

Práctica. Segunda Fase

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

En esta segunda parte debe realizarse el siguiente trabajo:

- 1) Desarrollo manual de un analizador sintáctico para **Tiny(0)**. Para ello, deberá entregarse:
 - Un apartado en la memoria con las siguientes secciones:
 - 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.
 - 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**. Para ello, deberá entregarse:
 - Un apartado en la memoria con las siguientes secciones:
 - Especificación sintáctica (gramática) para **Tiny**. 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 desarrollada con CUP. Dicha implementación integrará el analizador léxico para **Tiny** 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.

En la memoria que deberá entregarse con esta práctica habrá que incluir una portada en la que aparezcan los nombres y apellidos de los integrantes del grupo, y el número de grupo.

Fecha límite de entrega: **Viernes 8 de marzo de 2024, a las 23:5h.**

Modo de entrega: A través del campus virtual, en un único .zip. Dicho archivo debe contener: (i) un documento PDF `memoria.pdf` con la memoria; (ii) una carpeta `implementación_tiny0`, en el interior de la cuál debe incluirse toda la implementación requerida para **Tiny(0)**; (iii) una carpeta `implementación_tiny`, en el interior de la cuál debe incluirse la implementación requerida para **Tiny**; (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**. La entrega debe ser realizada solamente por un miembro del grupo.

Las implementaciones deberán, además, entregarse a través del juez DomJudge de la asignatura, siguiendo las indicaciones que se proporcionarán más adelante.