第三章 词法分析

1. 词法分析的功能
2. 词法分析程序的设计与实现
   - 状态图
3. 词法分析程序的自动生成
   - 有穷自动机

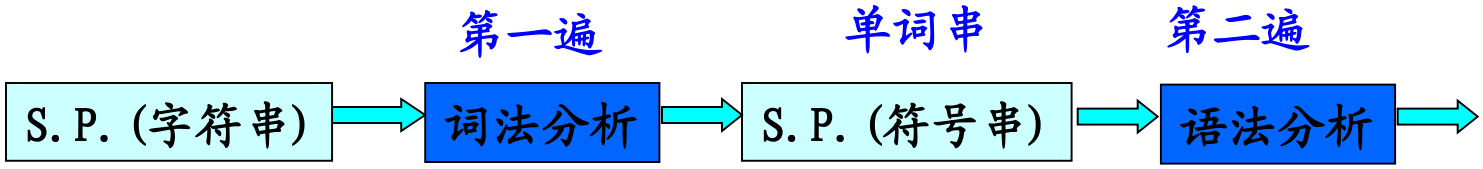
3.1 词法分析程序的功能及实现方案

➢ 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。
- ♦ 删去空格字符和注解。

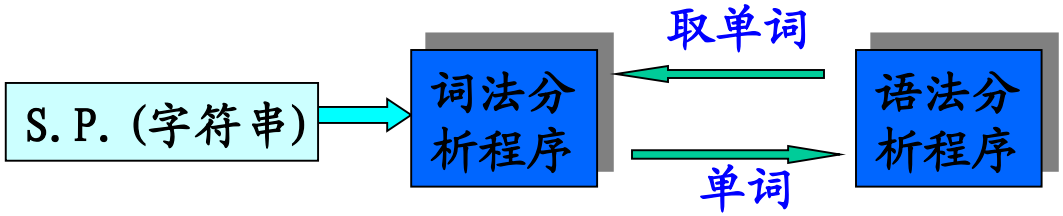
实现方案：基本上有两种

1. 词法分析单独作为一遍



优点： 结构清晰、各遍功能单一
缺点：效率低

2. 词法分析程序作为单独的子程序



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:
3. 常数：无符号数、布尔常数、字符串常数等
4. 分界符：+、-、*、/

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[illegible]部形式

不同的单词有不同的值

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可为指示字（指针值）

方案1、按单词种类分类

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
保留字	6	保留字或内部编码
分界符	7	分界符或内部编码

方案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	-
(	…….	--

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

L(G[Z]) = { B^n | n>0 }，其中 B= {01,10}

例：正则文法

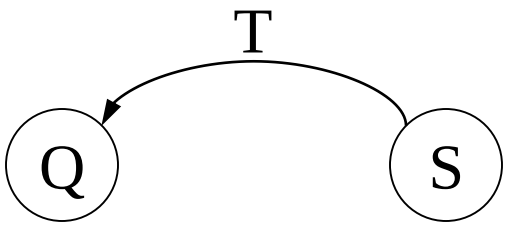
Z::= U0 |V1

U ::=Z1 |1

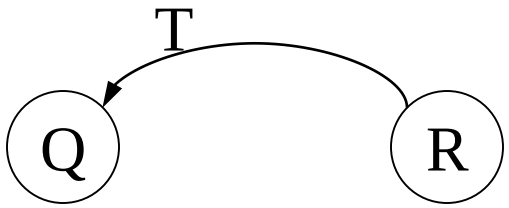
V ::=Z0 | 0

左线性文法状态图的画法:

