**语义分析器**

1. **静态语义检查**
2. 主要功能

（1）语法错误：输入的单词符号与文法产生式的“预期”不相符合。

（2）类型错误：需要借助于符号表才能发现的“前”、“后”不一致的情况，包括：

（a）表达式中引用的变量未经声明；

（b）表达式中运算符与运算对象的类型不符，如：布尔型量参与算术运算，整型值参与逻辑运算等；

（c）控制流语句中的控制条件 id 不是布尔型，或者在引用时没有确定的值；

1. 输入输出

【输入】一个名字类似 source.txt 的纯文本文件，内容是字符串形式的源程序。

【输出】源程序中是否存在错误的提示。可以写入输出文件或打印到屏幕。

1. 实现细节
2. 建立符号表，存储变量的性质，用于类型判断，检查变量是否声明和初始化；

逻辑结构采用线性表，将语义分析中需要的信息写入表中；

typedef struct node{

    char id[MAXSIZE];  //标识符

    int type;          //类型 1 means "int", 0 means "bool"

    int value;         //常数值

    int is\_valued;     //valued or not,1 means values,0 means invalued.

}SYMBOL;//符号表的基本元素

B.实现方法：语法制导翻译

1. 属性文法
2. 继承属性

宜采用自顶向下语法分析，采用前缀遍历，适用于类型的声明decls();

1. 综合属性

宜采用自下而上语法分析，采用中缀遍历，适用于子表达式和运算符的检查。Eg. EXPRESSION 和 BOOL\_EXPRESSION

typedef struct tree{

    struct tree \* left;

    struct tree \* right;

    struct tree \* third;

    char value[20];

    int token;             //类别id

    int inh;               //继承属性 1 means "int", 0 means "bool"

    int sys;               //综合属性 1 means "int", 0 means "bool"

}TREE;

1. **代码翻译**
2. 主要功能

对语法树进行一次遍历，生成类汇编代码

1. 源语言与目的语言

源语言采用的是类C的littleC；

目的语言采用的是四元组

（operater，src1，src2，result）；

operater代表操作符，src1代表操作数1，src2代表操作数2，result代表此次运算的结果，一般用标识符表示；

typedef struct instruction

{

    char operater[10]; // 操作符

    char src1[20];

    char src2[20];

    char result[10];

} INST;

1. 实现细节

代码按照前序遍历语法树的顺序进行翻译

PC表示指令在文件中的行号；

对于跳转指令，采取的方法是：先记住该指令的行号R，等到跳转地址计算出来之后，该指令写入文件的第R行

而对于其他指令，采取的方法则是：

一旦获取到指令的全部元素，便立即写入下一行，所以行号的递增直接放到函数printf\_中实现

void printf\_(INST \*inst)

{

    pc++;

    if(fseek(wf,44\*pc,SEEK\_SET))

    return;

    if (inst != NULL)

        fprintf(wf, "%3d (%3s %10s %10s %10s)\n", pc, inst->operater, inst->src1, inst->src2, inst->result);

    fseek(wf, 0, SEEK\_END);

}

void printf\_to(INST \*inst, int row)

{

    if(fseek(wf,44\*row,SEEK\_SET))

    return;

    if (inst != NULL)

        fprintf(wf, "%3d (%3s %10s %10s %10s)\n", row, inst->operater, inst->src1, inst->src2, inst->result);

    fseek(wf, 0, SEEK\_END);

}

1. **语义分析器**

写入文件的第R行

写入文件的第R行

1. 题目说明

设计一个程序，针对选定的源语言，从输入文件中得到源程序的语法树，对其依次做类型检查和代码翻译。若源程序中存在类型错误，则报错，退出分析。若源程序不含类型错误，则将其翻译成类汇编代码并输出。程序应将所有结果信息写入输出文件。

2. 输入输出

【输入】一个名字类似 parse\_out.txt 的纯文本文件，内容是一棵完整的语法分析树或其抽象表示，它表达了一个源程序的语法结构。

【输出】一个名字类似 semantic\_out.txt 的纯文本文件，内容要么是对源程序中存在类型错误的提示，要么是一串类汇编代码，它与源程序语义等价。