

Laboratorio 1 - Introducción a ANTLR

Carlos Alberto Valladares Guerra

Carné: 221164

Análisis del funcionamiento de ANTLR, la gramática y el driver

Este laboratorio introduce el uso de ANTLR para construir analizadores léxicos y sintácticos. El entorno se ejecuta dentro de un contenedor Docker preconfigurado para facilitar la compilación de gramáticas y pruebas con Python.

¿Qué es un archivo `.g4`?

- Es un archivo que define una gramática en ANTLR. Se compone de dos secciones principales:
- Reglas léxicas (tokens): especifican los elementos básicos del lenguaje como identificadores (`ID`), números (`INT`), operadores (`+`, `-`, `*`, `/`), etc.
- Reglas sintácticas (parser rules): definen cómo se combinan los tokens para formar expresiones válidas.
- Etiquetas de reglas (`#Nombre`): sirven para nombrar alternativas específicas de las reglas sintácticas y ayudan al análisis del árbol.
- `WS -> skip`: define espacios en blanco a ignorar.

Ejemplo de tokens en `MiniLang.g4`:

```
ID : [a-zA-Z]+ ;
INT : [0-9]+ ;
ADD : '+' ;
SUB : '-' ;
```

Ejemplo de reglas sintácticas:

```
expr: expr ( '*' | '/' ) expr # MulDiv
    | expr ( '+' | '-' ) expr # AddSub
    | INT # int
    | ID # id
    | '(' expr ')' # parens
    ;
```

¿Qué hace el `#Nombre`?

Permite etiquetar una alternativa para que ANTLR pueda identificarla mejor al generar el árbol sintáctico. Por ejemplo, `#AddSub` es una etiqueta para la suma y resta.

¿Qué hace el archivo `Driver.py`?

1. Este archivo es el punto de entrada en Python para analizar un archivo de entrada:

1. Abre un archivo de texto (como `program_test.txt`).
2. Usa el lexer generado por ANTLR (`MiniLangLexer`) para convertir el texto en tokens.
3. Pasa esos tokens al parser (`MiniLangParser`).
4. Ejecuta la regla inicial definida en la gramática (`prog`).
5. Si la sintaxis es válida, no muestra nada. Si hay errores, ANTLR los muestra en consola.

Comando de prueba:

```
python Driver.py program_test.txt
```

Funcionamiento en la práctica

Cuando el archivo `program_test.txt` contiene:

```
a = 5
b = 4
c = a + b
```

El analizador se ejecuta correctamente y no muestra errores.

Si se comete un error de sintaxis:

```
a 5 <-- Falta el '='
b = 4
```

ANTLR detecta y muestra un error en la consola indicando la línea y posición del fallo.

Comandos usados en el laboratorio

```
antlr -Dlanguage=Python3 MiniLang.g4
python3 Driver.py program_test.txt
```

Enlace de GitHub: <https://github.com/vgcarlol/Compis/tree/main/lab-1>

Enlace de video: <https://youtu.be/MDEMAckImp0>