

## Práctica. Cuarta fase

### Finalización del procesador para Tiny

En esta última fase debe completarse el desarrollo de un procesador para **Tiny**. Para ello deberá realizarse el siguiente trabajo:

- 1) Una memoria con los siguientes aspectos:
  - Especificación del procesamiento de vinculación.
  - Especificación del procesamiento de comprobación de tipos.
  - Especificación del procesamiento de asignación de espacio.
  - Descripción del repertorio de instrucciones de la máquina-p necesario para soportar la traducción de **Tiny** a código-p.
  - Especificación del procesamiento de etiquetado.
  - Especificación del procesamiento de generación de código.
- 2) Una implementación del procesador para **Tiny**. Dicho procesador integrará los constructores de ASTs desarrollados en la fase 3. El programa recibirá como argumento (i) el archivo a analizar; (ii) una opción *op* que indique el constructor de árboles a aplicar (si *op* es *desc* el constructor a aplicar será el descendente; si es *asc* será el ascendente). Entonces:
  - El procesador aplicará el constructor de árboles elegido para construir el AST. En caso de detectar errores durante esta fase mostrará un mensaje legible del primer error (léxico o sintáctico) detectado, y terminará la ejecución.
  - Si el procesador ha construido satisfactoriamente el AST, aplicará sobre el mismo el procesamiento de vinculación. En caso de descubrir errores de vinculación, mostrará mensajes legibles de **todos** los errores encontrados. Dichos mensajes deberán ir acompañados de la fila y la columna en el texto fuente donde se ha producido el error. Una vez finalizado el procesamiento, si ha detectado algún error, deberá terminar la ejecución del procesador.
  - Si la vinculación se ha realizado correctamente, el procesador llevará a cabo el procesamiento de comprobación de tipos. Lo mismo que en el caso anterior, si descubre errores deberá mostrar mensajes legibles de **todos** los errores descubiertos (junto con su localización en el texto fuente). En caso de detectarse errores, al finalizar el procesamiento el procesador detendrá su ejecución.
  - Una vez superadas con éxito las fases de vinculación y de comprobación de tipos, el procesador realizará el procesamiento de asignación de espacio, seguido del procesamiento de etiquetado, seguido del procesamiento de generación de código. Durante este último procesamiento, poblará la memoria de datos de la máquina-p.
  - Por último, el procesador invocará la ejecución de la máquina-p, con el fin de proceder a ejecutar el programa procesado.

Para realizar los procesamientos, pueden elegirse los estilos de procesamiento que se estimen más oportunos, e, incluso, combinar estilos de procesamiento (no obstante, se recomienda que el estilo de procesamiento principal sea el visitante, utilizándose programación recursiva para procesamientos minoritarios; el estilo basado en el patrón “intérprete” se desaconseja).

- 3) Una implementación de un emulador para la máquina-p apropiado para ejecutar programas **Tiny**. Para ello, podrá completarse el emulador básico que se proporciona como código de apoyo (dicho emulador proporciona ya soporte para todas las características de la máquina-p, incluida un mecanismo básico de gestión de memoria dinámica, e instrucciones para el manejo de registros de activación).

Fecha límite de entrega: **Viernes 10 de mayo de 2024, 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.pdf* con la memoria requerida en el punto 1) del trabajo a realizar; (ii) una carpeta *implementacion*, en el interior de la cuál debe incluirse toda la implementación requerida por el procesador (incluyendo el emulador de la máquina-p); (iii) una carpeta *pruebas* con distintos programas de prueba que permitan probar el procesador. 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.