

天津商业大学学生实验报告

开课实验室：		开课时间 2015 年 3 月 26 日				实验报告 2015 年 4 月 17 日			
学院名称	信息工程学院	年级、专业、班	软件工程 1201 班	学号	20125041	姓名	王靖伟	同组姓名	沈宇帆、康振山、管立洋
课程名称	编译原理	实验项目名称	实验一 词法分析器			指导教师	周艳聪		
实验类型		验证 <input type="checkbox"/>	综合 <input type="checkbox"/>	设计 <input type="checkbox"/>	创新 <input type="checkbox"/>	成绩			
教师评语	教师签名：年 月 日								

- 一、实验目的：
掌握计算机语言的词法分析器的开发。
- 二、实验内容：
1. 查找某语言的词法规则。
 2. 利用状态转换图编程实现识别某些单词（>=5 类）的小型词法分析器。
- 三、可以识别的 Token
- (1) 关键词
"int","double","float","for","printf"
 - (2) 用户自定义标识符
以下划线（_）或英文半角字符开始，由下划线（_）、英文半角字符、数字半角字符组成的单词
 - (3) 数字
整数、小数、科学记数法
 - (4) 空白字符
空格（space）、回车（\n）、制表符（\t）
 - (5) 符号
包括“+”、“-”、“*”、“/”、“(”、“)”、“{”、“}”、“;”、“:”、“<”、“>”、“=”、“!”
- 四、正则表达式

```
id -> [_A-Za-z]([0-9]|[_A-Za-z])*
num -> [0-9]+(.[0-9]+)?([Ee][+-]?[0-9]+)?
ws -> (blank|tab|newline)*
symbol -> +|-|*|/|(|){|}|,|;|<|>|=|!
```

