

Compilador de COOL

Vivian Plasencia Calaña C412

Olivia González Torres C411

Introducción

El compilador para el lenguaje COOL fue desarrollado en C# ,usando Visual Studio 2019.Se dividió en tres fases fundamentales Parsing, Chequeo semántico y Generación de código .

Parsing

Debido a que ya existen generadores de parser implementados, para esta fase, se decidió utilizar ANTLR v4.7.1 . Entre sus múltiples funcionalidades, además de generar el parser y el lexer, brinda un mecanismo para recorrer el árbol de derivación obtenido utilizando el patrón Visitor,a partir de una gramática que se le brinde. Esto se aprovechó para la construcción del AST utilizado en la siguiente fase.

Chequeo Semántico

Para esta fase se realizaron dos recorridos sobre el AST ambos usando el conocido patrón Visitor ,el primero (TypeBuilderVisitor) para recolectar los tipos ,métodos y variables del programa, que en el segundo recorrido se chequeará si su uso está o no correcto que son almacenados en una estructura denominada Contexto,la cual también forma una jerarquía, esta es una solución elegante a, por ejemplo ,el problema del ocultamiento de variables .

En el segundo recorrido (TypeCheckerVisitor), también se usa el Visitor para recorrer el AST y computar los tipos correspondientes, además de encontrar los errores de tipo semántico que puedan existir. Estos se almacenan en una lista que, si se encuentra vacía se procede a ejecutar la tercera fase ,sino se imprimen los errores en consola y se termina la ejecución.

Generación de Código

Esta fase se dividió a su vez en dos fases, generación de un código intermedio y del código intermedio generar MIPS.

Generación de código intermedio

Para llevar a cabo la generación de código se decidió utilizar un lenguaje intermedio lo más cercano a MIPS, cada instrucción de este código intermedio tiene una correspondiente en MIPS bastante intuitiva, ya que de esta manera se facilita la posterior fase. También se usó el patrón Visitor para recorrer el AST de COOL. Además de que se implementó una Virtual Table y un manejador de variables para apoyar el proceso.

Generación de MIPS

Para generar código Mips se realizan dos pasadas por los nodos que representan el código intermedio anteriormente generado. La primera pasada para obtener y guardar información importante que facilitará que en la segunda pasada se genere el código Mips.

Uso del compilador

Para ejecutar el compilador:

En Windows:

- En la consola poner : >dotnet <ruta de la librería del programa.dll> y luego cuando lo pidan, poner la ruta del archivo .cl.