1. 词法分析

§2.1 词法分析器(lexical analyzer)的作用

一.作用

词法分析阶段的主要任务是读输入字符流,产生用于语法分析的记号序列.

\*词法分析器作为语法分析器(parser)的一个子程序或联立程序来实现,当收到来自分析器的“取下一个记号”的命令时,词法分析器读输入字符,直到它能确认下一个记号为止.然后,词法分析器将该记号返回给语法分析器.

词法分析器的几个其它任务:

1.删去注释和无用空白;

2.把错误信息和源程序联系起来;

3.作预处理.

二.分离词法分析的理由

我们将词法分析阶段从语法分析阶段中分离出来的理由是:

1.简化设计是重要的原因;

2.编译器的效率会改进;

3.编译的可移植性加强.

三.记号﹑模式﹑单词

1.记号(token)

单词在编译程序内部的表示. 很多字符串有同样的记号输出.

2. 模式(pattern)

具有同一记号输出的字符串集合由叫做模式的规则来描述.

\*模式能匹配该集合的任一字符串.

3. 单词(lexeme)

是源程序中具有逻辑含义的极小字符串.

4. 例子 (见书 P69, P112)

\*我们把记号作为源语言文法的终结符.

5.记号包括: 关键字﹑算符﹑标识符﹑常数﹑文字串和标点符号, 如括号﹑逗号和分号.

1. 记号的属性(attribute)

由于一个记号代表不只一个单词,必须提供附加信息,区别对应同一记号的不同单词. 另外,编译中后面的阶段需要用到这些单词的属性.

1.记号的内部形式

(类码, 属性)

通常类码标识不同的记号, 属性为指向该标识符在符号表中条目的入口的指针.对常数来说,则是常数的值.

2. 例子: FORTRAN中

E=M\*C\*\*2

输出: (见书 P70, P113)

§2.2 输入缓冲区(Input buffering)

\*由于词法分析器是编译器中逐个读源程序字符的唯一阶段,编译的相当可观的时间消耗在词法分析阶段.所以加快词法分析是编译器设计需要关心的问题.

1. 缓冲区对

\*设两半缓冲区,每一半是4096字节,即磁盘的一块

\*两个指针

\*工作原理

\*向前扫描的情况

例: FORTRAN语言中

DO 5 I=1.25

DO 5 I=1,25

1. 标记

\*缓冲区结束标记

\*文件尾标记

\*工作原理

§2.3 记号的说明

1. 串和语言

1.字母表(alphabet): Σ 非空﹑有穷集

2.串(string): 是字母表中符号的有穷序列. 也叫句子或字.

例如: Σ={a, b},

串: s=abb, 串长 |s| = 3

空串: ε , |ε| = 0

3.语言(language): 字母表上串的集合

例如: Σ={a, b}, L= {a, b, aab, abb, ba}

空语言:φ, 只含空串的语言: {ε}

4. 连接运算(concatenation):

设 x = ab, y = bb, 则 xy = abbb

 x, εx = xε= x

指数(exponentiation): s0 =ε, si+1 = sis , 其中s是任意串.

1. 语言的运算(Operations on languages)

设L﹑M为字母表Σ上的语言:

1.L∪M = {x | x∈L或x∈M} (union)

LM = {xy | x∈L且y∈M} (concatenation)

指数: L0 = {ε}, Li+1 = LiL (exponentiation)

L\* =  ((Kleene) closure)

L+ =  (positive closure)

2.例子:

设L={A,B,…,Z,a,b,…,z}, D={0,1,2,…,9}

例1: L∪D

例2: LD

例3: L4

例4: L\*

例5: L(L∪D)\*

例6: D+

\*阅读: 教材 3.1, 3.2, 3.3.1, 3.3.2 节

作业1：

1. 将下列程序的字符序列转换为记号序列,并给每个记号以合理的属性值.

PASCAL语言：

Function max(i,j:integer):integer;

/\* return maximum of integers i and j \*/

begin

if i > j then max := i

else max := j

end;