



TC

☰ Etiquetas	Informe Proyecto
🕒 Creada	@23 de mayo de 2023 11:01
🕒 Actualizado	@23 de mayo de 2023 11:17
➤ Artículo principal	
📅 Fecha	@23 de mayo de 2023
➤ Subítem	

Evaluación del Parser ANTLR para la Interpretación de Código en C

Contexto y Desafío Enfrentado

Estrategia y Desarrollo de la Solución

Reflexiones y Conclusión

Evaluación del Parser ANTLR para la Interpretación de Código en C

Contexto y Desafío Enfrentado

El lenguaje de programación C, notorio por su eficiencia y versatilidad, es ampliamente utilizado en la comunidad de desarrolladores. Para garantizar la correcta ejecución de los programas escritos en este lenguaje, es imprescindible

adherirse a ciertas reglas sintácticas y semánticas definidas por el propio lenguaje. Consecuentemente, se requiere la implementación de un parser que pueda interpretar y analizar el código para garantizar el cumplimiento de estas normas.

Este estudio se enfocó en la construcción de un parser para código C capaz de identificar y analizar elementos críticos del lenguaje, tales como bloques de código, declaraciones y asignaciones, operaciones aritméticas, llamadas a funciones y estructuras de control como `if`, `for` y `while`. El objetivo principal era la correcta generación de un Árbol Sintáctico Abstracto (AST) para cualquier archivo de entrada escrito en C.

Estrategia y Desarrollo de la Solución

Para abordar este desafío, optamos por ANTLR, un generador de parsers poderoso y flexible que nos permitió definir la gramática del lenguaje C y generar el parser correspondiente.

La gramática fue definida en un archivo `.g4`, que encapsula las reglas sintácticas del lenguaje C. Esta gramática abarca definiciones para diversos componentes vitales del lenguaje, como bloques de código, declaraciones y asignaciones, operaciones aritméticas, llamadas a funciones y estructuras de control `if`, `for` y `while`.

Con la gramática definida, utilizamos ANTLR para producir el parser. Este parser es capaz de tomar un archivo de entrada en C y generar un AST correspondiente, que refleja la estructura del programa en cuestión. Adicionalmente, el parser puede validar si el archivo de entrada se adhiere a las reglas descritas en la gramática y reportar cualquier inconsistencia que detecte.

Reflexiones y Conclusión

La utilización de ANTLR para la implementación de un parser de código C demostró ser una estrategia eficaz para abordar el desafío propuesto. ANTLR proporcionó un medio robusto y flexible para la definición de la gramática del lenguaje C y la generación del parser correspondiente.

Durante el desarrollo de este proyecto, obtuvimos un entendimiento profundo sobre las reglas sintácticas y semánticas del lenguaje C, así como también de cómo se pueden representar en forma de gramática. Aprendimos cómo generar un parser a partir de dicha gramática utilizando ANTLR y cómo este puede ser utilizado para interpretar el código C y generar un AST.

En síntesis, consideramos que el proyecto fue exitoso. No solo cumplimos con los objetivos planteados, sino que también adquirimos un valioso conocimiento que nos será útil en futuros proyectos de programación.

<https://github.com/SecondPort/Tecnicas-de-Compilacion>