

# 实验六 语义分析和中间代码生成

## 一、实验目的与任务

通过设计一个简单的语义分析和中间代码生成器，加深对语法制导翻译方法的理解，掌握将语法分析所识别的语法范畴变换为中间代码的语义翻译方法。

## 二、实验要求

采用递归下降语法制导翻译方法，构造语义分析和中间代码生成器，实现对算术表达式、赋值语句、条件语句、循环语句进行语义分析，生成四元式中间代码的序列。

## 三、实验原理

(1) 待分析的语言为 C 语言的一个子集，其文法用扩充的 BNF 描述如下：

程序	→	main( ) 语句块
语句块	→	{ '语句串' }
语句串	→	语句 { ; 语句 } ;
语句	→	赋值语句   条件语句   循环语句
- 赋值语句	→	ID = 表达式
- 条件语句	→	if(条件) 语句块
- 循环语句	→	while(条件) 语句块
- 条件	→	表达式 关系运算符 表达式
- 表达式	→	项 { + 项   - 项 }
项	→	因子 { * 因子   / 因子 }
因子	→	ID   NUM  (表达式)
关系运算符	→	<   <=   >   >=   ==   !=

上面语句块产生式中的{和}是终结符号，加引号以示区别。

(2) 通过为上面文法中的每一个非终结符构造一个递归子程序，设计一个递归下降分析器。

(3) 在条件、表达式、赋值语句、条件语句、循环语句的递归子程序中，加入语义分析和生成四元式的代码。

构造下面的语义过程：

int gen(op,arg1,arg2,result): 该函数是将四元式(op, arg1,arg2,result) 送到四元式表中。

char \* newtemp(): 该函数回送一个新的临时变量名，临时变量名产生的顺序为 T1, T2.....

int merg(p1,p2): 该函数将以 p1 和 p2 为头指针的两条四元式链合并为一条链，合并后的链首为返回值。

int bp(p,t): 该函数的功能是把指针 p 所链接的每个四元式的第四区段都填为 t。

## 四、实验的输入和输出

输入是语法分析提供的正确的单词串，输出是四元式序列。

例如，对于语句串

```
i = 2 * 3 + 4;  
if ( i > 8 ) j = 10;  
while ( j > 0 ) {  
    k = k + 1;  
    j = j - 1;  
}
```

输出的四元式序列如下：

1. (\*, 2, 3, T1)
2. (+, T1, 4, T2)
3. (=, T2, , i)
4. (j >, i, 8, 6)
5. (j, , , 7)
6. (=, 10, , j)
7. (j >, j, 0, 9)
8. (j, , , 14)
9. (+, k, 1, T3)
10. (=, T3, , k)
11. (—, j, 1, T4)
12. (=, T4, , j)
13. (j, , , 7)
14. ....