

第三章:词法分析

- 3.1 词法分析的功能
 - 一一识别单词,返回单词的类别和值
- 3.2 词法分析程序的设计与实现
 - 一一状态图: 有穷自动机的非形式表示
 - ---正则文法和状态图





会回何头八江首 今圆处水月池年 **乘昨是棺十畔**~ 清风东南五杏月 光一南西庭园十 似吹见北隘边五 往无月望浦今乘 年人几乡沙年曲



八月十五日夜湓亭望月 白居易

昔年八月十五夜,曲江池畔杏园边。 今年八月十五夜,湓浦沙头水馆前。 西北望乡何处是,东南见月几回圆。 昨风一吹无人会,今夜清光似往年。



八月十五日夜湓亭望月 白居易

昔年八月十五夜,曲江池畔杏园边。 今年八月十五夜,湓浦沙头水馆前。 西北望乡何处是,东南见月几回圆。 昨风一吹无人会,今夜清光似往年。

词法分析: 断词,并给出词性(分类)

语法分析: 断句,并给出句的结构、分类





程序

```
int main() '\n' { int count=read(); '\n'
//if number of entries read is greater
than 1 '\n' //then sort() and compact()
'\n' if (count > 1) { sort();
compact(); } '\n' if (count ==0) '\n'
count << "no sales for this month\n";
'\n' else write(); '\n' return; '\n' }</pre>
```





程序

```
int main()
  int count=read();
  //if number of entries read is greater than 1
 //then sort() and compact()
  if (count > 1) { sort(); compact(); }
  if (count == 0)
    count << "no sales for this month\n";
  else write();
  return;
```





内容

- 3.1 词法分析程序的功能及实现方案
- 3.2 单词的种类及词法分析程序的输出形式
- 3.3 正则文法和状态图
- 3.4 词法分析程序的设计与实现





3.1 词法分析程序的功能及实现方案

ca 词法分析程序的功能

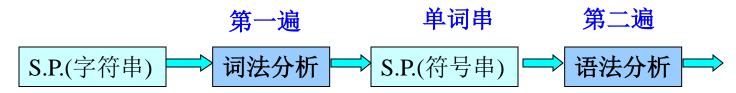
- ◆ 词法分析: 根据词法规则识别及组合单词,进行词法检查。
- ◆ 对数字常数完成数字字符串到二进制数值的转换。
- ◆删去空格字符和注释。





实现方案: 基本上有两种

1.词法分析单独作为一遍

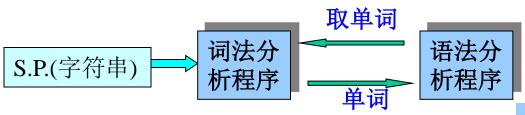


优点:结构清晰、

各遍功能单一

2.词法分析程序作为单独的子程序

缺点:效率低



优点:效率高



3.2 单词的种类及词法分析程序的输出形式

单词的种类

1. 保留字: begin、end、for、do...

2. 标识符: 由用户定义,表示各种名字的字符串

3. 常数: 无符号数、布尔常数、字符串常数等

4. <mark>分界符</mark>: +、-、*、/、...





表示单词的种类,可用整数编码或记忆符表示

词法分析程序的

部形式

不同的单词不同的值

单词类别	单词值
整型	58
保留字	"for"

几种常用的单词内部形式:

- 1、按单词种类分类
- 2、保留字和分界符采用一符一类
- 3、标识符和常数的单词值又为指示字(指针值)





1、按单词种类分类

类别编码 单词值

单词名称	类别编码	单词值
标识符 无符号常数(整) 无符号浮点数 布尔常数 字符串常数 保留字 分界符	1 2 3 4 5 6	内部字符串整数值数值 数值 0或1 内部字符串 保留字或内部编码 分界符或内部编码
, , , , ,	,	24 21 14 -241 4 El. Alida . 4





2、保留字和分界符采用一符一类

单词名称	类别编码	单词值
 标识符	1	内部字符串
无符号常数(整)	2	整数值
无符号浮点数	3	数值
布尔常数	4	0或1
字符串常数	5	内部字符串
BEGIN	6	-
END	7	-
FOR	8	-
DO	9	-
•••••	•••••	
:	20	-
+	21	-
*	22	-
,	23	-
(•••••	



内容

- 3.1 词法分析程序的功能及实现方案
- 3.2 单词的种类及词法分析程序的输出形式
- 3.3 正则文法和状态图
- 3.4 词法分析程序的设计与实现





3.3 正则文法和状态图

• 状态图的画法(根据文法画出状态图)