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。

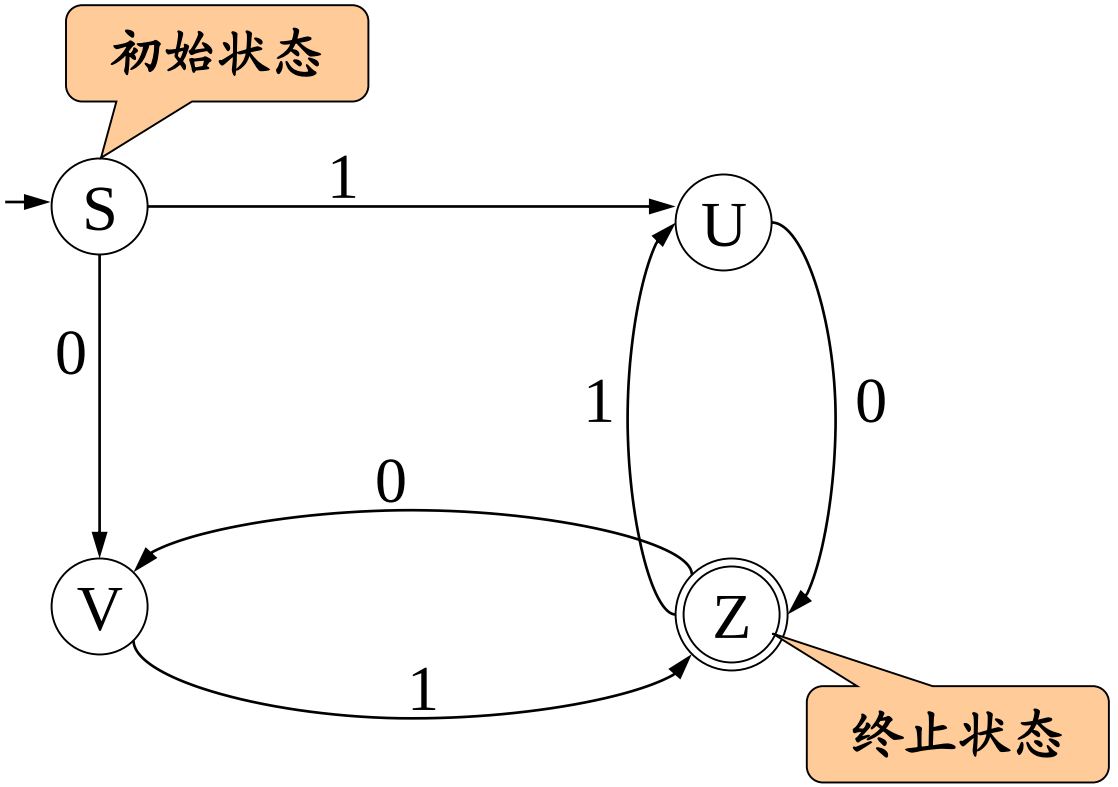
例如：正则文法

Z::= U0 | V1

U ::=Z1 | 1

V ::=Z0 | 0

其状态图为：



识别算法（自然语言描述）

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，从x的最左字符开始，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；

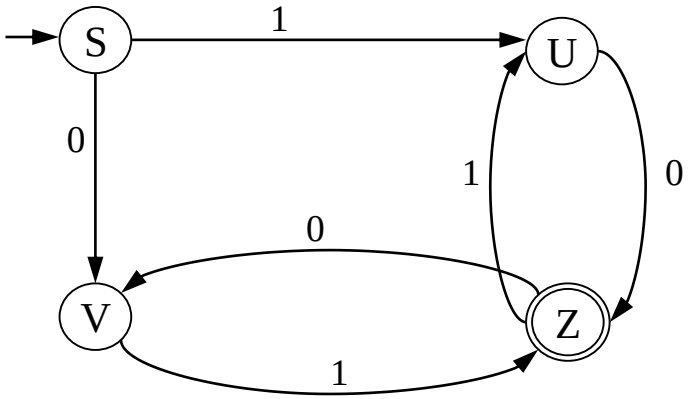
如果扫描的是x的最右端字符，并从当前状态出发沿着标有该字符的弧过渡到下一个状态为终止状态Z，则x是句子。

