实验三 词法分析

一、 实验目的

1. 根据给定源源语言的构词规则，从任意字符串中识别出该语言所有的合法的单词符号符号，并以等长的二元组形式输出
2. 输入：字符串形式的源程序
3. 输出：单词符号所构成的串，单词以等长的二元组形式呈线

二、 实验过程

1. **删除注释的函数**

   首先考虑将单行注释和多行注释删除，从 input.tx 文件中读入源程序，并将删除注释后的源文件写入 result.txt 文件。算法的编写可以参考以下的状态自动机：

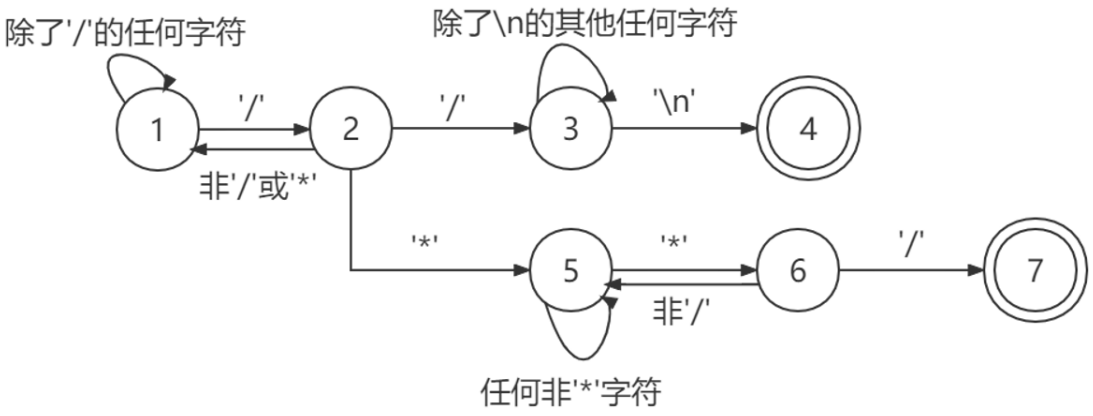

图 1 删除单行或多行注释的状态自动机

具体思路如下：

a. 当输入一个字符串后，开始读取字符
b. 当读入的字符为’/’时，进入状态 2
c. 上一个读取的字符为’/’，当前状态为 2，当前读取的字符为’/’则进入状态 3
d. 进入状态 3 则表明该行后面的都是注释内容可以删除，直到遇见换行符，进入结束状态 4
e. 上一个读取的字符为’/’，当前状态为 2，当前读取的字符不为’/’也不为’*’则返回状态 1
f. 上一个读取的字符为’/’，当前状态为 2，当前读取的字符为’*’则进入状态 5
g. 进入状态 5 后表明后续的内容可能都是注释，一直读直到遇见’*’进入状态 6
h. 进入状态 6 后，如果后一个字符为’/’说明/* */之间的内容为多行注释内容，可以删除，然后进入结束状态 7
i. 若进入状态 6 后，后一个字符不为’/’则返回状态 5

编写代码时需要考虑的问题：

a. 当一开读入的字符就是’/’时，如何判断其是否应该保存？

——需要在开头时进行特判，默认的状态不再是 1

b. 为了简化处理操作，这里不对多余的空格和换行符再做处理

在后续词法分析中，会自动跳过多余的空格和换行符，所以在该函数中不需要处理多余的空格和换行符。

c. 当存在多行注释时，应当先将原来的指针位置保存，等待确定了多行注释的具体位置之后，再将其位置更新

d. 需要时刻考虑遍历读入字符串时指针的溢出情况

2. 标识符的识别

考虑到后续符号表的建立以及和语法、语义分析的对接，这里需要对所有的字符进行分类，按照《编译原理》课程中的介绍，所有的字符可以分成五类：

a. 关键词：if else while main int

b. 运算符：+ - * / > >= < <= = == !=

c. 分隔符：{ } ( ) , ;

d. 标识符：(_ | letter)(letter | digit | _)*

e. 常数：(| -)(digit)(digit)* （仅考虑整正负常数）

所以这里按照分类给出其对应的编号：

表 1 字符分类表

类别	符号
keyword	if else while main int
op	+ - * / < <= > >= = == !=
id	(_ | letter)(letter | digit | _)*
sep	{ } ( ) , ;
num	(| -)(digit)(digit)*

考虑到后续的单词表的建立，这里使用数字对上述的符号进行编码：

表 2 字符编码表

单词符号	编码	单词符号	编码	单词符号	编码
main	0	-	22	<	30
if	1	*	23	<=	31
else	2	/	24	(	32
while	3	=	25	)	33
int	4	==	26	{	34
id	10	!=	27	}	35
num	20	>	28	;	36
+	21	>=	29	,	37

