



INSTITUTO TECNOLÓGICO DE COSTA RICA

Proyecto #1: Scanner

Escuela de Ingeniería en Computación
Compiladores e Intérpretes IC-5701

Alonso Navarro Carrillo, c. 2022236435

Carlos Venegas Masis, c. 2022153870

Valeria, c.

Ing. Ericka Marín Schumann
II Semestre 2024

Tabla de contenidos

Introducción	2
Estrategia de solución	2
Análisis de resultados	2
Lecciones aprendidas	2
Casos de prueba	2
Manual de usuario	2
Bitácora	4
Bibliografía	5

Introducción

Este proyecto se sitúa en la primera etapa del desarrollo de un compilador para el lenguaje de programación C, conocida como el Análisis Léxico. El objetivo principal de esta etapa es diseñar y construir un scanner que sea capaz de identificar y clasificar los diferentes tokens que conforman un programa en C. Para lograr esto, se utilizó la herramienta JFlex, la cual permite definir expresiones regulares que describen los patrones de los tokens a reconocer.

Estrategia de solución

Después de leer extensamente la documentación de JFlex, se comenzó a diseñar las expresiones regulares de los tokens que el scanner debía reconocer. Aquí surgió el primer problema: el scanner reconoce tokens según la prioridad de orden. Es decir, si la primera expresión regular es un punto, no se reconocerá ningún otro token, ya que este metacaracter coincidiría con cualquier carácter, identificándolo como un error.

Análisis de resultados

Lecciones aprendidas

Casos de prueba

Manual de usuario

Instalación

Para construir y ejecutar el proyecto, es necesario tener Java instalado en tu sistema. Sigue estos pasos para configurar el proyecto:

1. Clona el repositorio:

```
git clone https://github.com/AlonsoNav/CCompilerJFlex.git
cd your-repo
```

2. Genera el archivo CLexer:

Actividad	Porcentaje realizado	Justificación
Desplegar lista de errores léxicos	100%	
Desplegar listado de tokens encontrados	100%	
Mostrar tipo de token, línea y cantidad de apariciones por cada token	100%	
Manejar 4 tipos grandes (operadores, literales, ids, palabras reservadas) de tokens	100%	
Ignorar comentarios en línea y bloque	100%	
Identificar todos los operadores válidos de C	100%	
Identificar todos los literales válidos de C	100%	
Identificar todos los identificadores válidos de C	100%	
Identificar todas las palabras reservadas de C	100%	
Definir errores léxicos	100%	

```
java -jar lib/jflex-full-1.9.1.jar src/scanner/CLexer.flex
```

3. Compila el proyecto:

```
javac -d bin -sourcepath src src/app/Main.java src/scanner/CLexer.java .\src\
```

Uso

Para ejecutar el compilador con un archivo de entrada, utiliza el siguiente comando:

```
java -cp bin app.Main input_file
```

También puedes enviar la salida a un archivo .txt con

```
java -cp bin app.Main input_file > output.txt
```

Bitácora

Fecha: 26-08-2024

En la primera reunión del equipo de trabajo, se acordó que CV se encargará de los expresiones regulares de los operadores y del formato de impresión de la tabla. AN diseñará la estructura de los tokens y sus errores, así como las expresiones regulares de los identificadores y palabras reservadas. VG se responsabilizará de los literales. Además, se decidió que la documentación se realizará en LaTeX y que GitHub será utilizado como sistema de control de versiones.

Bibliografía

- [1] Klein, G., Rowe, S., & Décamps, R. (marzo de 2023). *JFlex User's Manual*. JFlex Team. En: <https://www.jflex.de/manual.html>.