**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

reloop

any

calop/delim:

delim

\*

+ | -

/

digit

digit

\*

other

digit

other

other

/

id:

letter\_

other

letter\_ | digit

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 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\_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\_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

**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🡪TBB1

B1🡪‘||’TBB1 | ε

TB🡪FBTB1

TB1🡪&&FBTB1 | ε

FB🡪 E>[=] E | E <[=] E | E== E | E!=E | E | !B | TRUE | FALSE

E🡪TEE1

E1🡪+TEE1 | -TEE1 | ε

TE🡪FTE1

TE1🡪\*FTE1 | /FTE1 | ε

F🡪id | num | (E)