五、状态转换图

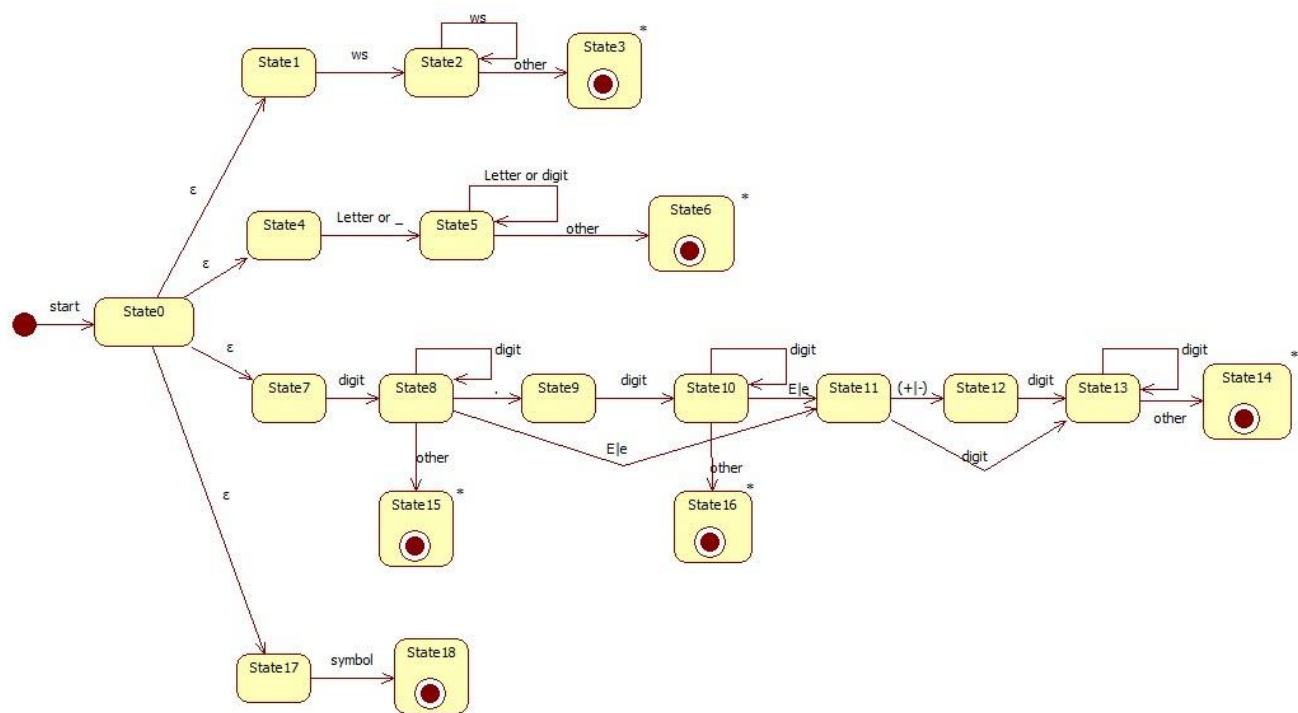


图 1. 状态转换图

六、字符串的识别路径

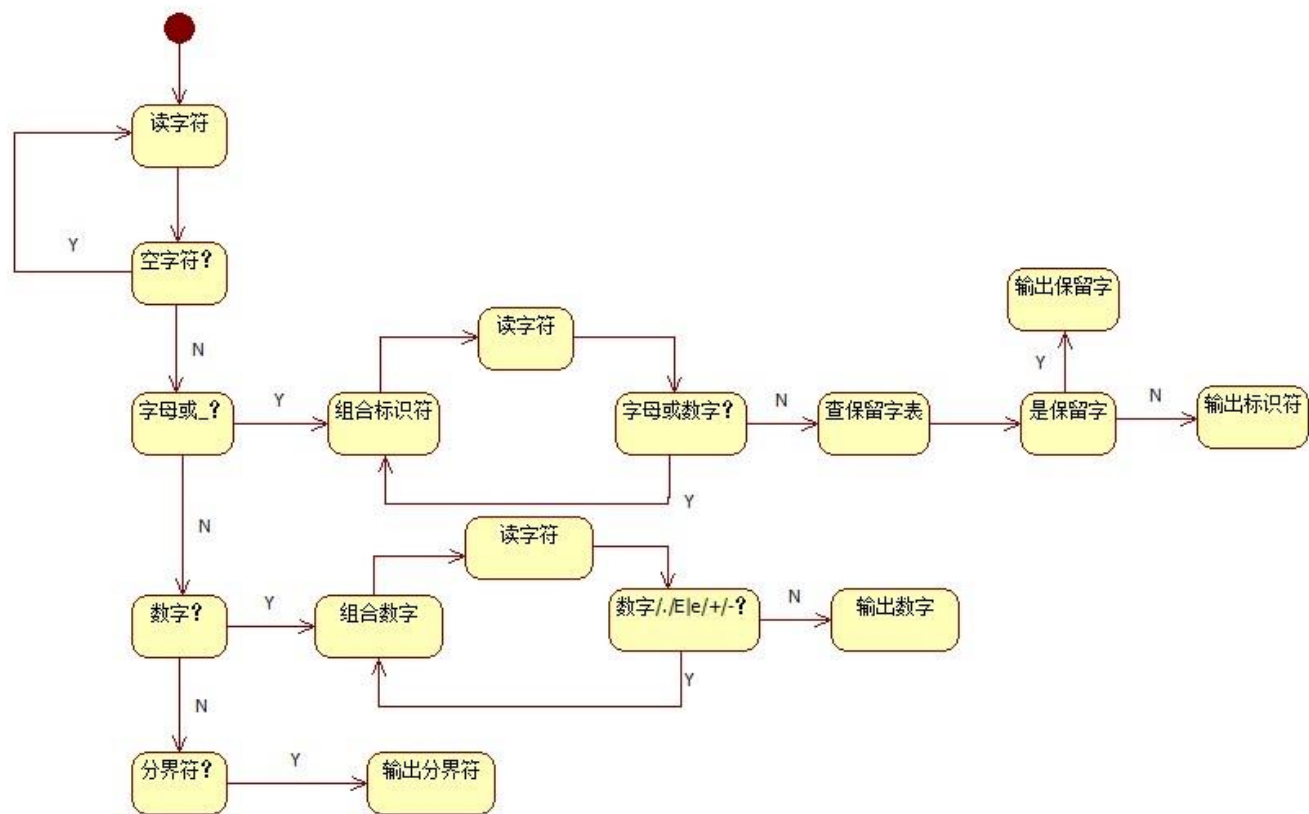


图 2. 字符串的识别路径

天津商业大学学生实验报告

七、程序运行截图

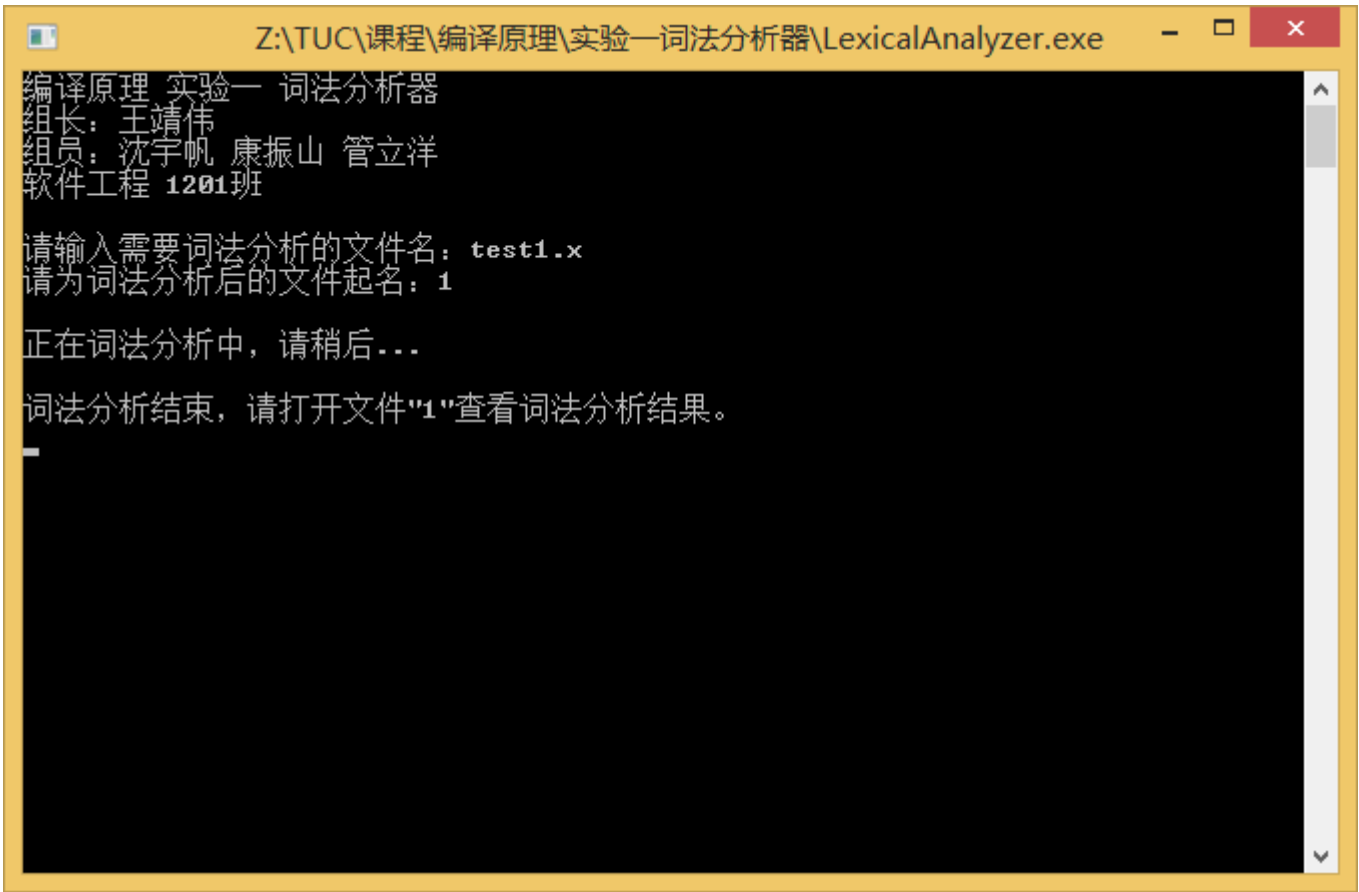


图 3. 词法分析器的运行

八、运行结果

(1) test1.x 测试

```
int _a=0;
for(int i1=0; i1<100; i1++)
{
    _a = _a+3;
}
printf(_a);
```

词法分析结果:

<int,int>	<+,+>
<Id,_a>	<+,+>
<=,=>	<),)>
<Num,0>	<{,{>
<;,;>	<Id,_a>
<for,for>	<=,=>
<(,(>	<Id,_a>
<int,int>	<+,+>
<Id,i1>	<Num,3>
<=,=>	<;,;>
<Num,0>	<},}>
<;,;>	<printf,printf>
<Id,i1>	<(,(>
<<,<>	<Id,_a>
<Num,100>	<),)>
<;,;>	<;,;>
<Id,i1>	

天津商业大学学生实验报告

(1) test1.x 测试

```
int _a=0;
double b1=3.5;
for(int i1=0; i1<100; i1++)
{
    _a = _a+3;
    b1 = b1+1.1E-100;
}
printf(_a);
```

词法分析结果:

<int,int>	<+,+>
<Id,_a>	<+,+>
<=,=>	<),)>
<Num,0>	<{,{>
<;,;>	<Id,_a>
<double,double>	<=,=>
<Id,b1>	<Id,_a>
<=,=>	<+,+>
<Num,3.5>	<Num,3>
<;,;>	<;,;>
