



Informe Parcial 2

☰ Etiquetas	Informe
🕒 Creada	@4 de junio de 2023 11:58
🕒 Actualizado	@4 de junio de 2023 12:12
➤ Artículo principal	<u>TC</u>
📅 Fecha	
➤ 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

La comprensión de lenguajes de programación propios se ha convertido en una necesidad en diversas áreas de la informática, como la creación de compiladores e intérpretes, el análisis de código y la construcción de editores de código y entornos

de desarrollo integrado (IDE). Para hacer frente a esta necesidad, es esencial desarrollar un parser capaz de interpretar el código escrito en estos lenguajes.

En este estudio, desarrollamos un parser para un lenguaje de programación propio con una gramática definida. Este lenguaje cuenta con características fundamentales como declaraciones e inicializaciones de variables, operaciones aritméticas y bloques de instrucciones, entre otros. El objetivo principal era construir un parser que pudiera interpretar el código, validar su sintaxis y semántica, y generar un Árbol Sintáctico Abstracto (AST).

Estrategia y Desarrollo de la Solución

Para desarrollar la solución, elegimos ANTLR, un potente generador de parsers, para definir la gramática de nuestro lenguaje y producir el parser correspondiente.

La gramática, definida en un archivo `.g4`, encapsula las reglas sintácticas y semánticas de nuestro lenguaje. Contiene definiciones para declaraciones y asignaciones de variables, operaciones aritméticas y bloques de instrucciones.

Una vez definida la gramática, usamos ANTLR para generar el parser. Este parser es capaz de tomar un archivo de entrada en nuestro lenguaje y generar un AST correspondiente, que representa la estructura del programa. Además, el parser puede validar si el archivo de entrada cumple con las reglas descritas en la gramática y reportar errores sintácticos y semánticos.

Reflexiones y Conclusión

El uso de ANTLR para la implementación de un parser para nuestro lenguaje propio demostró ser una estrategia efectiva. ANTLR proporcionó un medio robusto y flexible para definir la gramática de nuestro lenguaje y generar el parser correspondiente.

Durante el desarrollo de este proyecto, adquirimos un entendimiento profundo sobre cómo se pueden representar las reglas sintácticas y semánticas de un lenguaje de programación en forma de gramática. Aprendimos a generar un parser a partir de dicha gramática utilizando ANTLR y a utilizar este parser para interpretar el código y generar un AST.

En conclusión, el proyecto fue exitoso. No solo cumplimos los objetivos establecidos, sino que también adquirimos conocimientos valiosos que nos serán útiles en futuros proyectos de desarrollo de lenguajes de programación y análisis de código.

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