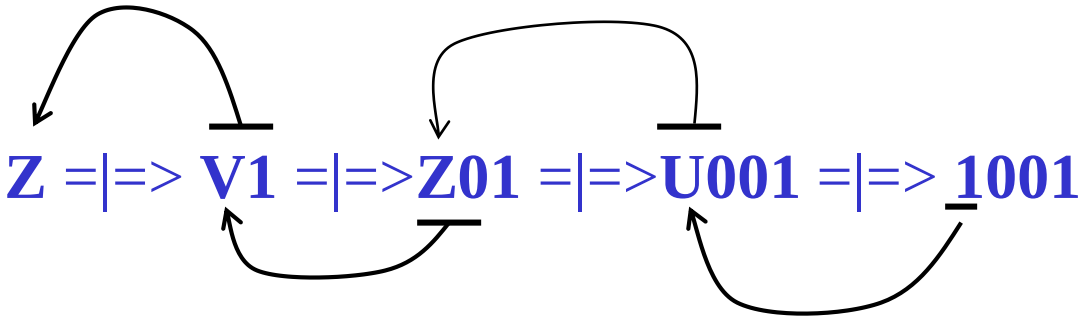
例： x=01101 和1011

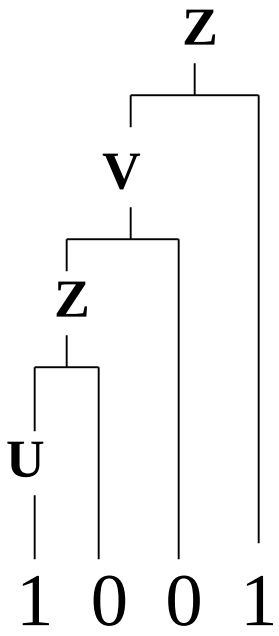
问题:

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、注释符号 /* 和 */ 以及 //。词法分析程序不输出注释!

