**词法分析器**

基于c语言设计的源语言LittleC

1. 结构和语义说明

a. 该语言的一个程序由且仅由一个块结构组成，该块中包含两部分：可选的声明语句和必须的执行语句。声明语句用于声明变量，执行语句用于表达计算过程，过程中使用这些变量来存储内部数据和计算结果。

b. 该语言支持两种数据类型：整型和布尔型。整型数据占4个字节，布尔型占1个字节。整型数据的常量是整常数，布尔型数据的常量是 true 和 false 。

c. 整型数据支持 +、-、\*、/ 四种算术运算，运算结果为整型值。整型数据可与>、>=、<、<=、==、六个关系运算符结合，运算结果为布尔型值。

d. 布尔型数据支持 || 、&& 、！三种逻辑运算，运算结果为布尔常量。

e. 整型的变量和常量可以用算术运算符组合成算术表达式，其值为整常数；布尔型的变量和常量可以用逻辑运算符组成成布尔表达式，其值为布尔常量；算术表达式可以加上关系运算符构造出布尔表达式。

f. 整常数和算术表达式可以用于整型变量赋值，布尔常量和布尔表达式可以用于布尔型变量赋值；两者不兼容（文法中使用不同的赋值运算符以示区别）。

g. 各种运算符的优先级和结合规则参照C语言语法理解。

h. 控制流语句中的控制条件限定为1个布尔型变量，该量应该在引用之前定值。

i. 该语言不支持数组、结构体、指针等复杂数据类型。

j. 该语言不含有子程序结构，也不支持过程调用。

k. 该语言有简单的I/O功能：使用read指令能够从键盘读入1个整常数，使用write指令能向屏幕打印输出1个整常数。

1. 词法单元的正则表达式

单词字符一般可以分为以下五类

1. 关键字，源语言的保留标识符
2. 标识符
3. 常数，比如bool常数，int常数
4. 运算符，比如算术运算符，逻辑运算符，赋值运算符
5. 界符，如；，（）{}
6. id -> [a-zA-Z\_]\w\*/g
7. int -> -?([1-9]\d\* | 0)
8. bool -> true | false
9. relop -> < | <= | > | >= | == | !=
10. arithop -> + | - | \* | /
11. assignop -> = | :=
12. bracket -> ( | ) | { | }
13. sign -> ; | ,
14. **keyword** -> while| do | if |else |then |int | bool|read |write

(3)

对词法单元进行编码

词法单元 编码

普通标识符id 1

int类型常数 2

int 3

bool 4

true 5

false 6

while 7

do 8

if 9

else 10

then 11

write 12

read 13

+ 14

- 15

\* 16

/ 17

|| 18

&& 19

! 20

< 21

> 22

<= 23

>= 24

== 25

！= 26

:= 27

= 28

( 29

) 30

{ 31

} 32

， 33

； 34

// 35

(4)

测试实例

见附件

注：因为输出文件是txt文件，所以打印的位数是相同的，但是并没有对齐（因为空字符与其他字符的打印长度不同）