

1- Opciones del compilador:

El compilador recibe como entrada un archivo .cl (COOL) y devuelve un archivo con código MIPS.

2- Arquitectura del compilador:

El programa funciona usando el patrón visitor para ejecutar cada una de sus fases.

Dispone de 5 módulos diferentes (sin contar el chequeo lexicográfico):

1. Fase de chequeo semántico
2. Fase de construcción de tipos
3. Fase de chequeo de tipos
4. Generación de código de 3 direcciones (CIL)
5. Generación de código MIPS

Se ha utilizado una gramática LALR para el lexer del compilador.

3- Problemas técnicos

Los problemas técnicos más interesantes presentados en el proceso de construcción del compilador fueron el hecho de hacer boxing/unboxing para transformar tipos por valor a tipos por referencia a través del tipo Object. Fueron necesario controles a la hora de generación de MIPS para detectar tipos por valor y reservarles espacio en memoria para almacenarlos como tipos por referencia.

En cuanto en el momento del unboxing en la instrucción case hubo que detectar si lo que se tenía en la variable requería de hacerle unboxing para hacer la comparación de tipos en las ramas de la instrucción.

4- Link al repositorio publico del codigo fuente:

<https://www.github.com/youkai95/compilador-2019>