具体思路如下：

a. 首先跳过所有的空格和换行符

b. 识别开头的字符是否是字母，如果是则只有两种情况：标识符或者关键字，我们先将后续连续的字母、数字和下划线读入组成token，再进行判断：如果是关键字则返回关键字，否则返回标识符

c. 识别开头的字符如果是数字，那么一定是常数，识别连续的数字并返回常数(需要注意负数的判断)

d. 然后剩下负数和一系列运算符，进行逐个判断并返回即可

实现细节注意

a. 需要和注释删除函数协作，首先利用注释删除函数得到不含注释的源程序字符串，再调用编写的分词函数进行标识符的识别

b. 由于负数和减法运算符的开头一致，所以需要额外的判断，在流

程示意图中已给出

c. 涉及到两个字符的运算符也需要进行连续判断，在源代码中有所体现

d. 词法分析的函数返回其编码 index，与全局的 token 一起，存入全局的 vector word_list 中

e. 考虑到非法字符的输入需要提示错误，所以我们引入了全局变量 row 来记录行号，通过\n 来判断第几行，如果当前无法识别当前字符则返回报错信息

3. 词法分析

将上述的两个函数和主程序的输入输出进行拼接，得到最终的词法分析器：

输入：所给文法的源程序文本 input.txt

输出：二元组[index, token]构成的序列以及删除注释后得到的源程序 result.txt

具体实现时，将二元组作为结构体，存入 vector 中

整个词法分析器的示意图如下：

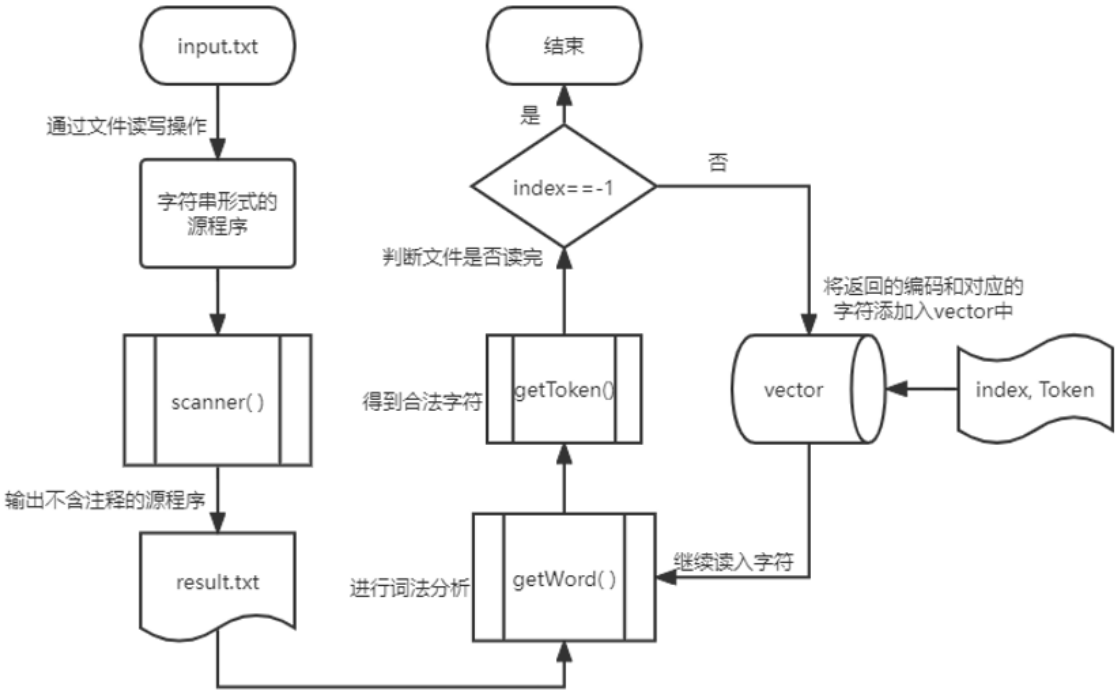

图 3 词法分析流程图

三、 实验结果

1. 功能测试：

   我们使用文件读写的方式将源程序输入 input.txt，在 result.txt 中得到删除注释的源程序代码，在 cmd 命令行中得到标识符识别的结果。

四、 实验总结

1. 编译原理实验不是独立分开的实验，每个实验都是紧密联系的，我们应该充分利用前面实验得到的函数、结果，并考虑到后续实验的要求，设计并实现本次实验的内容，为下一次实验打好基础

2. 善于使用流程图来完善程序设计的思路，使得写代码时效率更高
3. 善于利用状态自动机来解决状态变化的问题，让程序的逻辑更加清晰
4. 为了测试程序的健壮性，需要给定一些较为极端的输入，测试程序能否正常运行，从而排除问题，优化程序

五、 代码展示