

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 5 月 22 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	实验二 递归下降语法分析器			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	<div style="text-align: right;"> 教师签名： 年 月 日 </div>								

一、实验目的:

掌握计算机语言的语法分析器的开发。

二、实验内容:

1. 根据某文法, 构造递归下降语法分析程序。
2. 给出分析过程中所用的产生式序列。

三、产生式

$E \rightarrow TG$	G 为 E'
$E \rightarrow +TG \mid \wedge$	\wedge 为 ε
$T \rightarrow FS$	S 为 T'
$T \rightarrow *FS \mid \wedge$	
$F \rightarrow i \mid (E)$	

天津商业大学学生实验报告

四、截图

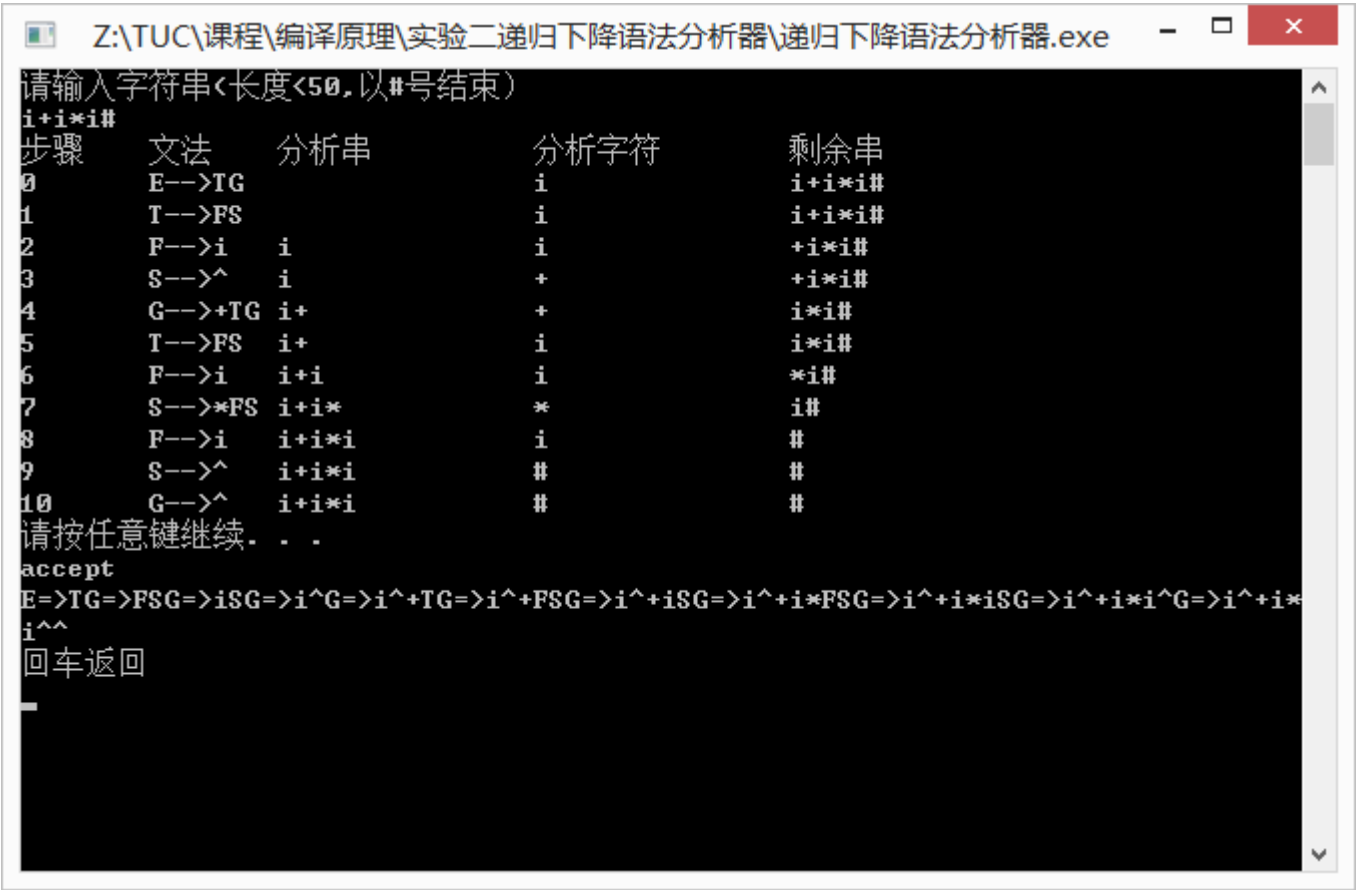


图 1. 运行截图

九、程序源代码

行号	LexicalAnalyzer.cc
1	#include <stdio.h>
2	#include <stdlib.h>
3	#include <string.h>
4	
5	char a[50],b[50],d[200],e[10],ch;
6	/*数组 a 存输入串，数组 d 存推导式*/
7	int n1,i1=0,flag=1,n=5; /*flag=1 处理非终结符，flag=0 处理终结符*/
8	int total=0; /*步骤计数器*/
9	int E(),E1(),T();
10	int G();/*E'*/
11	int S();/*T'*/
12	int F();
13	void input();
14	void input1();
15	void output();
16	
17	int main() /*递归分析*/
18	{

天津商业大学学生实验报告

```

19  int f,p,j=0;
20  char x;
21  d[0]='E';d[1]='=';d[2]='>';d[3]='T';d[4]='G';d[5]='#';
22  printf("请输入字符串(长度<50,以#号结束) \n");
23  do
24  {
25      scanf("%c",&ch);
26      a[j]=ch;
27      j++;
28  }while(ch!='#');
29  n1=j; /* n1 输入串实际长度 */
30  ch=b[0]=a[0];
31  printf("步骤\t 文法\t 分析串\t\t 分析字符\t 剩余串\n");
32  f=E1();
33  system("pause");
34
35  if (f==0)
36      return -1; /*有语法错误，失败退出*/
