

lan3 实验报告

151220128 吴一楠

一、实现功能

按规定完成了所有基础内容，考虑到改动代码的复杂性，本次实验与实验 2 独立，在语义分析结束后重新执行生成中间代码（主要代码在 IR.h 中）。

```
CodesPointer translate_exp(syntax_node *p,Operand t);
CodesPointer translate_logic(syntax_node *p,Operand t);
CodesPointer translate_cond(syntax_node *p,Operand ltrue,Operand lfalse);
CodesPointer translate_assign(syntax_node *p,Operand t);
CodesPointer translate_dot(syntax_node *p,Operand t);
CodesPointer translate_arith_exp(syntax_node *p,enum IC_kind ic_kind,Operand t);
CodesPointer translate_call(syntax_node *p,Operand t);
CodesPointer translate_arr(syntax_node *p,Operand t);
CodesPointer translate_tree(syntax_node *p);
CodesPointer translate_stmt(syntax_node *p);
CodesPointer translate_if(syntax_node *p);
CodesPointer translate_if_else(syntax_node *p);
CodesPointer translate_while(syntax_node *p);
CodesPointer translate_vardec(syntax_node *p,Operand t);
CodesPointer translate_fundec(syntax_node *p);
CodesPointer translate_param(syntax_node *p);
```

根据实验手册中的翻译模式，使用与实验 2 类似的结构逐步完成。

代码无法对选做样例生成中间代码，结构体与数组在传递参数过程中仍然存在问题暂未解决。

二、编译执行

与前两阶段一致，执行 make parser 生成可执行文件。

```
if (argc != 3 ) {
    printf("input error\n");
    return 1;
}
FILE *f =fopen(argv[1],"r");
if (!f)
{
    perror(argv[1]);
    return 1;
}
outfile=fopen(argv[2],"w");
```

只有满足输入要求的命令才会被执行（./parser aaa bbb）。

aaa 为输入文件 bbb 为中间代码的输出文件

三、其他

本次实验与实验 2 结构类似，更加细节内容根据实验手册完成。

发现实验 2 的 bug：在某些地方 FUNC 与 FUNCTION 混淆，由于他们都是 enum 结构，故实验 2 未发现问题，但是他们处于 enum 的不同位置（即对应的 int 值不同），在本次实验中对结果产生了影响。