# 实验 6：语义检查与中间代码生成实验

### 一、实验目的

通过设计、开发一个高级语言的语义检查和中间代码生成程

序，加深对相关课堂教学内容，包括语法制导翻译技术、类型确

定、类型检查、常见可执行语句（如赋值语句、条件语句、循环

语句）翻译技术的理解。

### 二、实验要求

(1)了解语义检查和中间代码生成的目的和意义；

(2)掌握语义检查和中间代码生成的一般内容；

(3)掌握语法制导翻译技术（特别是针对S-属性定义的自底向上翻

译方法和针对L-属性定义的深度优先翻译方法）；

(4)掌握根据翻译目标编制语义子程序的方法（可模仿教材中的语

义规则进行设计）。

三、实验原理

语义分析是任何编译程序必不可少的一个阶段，在整个编译过

程中，词法分析和语法分析是对源程序形式上的识别和处理，而

语义分析程序是对源程序的语义做相应的处理工作。中间代码生

成不是编译器的必须阶段，生成中间代码的目的是为了便于优化

和移植。

语义分析和中间代码生成的主流技术是语法制导翻译技术。语

法制导翻译的基本思想：为每个产生式配上一个语义子程序，（该

子程序描述了一个产生式所对应的翻译工作。这些工作包括：生

成中间代码，查填有关的符号表，检查和报错，修改编译程序某

些工作变量的值等）。在语法分析过程中，每当一个产生式用于

推导（自顶向下分析）或归约（自底向上分析）时，就调用该产

生式所对应的语义子程序，以完成既定的翻译任务。

### 四、实验步骤

(1)熟悉S语言的语义，了解语义分析阶段需要进行哪些语义检查

和需要对哪些可执行语句进行翻译；

(2)分析S语言的文法，必要的话可对文法进行改写（如增加标记

非终结符号并为其添加产生式）；

(3)为相应产生式编写语义子程序（模仿教材中的语义规则进行设

计），实现对运算及运算分量进行类型检查（运算的合法性与

运算分量类型的一致性或相容性）、变量定义的唯一性检查等

语义检查工作；

(4)理解掌握变量的中间代码、表达式的中间代码、语句的中间代

码的结构（见教材）；

(5)编写语义子程序（模仿教材中的语义规则进行设计），实现S

语言中各类语句（赋值语句、条件语句、循环语句）的翻译工

作，将源程序翻译为中间代码。

1. 撰写实验报告。

### 实现代码

#include<stdlib.h>

#include<fstream>

#include<iostream>

using namespace std;

#define MAX 100

int m=0,sum=0;//sum用于计算运算符的个数

//m用于标记输入表达式中字符的个数

char JG='A';//代表TEMP变量的名字

char str[MAX];//用于存输入表达式 也就是输入表的内容

int token=0;//左括号的标志

/\*\*\*\*\*\*\*\*\*\*\*用于更改计算后数组中的值\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//e 是运算表达式的起点 f是终点 如 1+2 从1的下标 到 2 的下标

void change(int e)

{

int f=e+2;

char ch=str[f];

if(ch>='A'&&ch<='Z')

{

for(int l=0;l<m+10;l++)

{

if(str[l]==ch)

str[l]=JG;

}

}

if(str[e]>='A'&&str[e]<='Z')

{

for(int i=0;i<m;i++)

{

if(str[i]==str[e])

str[i]=JG;

}

}

}

//运算表达式的起点和终点的下标 i m

void chengchuchuli(int i,int m)

{

//如 3\*b 这里 i的下标是\* i-1 是 3 i+1 是 b

i++;

for( ;i<=m-1;i++)//处理乘除运算

{

if(str[i]=='\*'||str[i]=='/')

{

cout<<"("<<str[i]<<" "<<str[i-1]<<" "<<str[i+1]<<" "<<JG<<")"<<endl;

change(i-1);

str[i-1]=str[i]=str[i+1]=JG;//把这个表达式的所有成员置成TEMP的值 方便下次来找

sum--;

JG=JG+1;

}

}

}

void jiajianchuli(int j,int m)

{

j++;

for( ;j<=m-1;j++)//处理加减运算

{

if(str[j]=='+'||str[j]=='-')

{

cout<<"("<<str[j]<<" "<<str[j-1]<<" "<<str[j+1]<<" "<<JG<<")"<<endl;

change(j-1);

str[j-1]=str[j]=str[j+1]=JG;

sum--;

JG=JG+1;

}

}

}

void scan(FILE \*fin)

{

int p[MAX];//这个数组用来存遇到的左括号下标

char ch='a';

int c=-1,q=0;//q记录右括号的位置

while(ch!=EOF)

{

ch=getc(fin);

//消除空格和换行符

while(ch==' '||ch=='\n'||ch=='\t')

ch=getc(fin);

str[m++]=ch;

//记录碰到的符号 符号数

if(ch=='='||ch=='+'||ch=='-'||ch=='\*'||ch=='/')

sum++;

else if(ch=='(')

{

p[++c]=m-1;//记录左括号的位置

}

else if(ch==')')

{

//先处理乘除再处理加减

q=m-1;//右括号位置

chengchuchuli(p[c],q);//从左括号处理到右括号

jiajianchuli(p[c],q);

JG=JG-1;

str[p[c]]=str[m-1]=JG;

c--;

JG=JG+1;

}

}

}

void four()

{

for(int i=0;i<=m-1;i++)//处理乘除运算

{

if(str[i]=='\*'||str[i]=='/')

{

cout<<"("<<str[i]<<" "<<str[i-1]<<" "<<str[i+1]<<" "<<JG<<")"<<endl;

change(i-1);

str[i-1]=str[i]=str[i+1]=JG;

sum--;

JG=JG+1;

}

}

for(int j=0;j<=m-1;j++)//处理加减运算

{

if(str[j]=='+'||str[j]=='-')

{

cout<<"("<<str[j]<<" "<<str[j-1]<<" "<<str[j+1]<<" "<<JG<<")"<<endl;

change(j-1);

str[j-1]=str[j]=str[j+1]=JG;

sum--;

JG=JG+1;

}

}

for(int k=0;k<=m-1;k++)//处理赋值运算

{

if(str[k]=='=')

{

JG=JG-1;

cout<<"("<<str[k]<<" "<<str[k+1]<<" "<<" "<<" "<<str[k-1]<<")"<<endl;

sum--;

change(k+1);

str[k-1]=JG;

}

}

}

int main(){

char in[MAX]; //用于接收输入输出文件名

FILE \*fin; //用于指向输入输出文件的指针

cout<<"请输入程序文件名：";

cin>>in;

cout<<endl;

if ((fin=fopen(in,"r"))==NULL) //判断输入文件名是否正确

{

cout<<endl<<"打开词法分析输入文件出错!"<<endl;

}

cout<<"四元式如下："<<endl;

scan(fin);//调用函数从文件中读入表达式

four();

if(sum==0) cout<<"成功生成中间代码！"<<endl;

else cout<<"有错误！"<<endl;

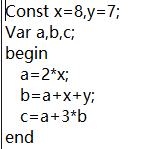
//关闭文件

fclose(fin);

return 0;

}

### 测试代码



### 结果展示

