

语义分析和代码生成

一、目的和内容

1、实验目的：通过完成语义分析和代码生成程序，了解语义分析和代码生成与词法分析语法分析的关系，从而体会编译程序的全过程

2、实验内容：在语法分析的基础上，配有相应的代码生成语句使源程序生成相应的目标代码（汇编代码）

3、实验要求：输入附录 A 程序，理解其对附录 B: PASCAL 程序的语法制导翻译方案，从而生成目标代码附录 C。

二、程序设计语言的语义解释

1、程序的数据类型有二种：即 `interge` 和 `long` 型，和标准 PASCAL 一样用二个字节表示 `interge` 型，其表示范围为-32768~+32767，而 `long` 型用四个字节表示，其表示范围为

-2147483648~2147483647。

2、相同类型的数据运算其结果为该类型的数据，不同类型的数据运算其结果是多字节的类型，即 `interge` 和 `long` 型混合运算时结果为 `long` 型。

3、比较运算和 `ODD` 运算的结果是逻辑值，逻辑值也占二个字节用 0 表示假，-1 表示“真”

4、不同类型的数据进行赋值时，低字节赋给高字节时，自动补足，高字节赋给低字节时，高位部分自动忽略，而不考虑其类型不匹配问题。

5、运算符和优先级

运算符	优先级
#	0
(1
= < > <= >= <>	2
+ -	3
* /	4
- (单目) + (单目)	5
(6

6、输入语句输入数据用空格隔开，但输入数据个数不得少于 `read` 语句中的变量（否则获得随机值）

7、对于一个输出语句先自动换行，然后输出其值，对于同一个输出语句中的量用空格隔开

8、过程允许递归，允许带一个参数也可不带参数

9、其它语义规则同标准 PASCAL

三、语义子程序需考虑的问题

1、标识符表的设计

我们定义标识符有常量标识符、变量标识符和过程标识符。它们均有符号名，对于常量标识符和变量标识符需要有类型、运行时的地址和定义符号的层数（不同层的同名标识符是不同的对象）对于过程标识符它分有参过和无参过程，有参过程还需说明参数在符号表的的入口，故标识符表的结构为：（符号名，类型，运行时的相对地址，层，属性，参数入口）

- 2、由于 8088 指令系统中不提供四字节的乘法和除法指令以及对一于 integer 和 long 型的输入和输出，故把它们写成相应的子程序并加上一标志当调用相应的子程序时，就生成相应的子程序。
- 3、当 PROGRAM 匹配时，输出汇编语言定义数据段、堆栈段和代码段的首标志
- 4、当 · 匹配时，输出数据段、堆栈段和代码段的结束标志
- 5、常量说明填写标识符表，分配运行时的地址，并产生运行时把常量放入相应地址的指令
- 6、变量说明填写标识符表，分配运行时的地址
- 7、过程说明对于过程标识符和形参标识符均填写填写标识符表，并统计层数和处理相应的子程序的其它部分，生成进入过程后的分配过程的存储空间，指示数据区的低部和顶部，拷贝区头向量的指令，还要注意因为我们采用的是一遍扫描，当处理过程的头时，尚不能计算出本过程式所需的存储空间，在这里产生一条调用求数据区长度的指令。在处理过程结束时，要产生返回分配数据区的指令，产生求出数据区长度的子程序。另外在处理本层过程结束后从标识符表中全部清除属于本层的标识符，以区分下一个并列的子程序的标识符
- 8、处理参数时将参数的入口地址填入过程标识符的参数入口中
- 9、在处理分程序时，要区分是 0 层分程序(即主程序)还是过程，其产生的代码略有不同，还要产生跳过过程代码的指令
- 10、赋值语句根据变量和表达式的类型产生对运算对象栈中存放的表达式的临时变量地址中取值存入到相应的变量的指令
- 11、过程调用语句如有参数应产生将参数的临时变量放入参数约定的单元、取新的数据区的低部，拷贝区头向量，和调用相应子程序等指令，
- 12、if 语句只要产生取出条件的逻辑值进行判断的指令为假，跳过相应的语句
- 13、while 语句先产生标号，然后根据条件判断的指令为假跳出循环，在 while 语句结束时产生一条无条件转移语句和假转出的标号。
- 14、read 语句先产生将符号串读入缓冲区的指令，然后对于 read 中的每一个变量调用相应的读数据的子程序。
- 15、write 语句先产生回车指令，然后对于 write 中的每个表达式调用输出子程序
- 16、在 (匹配是入运算符栈，并将其运算符的优先级改为 1
- 17、在) 匹配计算在运算符栈中的全部优先级大于 (的运算符然后 (退栈
- 18、对于表达式中的常量、常量标识符、变量标识符匹配时，进入运算对象栈。
- 19、对于运算符 +, -, *, /, + (单目), - (单目), =, <, >, <=, >=, <> 生成运算符栈中优先级大于等于本运算符优先级的指令，然后将本运算符入栈。

四、语义子程序的实现方法

我们采用的方法是在递归子程序识别法的基础上，按制导翻译的原理加上相应的语义动作，由于识别方法本身也是程序逻辑，故在识别程序中直接插入语义分析和代码生成程序代码即可。