2、各单词之间用空白符号（空格、制表、回车）分开。

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>=

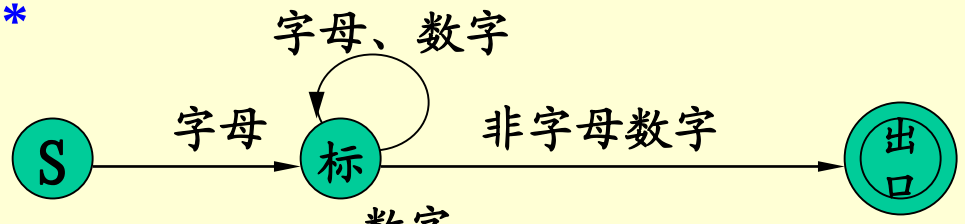
5. <冒号>::= ：

6. <注释头符号>::= <斜竖>*

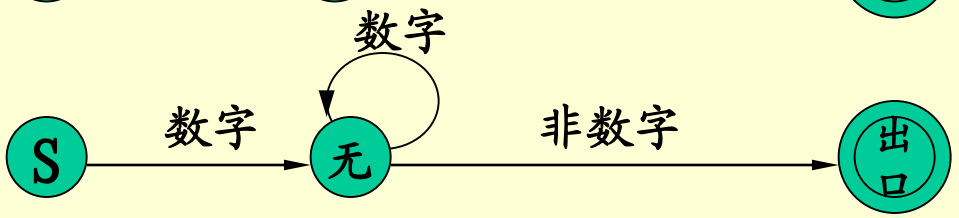
7. <斜竖>::= /

正则文法！

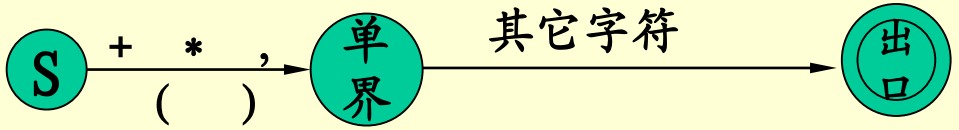
标识符



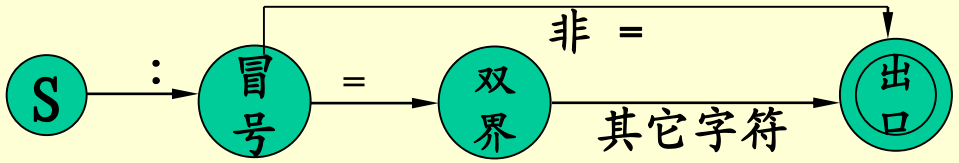
无符号整数



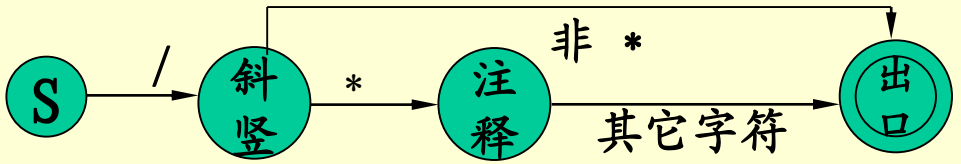
单字符分界符

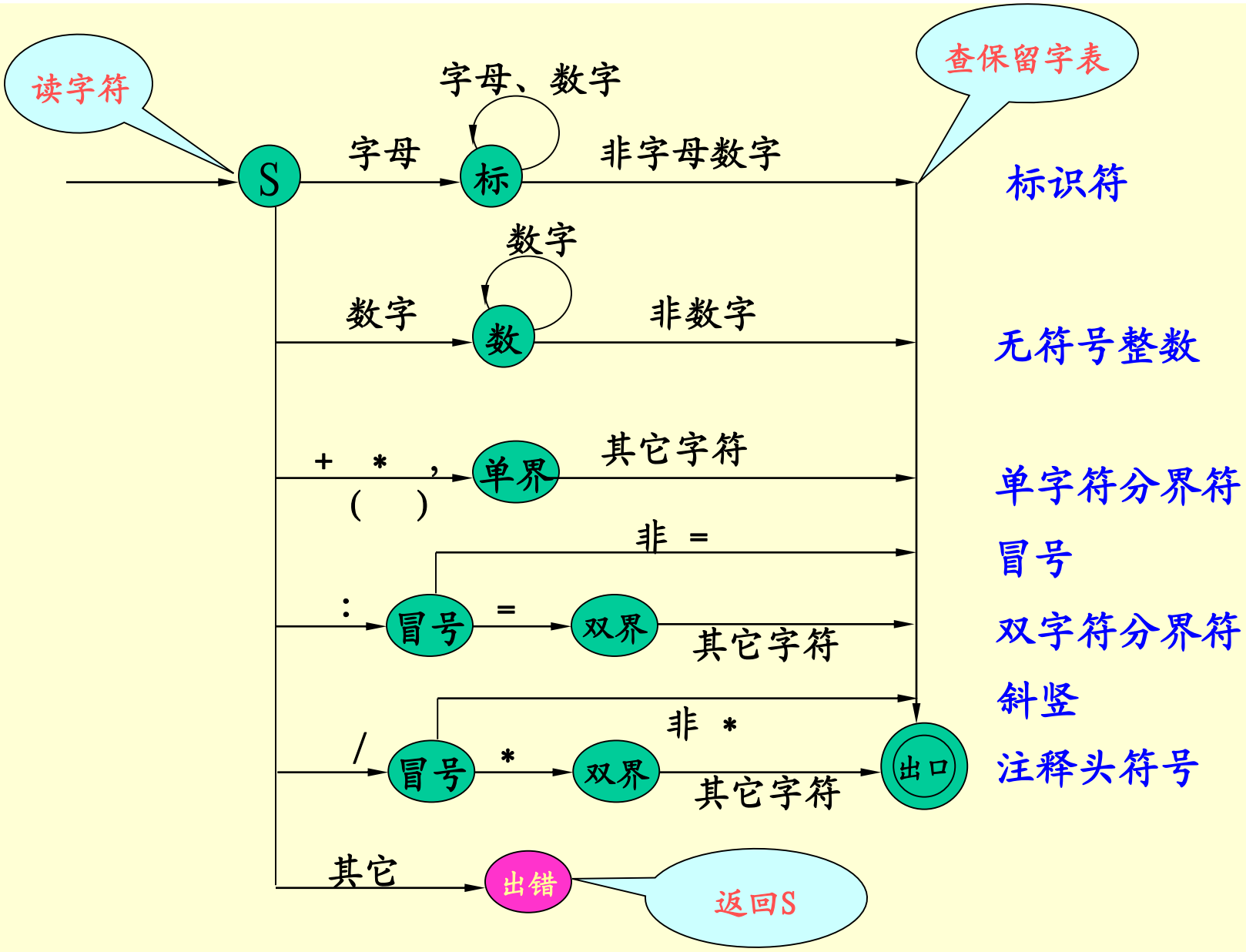