<for,for>	<Id,b1>
<,(,>	<=,=>
<int,int>	<Id,b1>
<Id,i1>	<+,+>
<=,=>	<Num,1.1E-100>
<Num,0>	<;,;>
<;,;>	<},}>
<Id,i1>	<printf,printf>
<<,<>	<,(,>
<Num,100>	<Id,_a>
<;,;>	<),)>
<Id,i1>	<;,;>

九、程序源代码

行号	LexicalAnalyzer.cc
1	#include <iostream>
2	#include <fstream> //用于读写文件
3	#include <cctype> //C 中的 ctype.h
4	
5	using namespace std;
6	
7	string keyword[] = {"int","double","float","for","printf"};
8	string pathin, pathout; //保存文件 io 路径
9	
10	ifstream fin; //创建读取流对象 fin
11	ofstream fout; //创建输出流对象 fout
12	
13	int LexAna ()
14	{
15	cout<<"请输入需要词法分析的文件名: ";
16	cin>>pathin;

天津商业大学学生实验报告

```
17     cout<<"请为词法分析后的文件起名: ";
18     cin>>pathout;
19     cout<<endl<<"正在词法分析中, 请稍后..."<<endl;
20
21     fin.open(pathin,ios_base::in); //以只读方式打开文件
22     if(!fin.is_open())
23     {
24         cout<<"打开"<<pathin<<"失败, 请检查路径是否有误"<<endl;
25         return -1; //打开文件有误, 返回值为-1
26     }
27     fout.open(pathout,ios_base::out|ios_base::trunc); //以只写方式打开文件, 如果文
28 件已存在, 则清零后写入
29     if(!fout.is_open())
30     {
31         cout<<"打开"<<pathout<<"失败, 请检查路径是否有误"<<endl;
32         return -1; //打开文件有误, 返回值为-1
33     }
34
35     int lexErr = 0; //词法分析错误代码
36     char ch;
37     string token;
38
39     while (fin)
40     {
41         if(!fin.get(ch))
42         {
43             break;
44         } //如果获取不到字符就跳出
45
46         while(' '==ch||'\n'==ch||'\t'==ch)
47         {
48             if(!fin.get(ch))
49             {
50                 break;
51             } //解决文件末尾空白字符问题
52         } //忽略掉所有空白字符
53
54         if ('_'==ch||isalpha(ch)) //[_A-Za-z]([0-9]|[_A-Za-z])*//判断是否为字母
55 或下划线'_'
56         {
57             token = "";
58             token += ch;
59             fin.get(ch);
60
61             while('_'==ch||isalnum(ch)) //判断是否为字母或数字或下划线'_'
```

