

Informe compiladores Parcial 2

Descripción de la Problemática

Este informe aborda el análisis sintáctico y semántico realizado en un compilador desarrollado para un lenguaje de programación específico, empleando la herramienta ANTLR para generar parsers y analizadores. ANTLR permite definir una gramática que guía el análisis sintáctico, asegurando que el código fuente cumpla con la estructura esperada del lenguaje, mientras que el análisis semántico se ocupa de la validación más profunda del contexto y uso de variables y funciones.

Desarrollo de la Solución Propuesta

1. Desarrollo de la solución propuesta

El análisis se implementa mediante dos clases extendidas a partir de las generadas por ANTLR: `CustomListener` y `CustomVisitor`. Estas clases se complementan con una estructura de tabla de símbolos para gestionar el ámbito y la validez de los identificadores a lo largo del código.

Clase `CustomListener`:

- **Análisis Sintáctico:** Supervisa la estructura del código asegurando la correcta delimitación de bloques (`{...}`), el uso apropiado de símbolos gramaticales y la sintaxis de las instrucciones.
- **Análisis Semántico:** Utiliza la `TablaSimbolos` para controlar las declaraciones y referencias de variables y funciones. Verifica la

declaración previa de identificadores antes de su uso, maneja el alcance de las variables correctamente y asegura que todas las variables estén inicializadas antes de su uso.

Clase `CustomVisitor` :

- **Recorrido de Nodos:** Enfocado principalmente en la contabilización y seguimiento de asignaciones, permitiendo un análisis de cuántas operaciones de asignación se realizan y su contexto dentro del árbol de sintaxis.

Tabla de Símbolos:

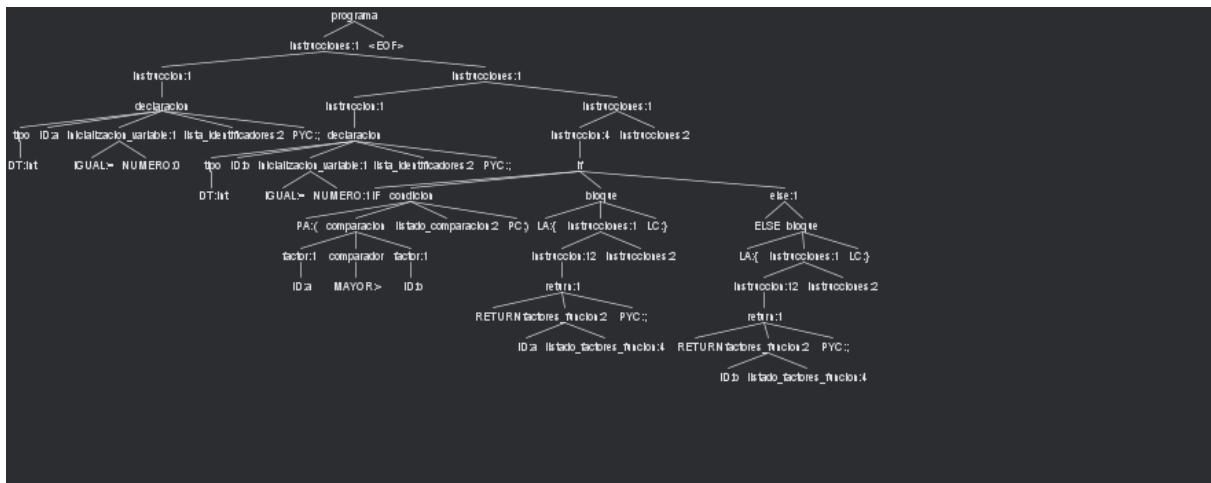
- **Gestión de Ámbitos y Símbolos:** Fundamental para el análisis semántico, permite añadir y remover ámbitos al entrar y salir de bloques de código. Añade y busca símbolos, llevando un registro de su tipo, si están inicializados y si han sido usados, lo que es crucial para validar la correcta declaración y uso de variables y funciones en diversos ámbitos

2. Conclusiones

La implementación del análisis sintáctico y semántico es crucial para la creación de un compilador eficaz. Mientras el análisis sintáctico se asegura de que el código fuente siga las reglas gramaticales del lenguaje, el análisis semántico profundiza en la lógica del uso de identificadores, evitando errores que podrían llevar a comportamientos incorrectos o fallas en tiempo de ejecución.

Informe sobre el Análisis Sintáctico y Semántico en Compiladores utilizando ANTLR

3. Arbol AST



4. Código de usado para el compilador:

```
int a = 0;
int b = 1;

if (a > b) {
    return a;
}
else {
    return b;
}
```

5. Repositorio

https://github.com/manudearagon/Parcial_2-TC