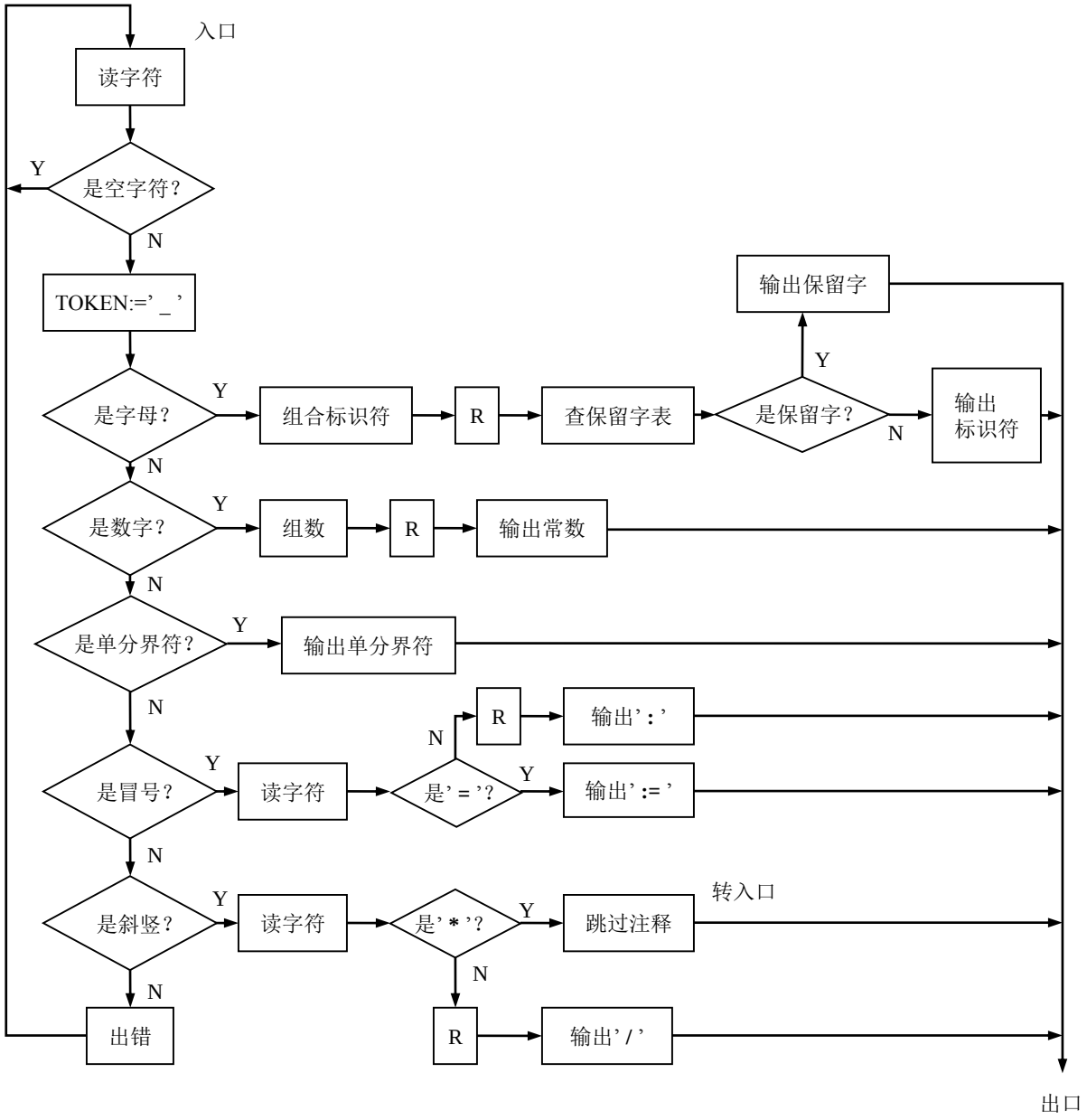
双字符分界符



注释头符号




读字符
S
字母
字母、数字
标
非字母数字
查保留字表
标识符
数字
数字
数
非数字
无符号整数
+ * ,
( )
单界
其它字符
单字符分界符
非 =
:
冒号
=
双界
其它字符
双字符分界符
斜竖
非 *
/
冒号
*
双界
其它字符
出口
注释头符号
其它
出错
返回S


入口
读字符
是空字符？
Y
N
TOKEN:=' _ '
是字母？
Y
组合标识符
R
查保留字表
是保留字？
Y
N
输出保留字
输出
标识符
是数字？
Y
组数
R
输出常数
N
是单分界符？
Y
输出单分界符
N
是冒号？
Y
读字符
是' = '？
N
Y
R
输出' : '
输出' := '
N
是斜竖？
Y
读字符
是' * '？
Y
N
跳过注释
