

Guía Práctica 4 - ICI 445

Generación de Código en ANTLR

TEORÍA DE AUTÓMATAS Y COMPILADORES

Profesor: Ricardo Soto

Ejercicio 1: Instale el generador de código para el lenguaje Mile

- Descargue el archivo `mile-gen.zip` desde <http://www.inf.ucv.cl/~rsoto/cursos/ICI445/mile-gen.zip>
- Descomprima y copie en su workspace.
- Cree un proyecto seleccionando como fuente del proyecto la carpeta recientemente descomprimida.

Ejercicio 2: Comprenda la nueva clase `CodeGenerator.java`

- Diríjase a `src->mileCompiler->compilers->CodeGenerator.java`. Esta clase permitirá generar código C desde nuestro archivo fuente `mile`. Los métodos definidos en esta clase serán invocados por el iterador de AST `mileCodeGen.g`
- La clase `CodeGenerator.java` contiene 7 métodos:
 - `CodeGenerator` es el constructor de la clase.
 - `buildFile` invoca a `createFile`, el cual crea un archivo con la ruta y el nombre indicado.
 - `addHeader` genera el código de las cabeceras del archivo C.
 - `addVar` genera el código de la declaración de variables.
 - `createFile` crea el archivo e inicializa el objeto `PW`, el cual nos permite escribir el código generado.
 - `end` finaliza el código.
 - `closeFile` cierra el archivo.

Ejercicio 3: Complete el iterador de AST `mileCodeGen.g` y la clase `CodeGenerator.java`

- El iterador de AST sólo incluye las reglas para explorar las cabeceras y la declaración de variables. Ud. deberá incluir las reglas faltantes (`if`, `for`, expresiones, etc).
- Análogamente, la clase `CodeGenerator.java` sólo incluye métodos para generar código C para cabeceras y declaración de variables. Incluya los métodos faltantes.

Ejercicio 4: Pruebe el generador de código

- Ejecute la clase `Tool` utilizando como programa fuente el archivo `examples/test1.mile`. La ejecución debería generar un archivo C en la carpeta `output`.
- Ejecute el archivo `examples/test2.mile`. Este archivo contiene errores semánticos, pero el archivo se genera de todas formas. Evite la generación de código cuando se produzcan errores semánticos.

- Cree y ejecute un nuevo archivo para verificar la generación de código de los elementos agregados en el ejercicio 3.

Ejercicio 5: Implemente y pruebe el generador de código de su lenguaje.



Pontificia Universidad Católica de Valparaíso
Prof. Ricardo Soto, Ph.D.