37  if (ch=='#')
38  {
39      printf("accept\n"); /*输入串是文法的句子*/
40      p=0;
41      x=d[p];
42      while(x!='#')
43      {
44          printf("%c",x);
45          p=p+1;
46          x=d[p]; /*输出推导式*/
47      }
48
49  }
50  else
51  {
52      printf("error\n");
53      printf("回车返回\n");
54      getchar();
55      getchar();
56      return -1;
57  }
58  printf("\n");
59  printf("回车返回\n");
60  getchar();
61  getchar();
62  }
63

```

天津商业大学学生实验报告

```

64 int E1()/* 有语法错误函数返回值为 0， 否则为 1*/
65 {
66     int f,t;
67     printf("%d\tE-->TG\t",total);
68     total++;
69     flag=1;
70     input();
71     input1();
72     f=T();
73     if (f==0)
74         return(0);
75     t=G();
76     if (t==0)
77         return(0);
78     else return(1);
79 }
80
81 int E()
82 {
83     int f,t;
84     printf("%d\tE-->TG\t",total);
85     total++;
86     e[0]='E';e[1]='=';e[2]='>';e[3]='T';e[4]='G';e[5]='#';
87     output();
88     flag=1;input();input1();
89     f=T();
90     if (f==0)
91         return(0);
92     t=G();
93     if (t==0)
94         return(0);
95     else
96         return(1);
97 }
98
99 int T()
100 {
101     int f,t;
102     printf("%d\tT-->FS\t",total); total++;
103     e[0]='T';e[1]='=';e[2]='>';e[3]='F';e[4]='S';e[5]='#';
104     output();
105     flag=1;input();input1();
106     f=F();
107     if (f==0)
108         return(0);