例如:正则文法

Z::= U0 |V1|

U := Z1 | 1

V ::= Z0 | 0

左线性文法。该文法所定义的语言为:

$$L(G[Z]) = \{ B^n | n>0 \}, 其中 B = \{01,10\}$$





左线性文法的状态图的画法:

例:正则文法

Z := U0 | V1

U := Z1 | 1

 $V := Z0 \mid 0$

- 1. 令G的每个非终结符都是一个状态;
- 2. 设一个开始状态S;
- 3. 若Q::=T, Q ∈ Vn,T ∈ Vt,则:
- 4. 若Q::=RT, Q、R∈Vn,T∈Vt, 则:
- 5. 按自动机方法,可加上开始状态和终止状态标志。



R

Compiler

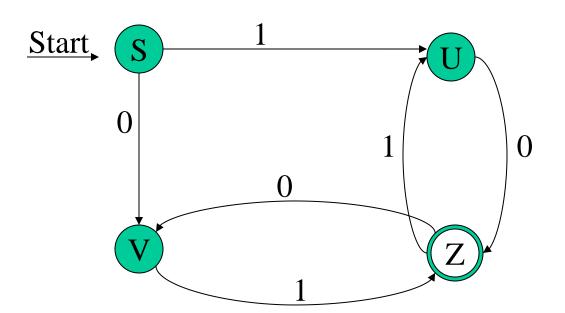
例如:正则文法

Z::=U0|V1

U := Z1 | 1

V ::= Z0 | 0

其状态图为:

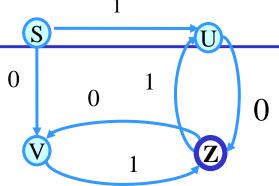


- 1. 每个非终结符设一个状态;
- 2. 设一个开始状态S;
- 3. 若Q::=T, Q \in Vn,T \in Vt,
- 4. 若 $Q::=RT, Q, R \in Vn, T \in Vt,$
- 5. 加上开始状态和终止状态标志





• 识别算法



利用状态图可按如下步骤分析和识别字符串x:

- 1、置初始状态为当前状态,<mark>从x的最左字符开始</mark>,重复步骤2,直到x右端为止。
- 2、扫描x的下一个字符,在当前状态所射出的弧中 找出标记有该字符的弧,并沿此弧过渡到下一个状态;如果找不到标有该字符的弧,那么x不是句子, 过程到此结束;如果扫描的是x的最右端字符,并 从当前状态出发沿着标有该字符的弧过渡到下一个 状态为终止状态Z,则x是句子。

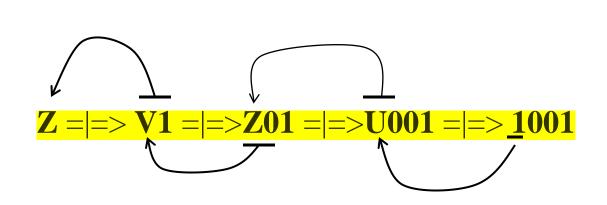
例: x=0110 和1011

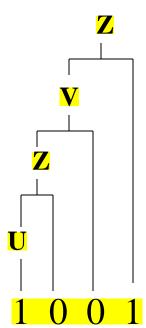




•问题:

- 1、上述分析过程是属于自底向上分析?还是自顶向下分析?
- 2、怎样确定句柄?









3.4 词法分析程序的设计与实现

词法规则 —— 状态图 —— 词法分析程序





3.4.1 文法及其状态图

语言的单词符号

标识符

保留字(标识符的子集)

无符号整数

单分界符 + *:,()

