

大作业第二部分 语法分析

在第一部分词法分析的基础上，完成 C 语言的语法分析器。每组同学自行分工合作完成作业，在 deadline 之前，每组由一人将完成的系统提交到系统的相应 folder。

目标

练习使用 Yacc 或者手工编写 C (C 语言子集) 语言的语法分析器，理解自上而下、自下而上的语法分析器的工作机制；掌握词法分析与语法分析之间的关系。语法分析器的输入为 C 语言源代码，输出为按扫描顺序进行分析或归约的正确/错误的判别结果。分析或归约按照语法规则进行。

软件需求

1. 完成 C 语言的语法分析器，语法分析器的输入为 C 语言源代码，输出语法树（或者产生语法树所用的产生式的推导/规约序列）。用 Yacc 生成语法分析器的，需要用 gcc 将语法分析器编译得到一个编译器。
2. 可以选择基于 Yacc 自动生成语法分析器，或者自行设计并实现语法分析器（可采用递归下降，LL(1)，算符优先或者 LR(1) 分析法之一）；
3. 实现语言：C, Java, Python 三者任选其一；

提交文件要求：

1. 基于 Yacc 实现的语法分析器，需要提交
C 语言的 LR(0)/LSR/LR(1) 文法 (pdf 文档)；
yacc.y 文件；
yacc.tab.h 文件；
yacc.tab.c 文件；
最终生成的编译器；
使用说明文件(readme file) (pdf 文档)；
2. 自动实现的语法分析器需要提交
语法分析器的完整设计，包括使用的语法分析方法进行分析的步骤 (pdf 文档)；
C 语言的 LR(0)/LSR/LR(1) 文法 (pdf 文档)；
语法分析器的全部源代码以及可执行文件；
使用说明文件(readme file) (pdf 文档)。

提交方式：

- ✓ 本科教学系统 → 编译原理 → 作业管理 → 大作业_语法分析
- ✓ 每组全部需要提交的文件放入一个文件夹，压缩 (.zip, .rar)，并以组名命名压缩包，例如“第一组.zip”

截止日期(Deadline)

2019 年 12 月 4 日 11: 59pm (GMT+08)

测试用例样本

```
void main() {  
    int i=2;  
    int j=4;  
    int result=0;  
    while(i <=j) {  
        result=result+j;  
        i++;  
    }  
    return result;  
}
```

注意事项

本次作业会进行抄袭检测，对检测出抄袭的作业，予以 0 分处理。

参考文献

[1] The Lex and Yacc Page <http://dinosaur.compilertools.net/#overview>