**实验七　　自底向上语义分析**

1. 实验目的：加深对自底向上语法制导翻译技术的理解与掌握
2. 实验内容：针对LR分析法中所使用的文法，为其设计翻译方案，并利用该翻译方案，对所给程序段进行语法制导翻译，生成中间代码和输出符号表。
3. 实验方式

每位同学上机编程实现，指导教师现场指导答疑

1. 实验报告内容
2. 写出适合LR分析的翻译模式(主要)

addr 声明时给予的空间

type 实际的类型

width 类型的宽度

offset 相对总程序的偏移

name 以名字或者常量的形式方便打印

queue 初始化的队列

syn 综合属性，方便计算

inh 继承属性，单纯用来计算，最好是double型

tokentype token的类型

50程序🡪扩展声明 程序|空

51扩展声明🡪函数定义

52函数定义🡪类型定义 函数名(形参列表) 复合语句

53类型定义🡪char{类型定义.type=char,类型定义.width=1} | int{类型定义.type=int,类型定义.width=4} | float{类型定义.type=float,类型定义.width=8} | double{类型定义.type=double,类型定义.wirth=16}

54形参列表🡪形参声明 形参列表|空

55形参声明🡪类型定义 标识符

56声明序列🡪初始化声明 声明序列|空

57初始化声明🡪类型定义 直接声明体 初始化程序

{if(直接声明体.tokentype==ID)then enter(类型定义.type,name,程序.offset)，offset=offset+类型定义.offset,将初始化程序.queue中的值从表中的offset开始放入。直接声明体.type=类型定义.type

else offset,enter(类型定义.type,name,程序.offset) offset=offset+类型定义.offset\*直接声明体.offset, 将初始化程序.queue中的值从表中的offset开始放入。直接声明体.type=类型定义.type }

58直接声明体🡪标识符

{直接声明体.name=标识符.name,直接声明体.tokentype=ID, 直接声明.offset=0}

|数组名[基本表达式]

{直接声明体.name=标识符.name,直接声明体.tokentype=ARRAY,直接声明.offset=基本表达式.value}

59初始化程序🡪;{初始化程序.queue=empty}|={初始化序列};

60初始化序列🡪基本表达式91{初始化程序.queue.push(基本表达式.value)} 初始化序列|空

61初始化序列1🡪空

62复合语句🡪{复合语句体}

63复合语句体🡪声明序列 语句序列

64语句序列🡪语句 语句序列|空

65语句🡪复合语句|表达语句|分支语句|循环语句|跳转语句

66表达语句🡪;|表达式;

67分支语句🡪if语句 else语句

68循环语句🡪while(逻辑表达式) 复合语句|for(表达式; 逻辑表达式; 表达式) 复合语句

69跳转语句🡪continue;|break;|return 基本表达式;

70表达式🡪标识符=条件表达式

{gencode(标识符.addr=条件表达式.addr)}

|数组名[基本表达式]=条件表达式{gencode(数组[基本表达式]=条件表达式.addr)}|函数名(实参列表)

71条件表达式🡪逻辑表达式{gencode(条件表达式.addr=逻辑表达式.addr)}

72三目运算式🡪?表达式:表达式|空

73逻辑表达式🡪关系表达式 逻辑表达式1|not逻辑表达式 逻辑表达式1 | (逻辑表达式) 逻辑表达式1

74逻辑表达式1🡪逻辑符 关系表达式 逻辑表达式1|空

75逻辑符🡪 and|or

76关系表达式🡪因式 关系表达式1

77关系表达式1🡪关系操作符 因式 关系表达式1|空

78关系操作符🡪<|>|<=|>=|<>|=={关系操作符.tokentype}

79因式🡪(因式)因式1|项 因式1

80因式1🡪加减操作符 项 因式1|空

81项🡪后缀表达式 项1

82后缀表达式🡪基本表达式{后缀表达式.addr=基本表达式.addr，基本表达式.name=标识符.name }|数组名[基本表达式]{后缀表达式.addr=数组.offset+基本表达式.addr\*类型.width genCode(t1=addr\*width,t2=数组名.name+t1,addr.name=t2)}|函数名(实参列表)

83基本表达式🡪标识符|整数常量|小数常量|字符串常数|字符

{基本表达式.addr=addr,基本表达式.name=标识符.name}

84实参列表🡪基本表达式,实参列表|空

85加减操作符🡪+|-

86乘除操作符🡪\*|/|%

87 if语句🡪 if(逻辑表达式) 复合语句

88 else语句🡪else复合语句|空

89 项1🡪乘除操作符 后缀表达式{项1.inh=左项1.inh op 后缀表达式.addr，genCode(t)} 项1|空{项1.syn=项1.inh}

91 {初始化程序.queue.push(基本表达式.value)}

②设计存放中间代码的数据结构

③写出语义分析后生成的中间代码序列和符号表(主要)

④主要模块的算法描述

⑤实验中遇到的问题及解决方案

⑥本次实验完成情况及指导教师评语

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 独立完成 | 独立未完成 | 未独立 | 指导教师签字 | 特殊评语 |
|  |  |  |  |  |
| 实验中用到的特色方法及设计技巧 | | | | |