

# 编译原理实验三 中间代码生成 实验报告

组长：刘松桦 171860621

组员：吴晓阳 171860622

邮箱：[141760584@qq.com](mailto:141760584@qq.com)

## 完成的功能：

- 完成所有必做和选作1

## 中间代码数据结构

- 中间代码采取三地址代码形式，分为如下字段：
  - 参数1
  - 参数2
  - 目标
  - 操作指令
- 其中两个参数字段在某些指令（例如赋值指令）中只会使用1个，或1个也不使用（例如GOTO指令）
- 内部所有表（符号表，中间代码表，优化用的基本块表等）都采取自己实现的通用链表数据结构

```
30 struct _ASTNode_  
31     int height;  
32  
33     enum ASTNodeType type;  
34     union ASTNodeValue value;  
35     char* name;  
36     Type varType;  
37  
38     struct _ASTNode_* lc;  
39     struct _ASTNode_* rc;  
40  
41     ListHead parents;  
42     ConstValue constValue;  
43  
44     int accessTag;  
45     int removeTag;  
46 };
```

## 中间代码生成算法

- 所有算法均基于课本与课堂上所讲内容实现

- 对于表达式部分，采取抽象语法树的方式，在语义分析的基础上，构造抽象语法树，这样在生成中间代码的同时也可进行一定的优化
- 对于布尔表达式部分，使用回填技术生成代码
- 按照课本方案实现IF-ELSE分支以及循环结构的结构化生成，保证跳转正确且清晰

## 中间代码优化

---

- 本次实验大部分时间花费在代码优化上，主要实现了如下优化：
  1. 预先计算常量表达式
  2. 消除公共子表达式的重复计算，通过分配可用节点池，并配合检索技术，能够高效检测公共子表达式，即已经算过的表达式不再重复计算
  3. 实现基本块内的优化方案，消除不必要的中间变量
  4. 循环不变式外提