双分界符 :=

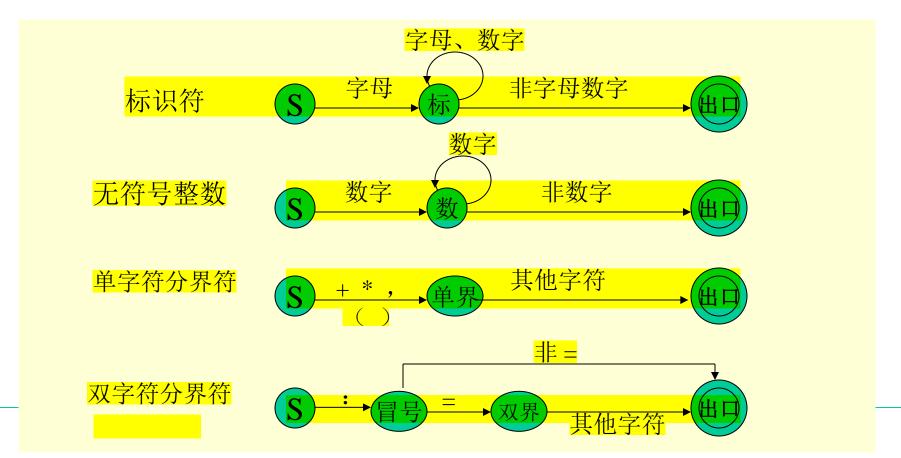
两点说明: 1. 空格的作用

2. 实数的表示

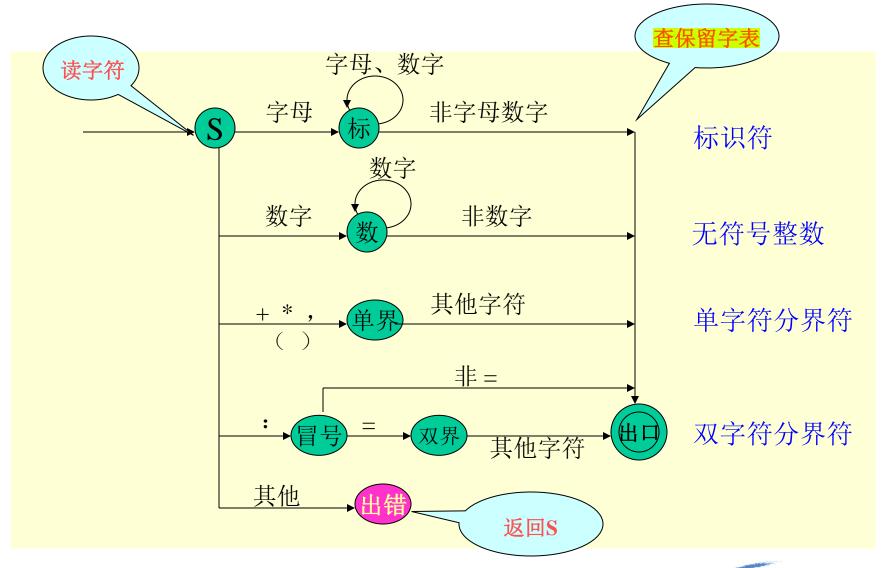




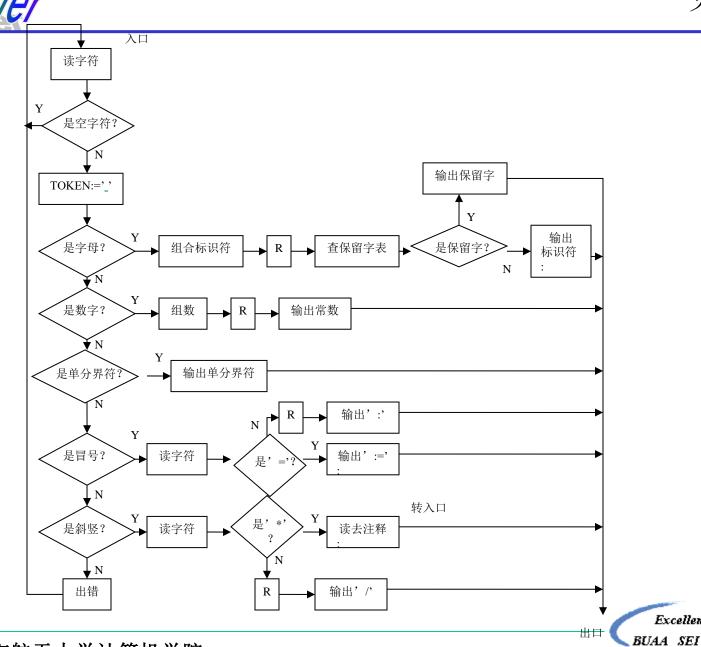
- 文法: 1. <标识符>::= 字母 | <标识符>字母 | <标识符>数字
 - 2. <无符号整数>::=数字 | <无符号整数>数字
 - 3. <单字符分界符>::=:|+|*|,|(|)
 - 4. <双字符分界符>::= <冒号>=
 - 5. <冒号>::=:







