



[M1] Fase de Compilación: Análisis Semántico

Santiago Pereira - 22318
Nancy Mazariegos - 22513

1. Estructura del Proyecto

El proyecto se organiza en los siguientes archivos principales:

- **main.py**
Es el punto de entrada del compilador. Su trabajo es: leer el archivo fuente en Compiscript, ejecutar las fases de análisis léxico y sintáctico con ANTLR4, llamar al analizador semántico, mostrar en consola los errores, advertencias y la tabla de símbolos.
- **analizador_semantico.py**
Contiene el Visitor semántico. Aquí se revisan las reglas de tipos, ámbitos, funciones, clases, objetos y estructuras de control.
- **sistema_tipos.py**
Maneja el sistema de tipos. Aquí se valida que las operaciones aritméticas, lógicas y comparaciones usen tipos correctos.
- **tabla_simbolos.py**
Implementa la tabla de símbolos, que guarda información sobre variables, constantes, funciones, clases y métodos. También administra los diferentes ámbitos del programa (global, funciones, clases, ciclos, etc.).

2. Funcionalidades Principales

2.1 Sistema de Tipos (sistema_tipos.py)

- Revisión de operaciones aritméticas (+, -, *, /, %) con números
- Revisión de operaciones lógicas (&&, ||, !) con booleanos
- Comparaciones (==, !=, <, <=, >, >=) con tipos compatibles
- Verificación de asignaciones: el valor debe coincidir con el tipo de la variable
- Constantes deben declararse con un valor
- Soporte básico para arrays: tipo de elementos y validación de índices

2.2 Manejo de Ámbitos (tabla_simbolos.py)

- Resolución de nombres de variables, funciones y clases según el ámbito
- Error si se usa una variable no declarada
- Bloqueo de redeclaración en el mismo ámbito
- Creación de nuevos entornos para bloques, funciones, clases y ciclos
- Advertencias cuando una variable no se usa

2.3 Funciones y Procedimientos (analizador_semantico.py)

- Validación del número y tipo de argumentos en llamadas
- Validación del tipo de retorno
- Soporte para funciones recursivas
- Error si una función se declara más de una vez
- return solo permitido dentro de funciones

2.4 Control de Flujo (analizador_semantico.py)

- Condiciones en if, while, for, do-while, switch deben ser booleanas
- break y continue solo pueden usarse dentro de bucles
- return no permitido fuera de funciones

2.5 Clases y Objetos (analizador_semantico.py y tabla_simbolos.py)

- Declaración de clases con atributos y métodos
- Validación de acceso a atributos y métodos con dot notation (obj.prop, obj.metodo())
- Verificación del constructor: argumentos deben coincidir con la definición
- Uso de this solo dentro de métodos de clase

2.6 Integración y Ejecución (main.py)

- Conecta todas las fases: léxica, sintáctica y semántica
- Muestra resultados de forma clara en consola
- Imprime la tabla de símbolos y estadísticas finales

3. Ejecución del Compilador

1. Abrir terminal en la carpeta del proyecto
2. Ejecutar el comando: python main.py archivo.cps (donde archivo.cps es un programa escrito en Compiscript)
3. El compilador mostrará:
 - a. Resultados del análisis sintáctico
 - b. Errores y advertencias del análisis semántico

4. Pruebas

Se hicieron pruebas con casos válidos y casos erróneos:

- Correctos: funciones bien definidas, variables con tipos correctos, constructores válidos
- Erróneos: variables no declaradas, tipos incompatibles, this fuera de un método, atributos inexistentes

Link al repositorio: <https://github.com/SRV-JSPA/analizador-semantico.git>