

# 编译原理 lab1 实验报告

李维璇 201250058

## 实验思路

- g4 代码中, 主要实现 IDENT 和 INTEGR\_CONST, 分别如下:

```
IDENT :  
    (LETTER|'_' ) (LETTER|DIGIT|'_' )*;
```

```
INTEGR_CONST :  
    DECIMAL | OCTAL | HEX ;  
fragment DECIMAL :  
    '0' | (NONZERO DIGIT*);  
fragment OCTAL : // 前缀加上oct允许的字符  
    '0' OCTALDIGIT+;  
fragment HEX : // 前缀加上hex允许的字符  
    HEXPRE HEXDIGIT+;  
fragment LETTER : [a-zA-Z] ;  
fragment DIGIT : [0-9] ;  
fragment NONZERO : [1-9] ;  
fragment OCTALDIGIT : [0-7] ;  
fragment HEXPRE : '0x'|'0X';  
fragment HEXDIGIT : [0-9a-fA-F];
```

规则和思路参考的是 <<SysY 语言定义(2022 版)>>

- Main 中, 主要实现 token 的打印。
  - 首先实现 MyErrorListener extends BaseErrorListener, 覆写 public void syntaxError(Recognizer<?, ?> recognizer, Object offendingSymbol, int line, int charPositionInLine, String msg, RecognitionException e) 方法

```
static class MyErrorListener extends BaseErrorListener{  
    @Override  
    public void syntaxError(Recognizer<?, ?> recognizer, Object  
offendingSymbol, int line, int charPositionInLine, String msg,  
RecognitionException e) {  
        System.err.print("Error type A at Line "+line+":\n");  
        a=1;  
    }  
}
```

- 无异常时, 直接打印。如果是数字, 则需要对16进制和8进制进行讨论。主要打印的代码是:

```
for (Token token:allTokens)  
{  
    String temp = token.getText();  
    if (Objects.equals(types[token.getType()], "INTEGR_CONST"))  
    {
```

```

        if (temp.startsWith("0x") || temp.startsWith("0X"))
            temp =
String.valueOf(Integer.parseInt(temp.substring(2),16));
        else if (temp.startsWith("0") && temp.length() >= 2)
            temp = String.valueOf(Integer.parseInt(temp,8));
    }
    if (Objects.equals(types[token.getType()], "LINE_COMMENT")
        || Objects.equals(types[token.getType()],
"MULTILINE_COMMENT"))
        continue;
    System.err.print(types[token.getType()]+" "+temp+" at Line
"+token.getLine()+".\n");
}

```

- 有异常时，正常的token不再打印，所以通过一个static变量a来体现。初始化为0，当出错时程序定位到syntaxError方法，a改为1；在打印token之前检测a的值，若为1则退出。

```

if (a==1)
    return;

```

## 精巧的设计

- 为了通过Type取到对应的名字，通过表驱动来实现：

```

String[] types = new String[] {
    null, "CONST", "INT", "VOID", "IF", "ELSE", "WHILE", "BREAK",
"CONTINUE",
    "RETURN", "PLUS", "MINUS", "MUL", "DIV", "MOD", "ASSIGN", "EQ",
"NEQ",
    "LT", "GT", "LE", "GE", "NOT", "AND", "OR", "L_PAREN",
"R_PAREN", "L_BRACE",
    "R_BRACE", "L_BRACKET", "R_BRACKET", "COMMA", "SEMICOLON",
"IDENT", "INTEGR_CONST",
    "WS", "LINE_COMMENT", "MULTILINE_COMMENT"
types[token.getType()];

```

当然，后续发现 antlr 有封装好的方法来取得，所以这也不算太精巧。

## 遇到的困难

- 困难主要发生在环境的配置和g4文件的编译，在助教的补充文档下顺利解决。