, bien un mensaje "OK" cuando el programa analizado sea sintácticamente correcto.
- 2) Desarrollo de analizadores sintácticos descendentes y ascendentes para **Tiny(1)**. Para ello, deberá entregarse:
 - Una memoria con las siguientes secciones:
 - Portada en la que aparezcan los nombres y apellidos de los integrantes del grupo, y el número de grupo.
 - Especificación sintáctica (gramática) para **Tiny(1)**. Dicha especificación deberá desarrollarse utilizando los patrones de escritura de gramáticas explicados en clase.
 - Acondicionamiento de la gramática para permitir la implementación de un analizador sintáctico descendente predictivo recursivo.
 - Una implementación de un analizador sintáctico descendente predictivo desarrollada con JavaCC.
 - Una implementación de un analizador sintáctico ascendente LR desarrollada con CUP. Dicha implementación integrará el analizador léxico para **Tiny(1)** desarrollado con JFlex en la primera fase de la práctica.
 - Un programa principal que integre ambos analizadores. Dicho programa recibirá como argumento (i) el archivo a analizar; (ii) una opción *op* que indique el analizador sintáctico a aplicar (si *op* es *desc* el analizador a aplicar será el descendente; si es *asc* será el ascendente). El programa producirá como salida, bien un mensaje legible del primer error (léxico o sintáctico) detectado, bien un mensaje "OK" cuando el programa analizado sea sintácticamente correcto.

Fecha límite de entrega: **Domingo 18 de abril de 2021, a las 11:59 pm.**

Modo de entrega: A través del campus virtual, en un único .zip. Dicho archivo debe contener: (i) un documento PDF `memoria_tiny_0.pdf` con la memoria requerida en el punto 1) del trabajo a realizar, y otro documento PDF `memoria_tiny_1.pdf` con la memoria requerida en el punto 2); (ii) una carpeta `implementación_tiny0`, en el interior de la cuál debe incluirse toda la implementación requerida para **Tiny(0)**; (iii) una carpeta `implementación_tiny1`, en el interior de la cuál debe incluirse la implementación requerida para **Tiny(1)**; (iv) una carpeta `pruebas_tiny_0` con distintos programas de prueba que permitan probar la implementación para **Tiny(0)**; y (v) una carpeta `pruebas_tiny_1` con distintos programas de prueba que permitan probar la implementación para **Tiny(1)**. La entrega debe ser realizada solamente por un miembro del grupo.