转入口
N
出错
R
输出' / '
出口

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

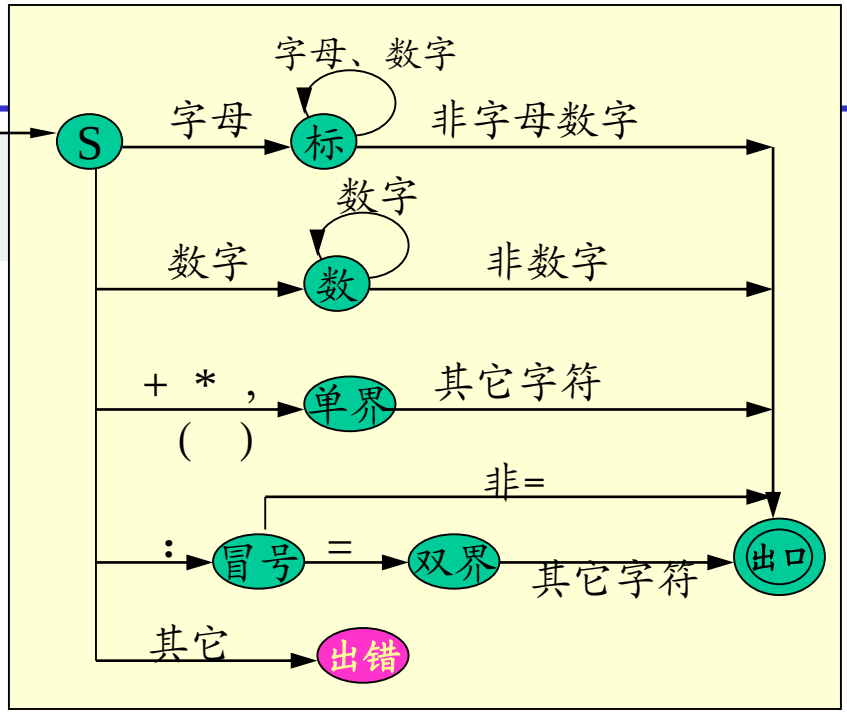
1.单词及内部表示：保留字和分界符采用一符一类

单词名称	类别编码	记忆符	单词值
BEGIN	1	BEGINSY	-
END	2	ENDSY	-
FOR	3	FORSY	-
DO	4	DOSY	-
IF	5	IFSY	-
THEN	6	THENSY	-
ELSE	7	ELSESY	-
标识符	8	IDSY	内部字符串
常数(整)	9	INTSY	整数值
:	10	COLONSY	-
+	11	PLUSSY	-
*	12	STARSY	-
,	13	COMSY	-
(	14	LPARSY	-
)	15	RPARSY	-
:=	16	ASSIGNSY	-

