UNIVERSIDAD DE LA DEFENSA NACIONAL

Centro Regional Universitario Córdoba IUA

Desarrollo de Herramientas de Software

Facultad de Ingeniería - Ingeniería Informática

Título: *"Desarrollo de Compiladores con ANTLR4"*

Autor: Vega Majul, Joaquín Ignacio.

Profesor: Eschoyez, Maximiliano.

Córdoba, 01 de Diciembre de 2023

Problemática Abordada

El trabajo final se centra en la implementación de un compilador para el lenguaje C utilizando ANTLR4, una herramienta eficiente para la generación de analizadores léxicos y sintácticos. La problemática principal se relaciona con la optimización del código intermedio generado, buscando mejorar la eficiencia y la legibilidad del código resultante.

Desarrollo de la Solución

El desarrollo del compilador se llevó a cabo en varias fases, comenzando con el diseño de la gramática para el análisis léxico y sintáctico con ANTLR4. Posteriormente, se emplearon listeners y visitors para realizar tareas específicas en distintas etapas del proceso de compilación.

1. Diseño de la Gramática:

La gramática fue diseñada para definir las reglas sintácticas del lenguaje fuente. Esto incluyó la especificación de tokens, la jerarquía de la estructura del programa y la relación entre las diferentes construcciones del lenguaje. ANTLR4 facilitó la generación automática del analizador léxico y sintáctico a partir de esta gramática.

2. Utilización de Listeners para Análisis:

Se implementaron listeners de ANTLR4 para llevar a cabo acciones significativas durante el recorrido del árbol de análisis sintáctico, abordando tanto aspectos léxicos como semánticos. Estos listeners no solo desempeñaron un papel vital en la identificación y gestión de errores durante el análisis, sino que también facilitaron la construcción de tablas de símbolos y otros elementos esenciales del proceso de compilación, incluyendo el análisis semántico para garantizar la coherencia en la generación de código intermedio.

3. Generación de Código Intermedio con Visitors:

Una vez ya completado el análisis léxico, sintáctico y semántico, se introdujo la fase de generación de código intermedio utilizando visitors de ANTLR4. Estos visitors recorrieron el árbol de análisis sintáctico para traducir las construcciones del lenguaje fuente a instrucciones en un formato intermedio. La modularidad y flexibilidad de los visitors permitieron una implementación clara y mantenible de esta etapa del compilador.

4. Optimización del Código Intermedio:

Dentro de la generación de código intermedio, se aplicaron optimizaciones como la propagación de constantes y la eliminación de líneas redundantes. Estas optimizaciones, implementadas sobre el código intermedio original, contribuyeron a mejorar la eficiencia y legibilidad del código original.

5. Evaluación y Corrección de Errores:

Los listeners también jugaron un papel crucial en la identificación y manejo de errores durante el análisis léxico, sintáctico y semántico. Un contador de errores se utilizó para rastrear y gestionar problemas encontrados en el código fuente, garantizando una experiencia de compilación más robusta.

Conclusión

El desarrollo de este compilador para el lenguaje C utilizando ANTLR4 ha sido un viaje integral a través de las fases fundamentales de la construcción de un sistema de compilación. Desde el análisis léxico, que se encarga de la identificación y clasificación de los tokens, hasta el análisis sintáctico, que estructura el programa según las reglas gramaticales definidas, cada etapa ha desempeñado un papel crucial.

El análisis léxico, sintáctico y semántico fue implementado con éxito, permitiendo una comprensión profunda y precisa del código fuente. La utilización de listeners y visitors de ANTLR4 resultó esencial para ejecutar acciones específicas durante el análisis sintáctico, y la generación de código intermedio proporcionó una representación eficiente y manipulable del programa.

La aplicación de técnicas de optimización, como la propagación de constantes y la eliminación de líneas redundantes, contribuyó significativamente a mejorar la calidad y eficiencia del código final. Este enfoque integral, que abarca desde el análisis hasta la optimización, refleja la importancia de una planificación detallada y una implementación cuidadosa en el desarrollo de compiladores.

Enlace al Repositorio de GitHub

URL: https://github.com/joaquin-iua/DHS_Compilador.git

Tag de Entrega: *"Entrega 01/12 Examen Final"*

.