```

天津商业大学学生实验报告

```

109     t=S();
110     if (t==0)
111         return(0);
112     else
113         return(1);
114 }
115
116 int G()
117 {
118     int f;
119     if(ch=='+')
120     {
121         b[i1]=ch;
122         printf("%d\tG-->+TG\t",total);
123         total++;
124         e[0]='G';e[1]='=';e[2]='>';e[3]='+';e[4]='T';e[5]='G';e[6]='#';
125         output();
126         flag=0;input();input1();
127         ch=a[++i1];
128         f=T();
129         if (f==0)
130             return(0);
131         G();
132         return(1);
133     }
134     printf("%d\tG-->^\t",total);
135     total++;
136     e[0]='G';e[1]='=';e[2]='>';e[3]='^';e[4]='#';
137     output();
138     flag=1;
139     input();
140     input1();
141     return(1);
142 }
143
144 int S()
145 {
146     int f,t;
147     if(ch=='*')
148     {
149         b[i1]=ch;
150         printf("%d\tS-->*FS\t",total); total++;
151         e[0]='S';e[1]='=';e[2]='>';e[3]='*';e[4]='F';e[5]='S';e[6]='#';
152         output();
153         flag=0;input();input1();

```

天津商业大学学生实验报告

```

154         ch=a[++i1];
155         f=F();
156         if (f==0)
157             return(0);
158         t=S();
159         if (t==0)
160             return(0);
161         else
162             return(1);
163     }
164     printf("%d\tS-->^\t",total); total++;
165     e[0]='S';e[1]='=';e[2]='>';e[3]='^';e[4]='#';
166     output();
167     flag=1;a[i1]=ch;input();input1();
168     return(1);
169 }
170
171 int F()
172 {
173     int f;
174     if(ch=='(')
175     {
176         b[i1]=ch;
177         printf("%d\tF-->(E)\t",total); total++;
178         e[0]='F';e[1]='=';e[2]='>';e[3]='(';e[4]='E';e[5]=')';e[6]='#';
179         output();
180         flag=0;input();input1();
181         ch=a[++i1];
182         f=E();
183         if (f==0)
184             return(0);
185         if(ch==')')
186         {
187             b[i1]=ch;
188             printf("%d\tF-->(E)\t",total); total++;
189             flag=0;
190             input();
191             input1();
192             ch=a[++i1];
193         }
194         else
195         {
196             printf("error\n");
197             return(0);
198         }

```

天津商业大学学生实验报告

```

199     }
200     else
201         if(ch=='i')
202         {
203             b[i1]=ch;
204             printf("%d\tF-->i\t",total); total++;
205             e[0]='F';e[1]='=';e[2]='>';e[3]='i';e[4]='#';
206             output();
207             flag=0;input();input1();
208             ch=a[++i1];
209         }
210     else
211     {
212         printf("error\n");
213         return(0);
214     }
215     return(1);
216 }
217
218 void input()
219 {
220     int j=0;
221     for (;j<=i1-flag;j++)
222         printf("%c",b[j]); /*输出分析串*/
223     printf("\t\t");
224     printf("%c\t\t",ch); /*输出分析字符*/
225 }
226
227 void input1()
228 {
229     int j;
230     for (j=i1+1-flag;j<n1;j++)
231         printf("%c",a[j]); /*输出剩余字符*/ printf("\n");
232 }
233
234 void output() /*推导式计算*/
235 {
236     int m,k,j,q;
237     int i=0;
238     m=0;k=0;q=0;
239     i=n;
240     d[n]='=';d[n+1]='>';d[n+2]='#';
241     n=n+2;
242     i=n;
243     i=i-2;

```

天津商业大学学生实验报告

244	while(d[i]!='>'&&i!=0)
245	i=i-1;
246	i=i+1;
247	while(d[i]!=e[0])
248	i=i+1;
249	q=i;m=q;k=q;
250	while(d[m]!='>')
251	m=m-1;
252	m=m+1;
253	while(m!=q)
254	{
255	d[n]=d[m];
256	m=m+1;
257	n=n+1;
258	}
259	d[n]='#';
260	for(j=3;e[j]!='#';j++)
261	{
262	d[n]=e[j];
263	n=n+1;
264	}
265	k=k+1;
266	while(d[k]!='=')
267	{
268	d[n]=d[k];
269	n=n+1;
270	k=k+1;
271	}
272	d[n]='#';
273	}