

编译原理 lab2

一、具体实现

1. 首先按照 Sysy 语法规则以及助教实验手册编写 SysyYParse.g4 文件,之后为避免出现问题,虽然上次实验 AC 但使用的仍是助教实验手册中的 SysyLexer.g4 文件。生成 tokens 后将 SysyLexer 的 tokens 插入写好的 SysyYParse.g4 文件,并生成相应的类。

2. 之后在助教实验手册的提示下,编写 Visitor 继承自 SysYParserBaseVisitor<Void>

成员变量:

private final String[] colors, 此数组利用表驱动,在 SysyLexer.rulesName[]表基础上,将有颜色的词对应的元素改为相应的颜色。

```
private final String[] colors = new String[] {
    "[orange]", "[orange]", "[orange]", "[orange]", "[orange]", "[orange]", "[orange]", "[orange]", "[orange]",
    "[blue]", "[blue]", "[blue]", "[blue]", "[blue]", "[blue]", "[blue]", "[blue]", "[blue]", "[blue]",
    "[blue]", "[blue]", "[blue]", "[blue]", "[blue]", "[blue]", "L_PAREN", "R_PAREN", "L_BRACE", "R_BRACE",
    "L_BRACKET", "R_BRACKET", "COMMA", "SEMICOLON", "[red]", "[green]",
    "DECIMAL_CONST", "OCTAL_CONST", "HEXADECIMAL_CONST", "WS", "LINE_COMMENT",
    "MULTILINE_COMMENT"
}
```

private String blanks 来记录即将打印的要空几个空格。

四个函数:

Override 了两个函数 visitChildren, visitTerminal, 创建了 UpperFirstLetter 和 needSkip。

visitChildren: 在方法源码基础上,在方法源码片段前,先打印(blanks+ruleName),再将 blanks 加两个空格,在源码片段后减去两个空格,以此实现语法树的打印。

UpperFirstLetter: 由于 SysyYParse 中的 ruleNames 均为首字母小写,所以写了此方法将字符串首字母变为大写。

needSkip: 实验要求跳过如:;, (,), {, }, :, 注释等非终结符,只打印终结符,故使用此方法。

本方法个人认为比较巧妙,因为终结符需高亮,即有颜色,所以判断一下符号是否有颜色即可判断是否为终结符。具体我是将上述成员变量中的 colors[] 中该元素对应的字符串是否和 SysyLexer.rulesName[] 表对应的字符串相同,若相同说明其没有颜色,即返回 true,否则返回 false。

visitTerminal: 除了上述 needSkip 判断是否为终结符外,此方法也判断了是否为十六进制或八进制,若是将其转换为十进制表示,具体方法和 lab1 中相同。之后将处理好的终结符带上 colors[] 表中查询到的颜色输出。返回值为 super.visitTerminal()。

3. 错误处理和 lab1 相同,自己创建了一个 MySysYParserErrorListener 类,继承自 BaseErrorListener 类,具体方法也是重写了 syntaxError 方法,并为判断是否有 error 新建了 haserror 成员变量。

4. Main 函数则按照说明手册使用 visitor 深度优先遍历了语法树,但在创建 visitor 前多加了一个判断是否有报错,若有则直接 return。

二、遇到的问题以及解决方法:

1. 在写好代码后运行助教给的例子 2 发现结果和例子不同,如下图,但并不影响 AC。

```
Error type B at Line 2: no viable alternative at input 'a[]'
Error type B at Line 3: mismatched input ']' expecting {'+', '-', '*', '/', '%', '[', ';'}
```

```
Error type B at Line 2: mismatched input ']' expecting {'+', '-', '!', '(', IDENT, INTEGR_CONST}
Error type B at Line 3: mismatched input ']' expecting {'+', '-', '*', '/', '%', '[', ';'}
```

2. 起初我直接在 **antlr4** 工具生成好的类中进行代码修改，但在 **OJ** 上跑的时候发现，在编译时会重新生成该类，故要将改动放在新建的类中。
3. **AC** 满分 4300 分，只得了 4000 分，发现要处理十六进制和八进制。