Excellence in





3.4.2 状态图的实现——构造词法分析程序

1. 单词及内部表示

2. 词法分析程序需要引用的公共(全局)变量和过程

3. 词法分析程序算法





1.单词及内部表示: 保留字和分界符采用一符一类

单词名称	类别编码	记忆符	单词值
BEGIN	1	BeginSym	-
END	2	EndSym	-
FOR	3	ForSym	-
DO	4	DoSym	-
\mathbf{IF}	5	IfSym	-
THEN	6	ThenSym	-
ELSE	7	ElseSym	-
标识符	8	IdSym	内部字符串
常数(整)	9	IntSym	整数值
:	10	ColonSym	-
+	11	PlusSym	-
*	12	StarSym	-
•	13	ComSym	-
(14	LparSym	-
)	15	RparSym	-
: =	16	AssignSym	-
			Excellence in 🛰



2.词法分析程序需要引用的公共(全局)变量和过程

名称

- char
- token
- getchar
- getNBC
- **CAT**
- ▶ IsLetter 和 IsDigit
- **▶** UnGetCH
- **RESERVE**
- ► ATOI
- **Error**

类别

字符变量 字符数组 读字符过程 过程

过程 布尔函数 过程 布尔函数

函数过程

功能

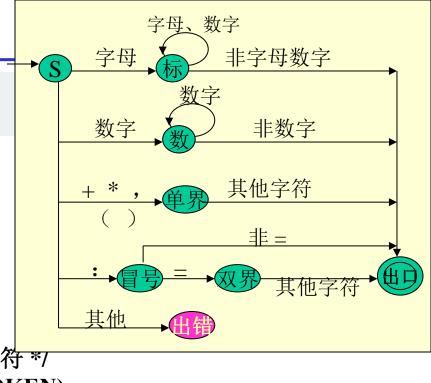
存放当前读入的字符 存放单词字符串 读字符到char,移动指针 反复调用getchar, 直至char进入一个非空白字符 char与token连接 判断 读字符指针后退一个字符 判断token中的字符串 是保留字,还是标识符 字符串到数字的转换 出错处理



Compiler

3、词法分析程序算法

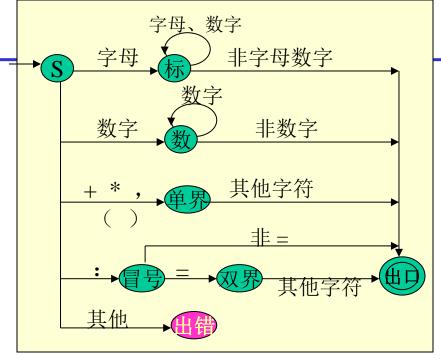
```
START: token := ''; /*置token为空串*/
      getchar; GetNBC;
 CASE char OF
 'a'..'z': BEGIN
         WHILE IsLetter OR IsDigit DO
             BEGIN CAT; getchar END;
         UnGetCH:
         C:= RESERVE; /* 返回0,为标识符 */
         IF C=0 THEN RETURN('IDSY': TOKEN)
         ELSE RETURN (C,-) /*C为保留字编码*/
       END;
'0'...'9': BEGIN
         WHILE IsDigit DO
              BEGIN CAT; getchar END;
           UnGetCH;
          RETURN ('INTSY', ATOI)
       END;
'+': RETURN('PLUSSY',-);
```





```
Compiler
```

```
'*': RETURN('StarSym',-);
    RETURN('ComSym',-);
    RETURN('LparSym',-);
')':
    RETURN('RparSym',-);
6.9.
    BEGIN
       getchar;
       if CHAR='=' THEN RETURN('AssignSym',-);
        UnGetCH;
       RETURN('ColonSym',-);
     END
END OF CASE;
ERROR;
GOTO START;
```



Excellence in BUAA SEI

练习:用C语言实现上述算法。



```
/*返回类别编码*、
int getsym()
        clearToken();
        while(isSpace()||isNewline()||isTab()) getchar(); /*读取字符, 跳过空格、换行和Tab*/
                           /*判断当前字符是否是一个字母*/
    if(isLetter())
        while(isLetter()||isDigit()) /*将字符拼接成字符串*/
         { catToken(); getchar();}
                            /*指针后退一个字符*/
        retract();
        int resultValue = reserver(); /*resultValue是查找保留字的返回值*/
        if(resultValue==0) symbol= IDSY; /*resultValue=0, token中的字符串为标识符*/
             else symbol= resultValue; /*否则token中的字符串为保留字*/
                            /*判断当前字符是否是一个数字*/
    else if(isDigit())
                              /*将字符拼接成整数*/
        while(isDigit())
         { catToken(); getchar();};
        retract();
                                 /*将token中的字符串转换成整数*/
        num= transNum(token);
                                         /*此时识别的单词是整数*/
        symbol= INTSY;
```



第三章作业:

P73 1, 2, 3 (词法见65-67页)

