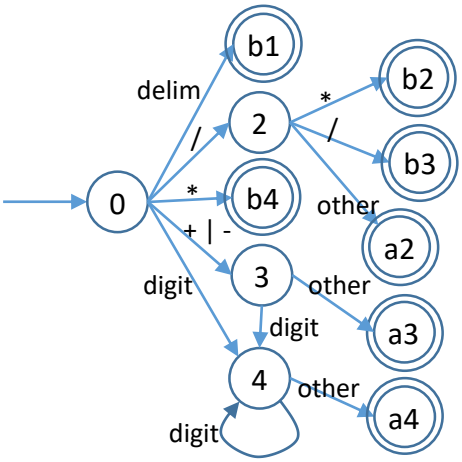
一、词法分析

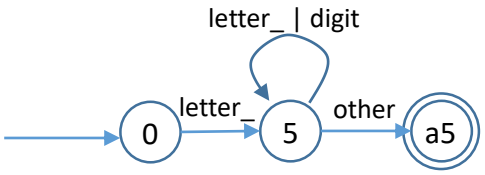
letter_→a..z | A..Z | _
digit → 0..9
reloop→ < | = | > | ! | & | '|'
calop→ + | - | * | /
id →letter_(letter_ | digit)*
number→digit digit*
delim→blank | tab | newline
ws→ | /* | */ | //
symbol→(|) | , | ; | { | }
赋值号=、非运算符!被归入关系符中分析，然后在细分关系符时再筛选出来。
关键字归入 id 中分析，然后再通过判断筛选出来。

relop 及 关系运算符

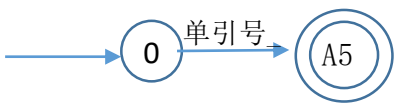
calop/delim:



id:



char:



	relop	delim	*	/	+ | -	digit	letter_	symbol
0	1	acc 201	acc 204	2	3	4	5	acc 205
1	acc 101	acc 101	acc 101	acc 101	acc 101	acc 101	acc 101	acc 101
2	acc 102	acc 102	acc 202	acc 203	acc 102	acc 102	acc 102	acc 102
3	acc 103	acc 103	acc 103	acc 103	acc 103	4	acc 103	acc 103
4	acc 104	acc 104	acc 104	acc 104	acc 104	4	acc 104	acc 104
5	acc 105	acc 105	acc 105	acc 105	acc 105	5	5	acc 105

✧ 10*表示接收状态，但多读了一个有用符号，需要缓存

✧ 20*表示接收状态，没有多读有用符号

词法符号 token

```
#define ERR             -1
#define SYN_NUM         1       // int 整数
#define SYN_ID          2       // id
#define SYN_LEETER      3       //char 型常量
#define SYN_LT          11      // <
#define SYN_GT          12      // >
#define SYN_LE          13      // <=
#define SYN_GE          14      // >=
#define SYN_EQ          15      // ==
#define SYN_NE          16      // !=
#define SYN_ADD         17      // +
#define SYN_SUB         18      // -
#define SYN_MUL         19      // *
#define SYN_DIV         20      // /
#define SYN_PAREN_L     21      // (
#define SYN_PAREN_R     22      // )
#define SYN_BRACE_L     23      // {
#define SYN_BRACE_R     24      // }
#define SYN_COMMA       25      // ,
#define SYN_SEMIC       26      // ;
#define SYN_SET         27      // =
#define SYN_QUO         28      // ‘ 单引号
#define SYN_D_QUO       29      // “双引号
#define SYN_AND         51      // &&
#define SYN_OR          52      // ||
#define SYN_NOT         53      // !
#define SYN_TRUE        54      // TRUE
#define SYN_FALSE       55      // FALSE
#define SYN_INT         56      // int
#define SYN_CHAR        57      // char
#define SYN_IF          58      // if
#define SYN_ELSE        59      // else
#define SYN_WHILE       60      // while
#define SYN_SHOW        61      // show
```

二、语法分析

采用 LL(1)分析，文法如下：

FUNC→main() {S}
S→D S
S→A S
S→show(E); S
S→if (B) {S} [else {S}] S
S→while(B) {S} S
S→ε
D→T id [=E] L;

L→, id [=E] L | ε
T→int | char
A→id=E;
B→$T_B B_1$
B_1→‘||’T_BB1 | ε
T_B→$F_B T_{B1}$
T_{B1}→&&$F_B T_{B1}$ | ε
F_B→ E>[=] E | E <[=] E | E== E | E!=E | E | !B | TRUE | FALSE
E→$T_E E_1$
E_1→+$T_E E_1$ | -$T_E E_1$ | ε
T_E→$F T_{E1}$
T_{E1}→*$F T_{E1}$ | /$F T_{E1}$ | ε
F→id | num | (E) | letter