

实验 1 词法分析程序

一、实验目的

构造 Training 语言的词法分析程序,程序要求能对输入的字符串流进行词法分析。在实验的过程中,学会应用单词分析的方法——NFA(非确定有穷自动机)和 DFA(确定有穷自动机),加深对词法分析原理的理解。

二、实验内容

了解 Training 语言的基本词法,并为之构造一个不确定有穷自动机 NFA,并将其转换成确定有穷自动机 DFA;然后依据此 DFA 编写 Training 语言的词法分析程序。

三、实验参考

(一) Training 惯用的词法

(1)下面是 Training 语言的关键字。

function if then while do endfunc

所有的关键字都是保留字,并且必须是小写。

(2)下面是专用符号。

= + - * / < <= ! = > >= == ; () #

(3)其他单词是标识符(id)和整型常数(num),通过下列正规式定义。

id = letter(letter| digit) *

num = digit digit *

letter = a | ... | z | A | ... | Z

digit = 0 | ... | 9

小写和大写字母是有区别的。

思考:构造实数的正规表达式,自己实现对实数的识别及表示。

(4)空格由空白、换行符和制表符组成。空格一般用来分隔 id、num、运算符和关键字,词法分析阶段通常被忽略。

(5)各种单词符号对应的种别码如附表 1 所示。

(二)词法分析程序的功能

(1)输入为所给文法的源程序字符串。

(2)程序的输出形式为单词串的输出形式。

所输出的每一单词,均按形如(syn, token 和 sum)的二元式编码。其中, syn 为单词种别码; token 为存放的单词自身字符串; sum 为整型常数。

(3)测试源程序片断。

附表 1 单词符号对应的种别码

单词符号	种别码	单词符号	种别码
function	1	=	18
if	2	<	20
then	3	<=	21
while	4	!=	22
do	5	>	23
endfunc	6	>=	24
Letter(letter digit) *	10	==	25
digit digit *	11	,	26
+	13	(27
-	14)	28
*	15	#	0
/	16		

function

 x= 9;

 if x> 0 then

 x= 2 * x+1/3;

 endfunc

 #

(4)输出结果。

 (1,function)

 (10,'x')

 (18,=)

 (11,9)

 (26,;)

 (2,if)

 ...

(三)算法设计思想见课本

实验2 递归下降语法分析

一、实验目的

构造文法的语法分析程序,要求采用递归下降语法分析方法对输入的字符串进行语法分析,实现对词法分析程序所提供的单词序列的语法检查和结构分析,进一步掌握递归下降的语法分析方法。

二、实验内容

编写为一上下文无关文法构造其递归下降语法分析程序,并对任给的一个输入串进行语法分析检查。程序要求能对输入串进行递归下降语法分析,能判别程序是否符合已知的语法规则,如果不符合(编译出错),则输出错误信息。

三、实验要求

利用C语言编制递归下降分析程序,并对 Training 语言进行语法分析。

(1)待分析的 Training 语言的语法。

用扩充的 BNF 表示如下。

①<程序>::=function<语句串>endfunc

②<语句串>::=<语句>{;<语句>}

③<语句>::=<赋值语句>

④<赋值语句>::=ID=<表达式>

⑤<表达式>::=<项>{+<项>|-<项>}

⑥<项>::=<因子>{*<因子>|/<因子>}

⑦<因子>::=ID|NUM|(<表达式>)

(2)实验要求说明。

输入单词串以“#”结束,如果是文法正确的句子,则输出成功信息,打印“success”,否则输出“error”,具体例子如下。

输入

```
function
    a= 9;
    x= 2 * 3;
    b= a+ x;
endfunc
#
```

输出

```
success
```

又如,输入

```
x= a+ b * c
```

```
endfunc
```

```
#
```

输出

```
error
```

(3)语法分析程序算法思想见课本第 4 章相关内容。

(4)语法分析程序的 C 语言程序框架见课本。

实验 3 语义分析

一、实验目的

通过上机实习,加深对语法制导翻译原理的理解,掌握将语法分析所识别的语法成分变换为中间代码的语义翻译方法。

二、实验内容

定义模拟的简单语言的语义成分,将语义分析程序编制成一个子程序,在实验 2 分析出各语法单位后,分析其含义,并将可执行语句或表达式翻译为四元式输出,并将错误信息输出。

三、实验要求

采用递归下降语法制导翻译法,对算术表达式、赋值语句进行语义分析并生成四元式序列。

输入是语法分析后提供的正确的单词串,输出为三地址指令形式的四元式序列。

例如:对于语句串

```
function
```

```
    a= 2+ 3 * 4;
```

```
    x= (a+ b)/c;
```

```
endfunc
```

```
#
```

输出的三地址指令如下。

① $t1=3 * 4$

② $t2=2+t1$

③ $a=t2$

④ $t3=a+b$

⑤ $t4=t3/c$

⑥ $x=t4$

实现的相关代码见附录 B 语义分析源代码部分。

实验4 综合程序

一、实验目的

在前面各章节的构造编译程序的基本理论和实验基础之上,编写一个综合控制程序,实现一个完整的 Training 语言编译程序(包括符号表的构造,词法分析,语法分析,中间代码产生,目标代码生成等重要子程序),使学生将理论与实际应用结合起来,理解编译程序的整个过程。

二、实验内容

首先设计一个目标代码生成器,能完成指定寄存器个数的情况下将一中间代码程序段翻译成汇编语言目标代码(汇编指令应包括加、减、乘、除),要求指令条数最少的情况下,尽量使用寄存器,尽量少访问内存,这样才能做到运行效率高。

然后设计一个综合控制程序,使目标代码生成器与前面三个子程序成为一个完整的编译程序。