

# Compiler Lab2 Parser

201250104 苏致成

November 2022

## 1 概述

### 1.1 目标

完成对 SysY 语言的语法分析与高亮。

1. 若输入文件不存在语法错误时：按照规定格式输出语法树 (含高亮)。
2. 当输入文件存在语法错误时：输出语法错误在哪一行。

## 2 实现

### 2.1 使用工具

git、Antlr、Intellij idea、JDK11、Makefile

### 2.2 实现功能

向 main 方法中传递文件参数，程序可以对其进行语法分析，其余过程详见“目标”。

### 2.3 实现过程

1. 编写 g4 文件，需注意最上方需要 parser grammar，需要注意终结符书写参考 lexer 中写法，注意左递归处理过程。
2. 利用 Antlr 生成对应的 SysYParser.java 文件。
3. 编写 Listener 并移除原有的 ErrorListeners，在此输出错误的语法行数。
4. 编写继承自 SysYParserBaseVisitor 的 visitor，分别重写 visitChildren 和 visitTerminal。其中 children 指的是除终结符之外的节点，terminal 指终结符节点。

## 3 遇到障碍

### 3.1 左递归

#### 3.1.1 问题描述

sysY 为了处理左递归采用了复杂的语法定义方式。

### 3.1.2 解决方式

Antlr 自己能够处理左递归。可以直接写成例如如下的形式：

```
exp
: L_PAREN exp R_PAREN
| lVal
| number
| IDENT L_PAREN funcRParams? R_PAREN
| unaryOp exp
| exp (MUL | DIV | MOD) exp
| exp (PLUS | MINUS) exp
;
```

### 3.2 无法获取该节点的深度

解决方式：找到接口 `node.getRuleContext().depth()`。

### 3.3 无法获取终结符的深度

解决方式：找到其父节点，根据其父节点的深度判断终结符的深度。

### 3.4 错误输出

#### 3.4.1 问题描述

始终无法发现错误的情况。

#### 3.4.2 解决方式

手册中的 `INTEGER_CONST`，被误写成 `INTEGR_CONST`。于是只能沿用 `INTEGR_CONST`。

### 3.5 类型错误

#### 3.5.1 问题描述

`sysYParser.program()` 返回类型为 `ProgramContext`，而不是 `ParserTree`。

#### 3.5.2 解决方式

导包错误，重新导包。