

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería



Laboratorio 1

Edwin Andrés Ortega Kou - 22305

Compiladores

Guatemala, julio 2025

Análisis sobre la gramática de ANTLR

Archivo .g4

Un archivo .g4 contiene la definición formal de un lenguaje y se divide en las siguientes partes:

1. Nombre de la gramática y es la siguiente:
grammar MiniLang;
2. Reglas sintácticas (parser), definen como se construyen las estructuras del lenguaje y se escriben siempre en minúscula:
prog : stat+ ;
stat : expr NEWLINE | ID '=' expr NEWLINE | NEWLINE ;
expr : expr ('*'|'/') expr | expr ('+'|'-') expr | INT | ID | '(' expr ')';
3. Reglas léxicas (lexer), definen los tokens o unidades básicas del lenguaje y empiezan en mayúsculas:
INT : [0-9]+ ;
ID : [a-zA-Z]+ ;
ADD : '+' ;
MUL : '*' ;
WS : [\t]+ -> skip ; // ignora espacios

También se pueden identificar algunos detalles importantes de la gramática utilizada como:

- prog: stat+;
Se utiliza como punto de entrada e indica que un programa es una o más instrucciones (stat).
- # nombreDeReglaAlternativa
Se usa para nombrar ramas alternativas del parser. Esto permite generar nodos más fácilmente cuando se implementan visitors o listeners. Como #printExpr, #assign, #blank que son etiquetas que ANTLR usará internamente.
- -> skip
Hace que un token, por ejemplo, que un espacio en blanco sea ignorado por el lexer.

Archivo Driver.py

El archivo Driver.py sirve como punto de entrada en Python para probar la gramática generada.

Como parte de su estructura, importa las clases generadas por ANTLR desde MiniLang.g4 y de esta forma lee el archivo de prueba program_test.txt, tokeniza la entrada con MiniLangLexer, Pasa los tokens al parser (MiniLangParser) y analiza a partir de la regla prog (la regla principal de la gramática).

Adicionalmente le he agregado un manejo de errores que permite dar una retroalimentación clara al usuario sobre si el archivo de entrada es válido o no.

Pruebas realizadas

Se escribieron varias entradas válidas e inválidas en `program_test.txt` para observar el comportamiento del parser. ANTLR demostró ser capaz de identificar errores de sintaxis y reportarlos con línea y posición, por ejemplo:

line 2:2 extraneous input '=' expecting NEWLINE

Esto facilitó entender cómo responde ANTLR cuando la entrada no cumple con la gramática.

Video explicativo

https://youtu.be/ZW_6SGGB5pQ

Repositorio de Github

<https://github.com/EdwinOrtegaK/Compis-Lab1>