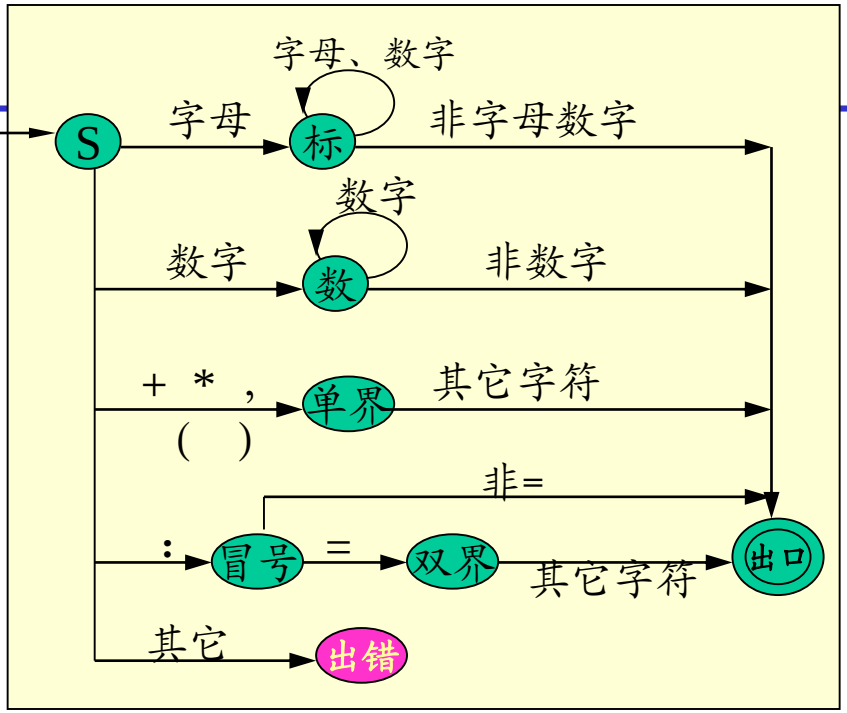
2.词法分析程序需要引用的公共（全局）变量和过程

名　称	类　型	功　能
CHAR	字符变量	存放当前读入的字符
TOKEN	字符数组	存放单词字符串
GETCHAR	读字符过程	读字符到CHAR，移动指针
GETNBC	过程	反复调用GETCHAR，直至CHAR进入一个非空白字符
CAT	过程	CHAR与TOKEN连接
ISLETTER 和 ISDIGIT	布尔函数	判断CHAR是字母还是数字
UNGETCH	过程	读字符指针后退一个字符
RESERVE	布尔函数	判断TOKEN中的字符串是保留字，还是标识符
ATOI	函数	字符串到数字的转换
ERROR	过程	出错处理

3、词法分析程序算法




```
START:  TOKEN := ' ';     /*置TOKEN为空串*/
                GETNBC; GETCHAR;
  CASE  CHAR OF
 'A'..'Z': BEGIN
                WHILE  ISLETTER OR ISDIGET DO
                        BEGIN CAT; GETCHAR;  END
                 UNGETCH;
                 C:= RESERVE;
                 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',-) ;
```




```
‘*’ :   RETURN(‘STARSY’,-) ;
‘,’ :   RETURN(‘COMMASY’,-) ;
‘(’ :   RETURN(‘LPARSY’,-) ;
‘)’ :   RETURN(‘RPARSY’,-) ;
‘:’ :   BEGIN
            GETCHAR;
            if  CHAR=‘=’ THEN  RETURN(‘ASSIGNSY’,-);
            UNGETCH;
            RETURN(‘COLONSY’,-) ;
        END
ERROR;
END OF CASE
GOTO START;
```

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

第三章作业：

P73：1、2、3（词法见65-67页）