天津商业大学学生实验报告

```
62         {
63             token += ch;
64             fin.get(ch);
65         }
66
67         int a = 0;
68         int b = 0;
69         for(string i : keyword)
70         {
71             a++; //记录字符串数组元素的个数
72             if(token.compare(i)==0) //两字符串相等则返回 0
73             {
74                 fout<<"<"<<token<<"", "<<token<<">"<<endl;
75                 b = a; //记录匹配的 keyword 字典的位置
76             }
77         }
78         if(0==b)
79         {
80             fout<<"<"<<"Id"<<"", "<<token<<">"<<endl;
81         }
82     }
83
84     else if(isdigit(ch)) // [0-9]+(.[0-9]+)?([Ee][+-]?[0-9]+)?
85     {
86         token = "";
87         token += ch;
88         fin.get(ch);
89         while(isdigit(ch))
90         {
91             token += ch;
92             fin.get(ch);
93         }
94         if('.'==ch)
95         {
96             token += ch;
97             fin.get(ch);
98             if(!isdigit(ch))
99             {
100                 cout<<endl<<"小数点后不是数字，词法错误，终止分析！"<<endl;
101                 fout<<"小数点后不是数字，词法错误，终止分析！"<<endl;
102                 return 1;
103             }
104             while(isdigit(ch))
105             {
106                 token += ch;
```

天津商业大学学生实验报告

```

107         fin.get(ch);
108     }
109 }
110 if('E'==ch||'e'==ch)
111 {
112     token += ch;
113     fin.get(ch);
114     if('+'==ch||'-'==ch)
115     {
116         token += ch;
117         fin.get(ch);
118     }
119     if(!isdigit(ch))
120     {
121         cout<<endl<<"科学记数法部分词法错误，终止分析！"<<endl;
122         fout<<"科学记数法部分词法错误词法错误，终止分析！"<<endl;
123         return 2;
124     }
125     while(isdigit(ch))
126     {
127         token += ch;
128         fin.get(ch);
129     }
130     if(!isdigit(ch) && ';'!=ch)
131     {
132         cout<<endl<<"科学记数法部分词法错误，终止分析！"<<endl;
133         fout<<"科学记数法部分词法错误，终止分析！"<<endl;
134         return 2;
135     }
136 }
137 fout<<"<"<<"Num"<<","<<token<<">"<<endl;
138 }
139
140 if ('+'==ch||'-'==ch||'*'==ch||
141     '/'==ch||'('==ch||')'==ch||
142     '{'==ch||'}'==ch||';'==ch||
143     ','==ch||'<'==ch||'>'==ch||
144     '='==ch||'!'==ch) //符号
145 {
146     token = "";
147     token += ch;
148     fout<<"<"<<token<<","<<token<<">"<<endl;
149 }
150 }
151

```

天津商业大学学生实验报告

```
152     fin.close();    //关闭对象到文件的连接
153     fin.clear();    //清理/重置 fin 对象
154     fout.close();
155     fout.clear();
156     cout<<endl<<"词法分析结束，请打开文件\"<<pathout<<\"查看词法分析结果。"<<endl;
157     cin.get();
158     cin.get();
159
160     return lexErr;
161 }
162
163 int main()
164 {
165     cout<<"编译原理 实验一 词法分析器"<<endl;
166     cout<<"组长：王靖伟"<<endl;
167     cout<<"组员：沈宇帆 康振山 管立洋"<<endl;
168     cout<<"软件工程 1201 班"<<endl<<endl;
169
170     int Err;
171     Err = LexAna();
172     return Err;
173 }
```