

北京邮电大学 计算机学院
《编译原理与技术》实验报告

姓名 王睿嘉

学号 2015211906

班级 2015211307

语法分析程序的设计与实现

一、实验内容 and 环境描述

1. 实验内容

编写语法分析程序，实现对算数表达式的语法分析。要求所分析算数表达式由如下的文法产生：

$E \rightarrow E+T \mid E-T \mid T$

$T \rightarrow T * F \mid T / F \mid F$

$F \rightarrow (E) \mid \text{num}$

要求：编程实现算法 4.1，构造 LL(1) 预测分析程序。在对输入的算数表达式进行分析的过程中，依次输出所采用的产生式。

2. 实验环境

本次实验在 Win10 下，使用 VS2015 作为 IDE 编写代码，并完成相关调试工作。

二、程序设计说明

1. 程序功能

手动算出 FIRST 集合和 FOLLOW 集合如下：

	FIRST	FOLLOW
E	(, num), \$
A	+, -, epsilon), \$
T	(, num	+, -,), \$
B	*, /, epsilon	+, -,), \$
F	(, num	+, -, *, /,), \$

根据两集合构造相应分析表。由已知分析表，实现了题目要求的算法 4.1（用户输入待分析符号序列，程序输出分析过程中输入缓冲区、符号栈状态变化及所调用的产生式）。并在此过程中，进行一定错误处理——加入同步化信息，即：

对于终结符 b 属于 FOLLOW (A)，若分析表中以 A 为行， b 为列的表项为空，则加入“synch”。在分析过程中，若遇到“synch”，则弹出相应非终结符。

2. 输入与输出

1) 输入的定义

用户只需要输入待分析符号序列。

2) 输出的定义

程序给出两部分输出：

该文法的分析表；

对应于输入符号序列的分析结果（各步骤中输入缓冲区、符号栈状态变化及所调用的产生式）。

3. 程序实现说明

1) 函数说明

```
char isDigit(char);           //判断是否为数字
int map(char);               //将文法符号映射为下标
void initial();              //初始化，构建分析表
void printTable();           //输出分析表
int terminalSymbol(char);    //判断是否为终结符
int getLength();             //获得栈长度
void analyze(string);        //预测分析控制程序
```

预测分析程序的关键在于决定哪个产生式运用于非终结符。对于栈顶非终结符和当前输入符号，在分析表中找到相应表项，并根据其进行一定分析动作。

根据栈顶符号 X 和当前输入符号 a ，分析动作有 4 种可能：

- (1) $X=a=\$$ ，宣告分析成功，程序退出，并输出信息提示用户输入符号序列符合文法规则；
- (2) $X=a\neq \$$ ，从栈顶弹出 X ，输入指针前移一个位置；
- (3) 表项为 synch，进行弹栈错误处理，继续进行后续分析；
- (4) 表项为 error，报告发现错误，程序退出，并输出信息提示用户输入字符串不符合文法规则；
- (5) 以 $M[X, a]$ 表示分析表表项。若 $M[X, a]=X\rightarrow Y_1Y_2\cdots Y_n$ ，先将 X 从栈顶弹出，然后把产生式的右部符号串按反序推入栈中；若 $M[X, a]=X\rightarrow\epsilon$ ，从栈顶弹出 X 。

2) 数据结构说明

本次实验的难点之一在于数据结构的使用：以何种方式实现栈？以何种方式存储输入字符串？以何种方式存储分析表，并能快速访问表项？

```
char stack[MAX];             //符号栈
```

其作用为：存放一系列文法符号， $\$$ 存于栈底。分析开始时，先将 $\$$ 入栈，以标识栈底，后再将文法的开始符号入栈。其中 MAX 代表符号栈容量。

```
string table[5][8];          //分析表
```

利用二维数组存储分析表，实现二元映射，分析时利用 $[][]$ 进行索引，从而快速获得所需要的表项。

```
string input;                //输入符号序列
```

三、 运行测试

1. 测试用例 1

输入及运行结果如下：

```

>>*****分析表*****<<
*****+ - * / ( ) num $*****
E      error      error      error      error      E->TA      synch      E->TA      synch
A      A->+TA      A->-TA      error      error      error      A->epsilon      error      A->epsilon
T      synch      synch      error      error      T->FB      synch      T->FB      synch
B      B->epsilon  B->epsilon  B->*FB      B->/FB      error      B->epsilon      error      B->epsilon
F      synch      synch      synch      synch      F->(E)      synch      F->num      synch
*****
请输入待分析符号串:
(9-3)/(2+4)+5*7
*****
>>*****分析过程*****<<
*****输入*****
*****输出*****
栈      输入      输出
SE      (9-3)/(2+4)+5*7$      E->TA
SAT      (9-3)/(2+4)+5*7$      T->FB
SABF      (9-3)/(2+4)+5*7$      F->(E)
SAB)E(      (9-3)/(2+4)+5*7$
SAB)E      9-3/(2+4)+5*7$      E->TA
SAB)AT      9-3/(2+4)+5*7$      T->FB
SAB)ABF      9-3/(2+4)+5*7$      F->num
SAB)AB9      9-3/(2+4)+5*7$
SAB)AB      -3/(2+4)+5*7$      B->epsilon
SAB)A      -3/(2+4)+5*7$      A->-TA
SAB)AT-      -3/(2+4)+5*7$
SAB)AT      3/(2+4)+5*7$      T->FB
SAB)ABF      3/(2+4)+5*7$      F->num
SAB)AB3      3/(2+4)+5*7$
SAB)AB      )/(2+4)+5*7$      B->epsilon
SAB)A      )/(2+4)+5*7$      A->epsilon
SAB)      )/(2+4)+5*7$
SAB      /(2+4)+5*7$      B->/FB
SABF/      /(2+4)+5*7$
SABF      (2+4)+5*7$      F->(E)
SAB)E(      (2+4)+5*7$
SAB)E      2+4)+5*7$      E->TA
SAB)AT      2+4)+5*7$      T->FB
SAB)ABF      2+4)+5*7$      F->num
SAB)AB2      2+4)+5*7$
SAB)AB      +4)+5*7$      B->epsilon
SAB)A      +4)+5*7$      A->+TA
SAB)AT+      +4)+5*7$
SAB)AT      4)+5*7$      T->FB
SAB)ABF      4)+5*7$      F->num
SAB)AB4      4)+5*7$
SAB)AB      )+5*7$      B->epsilon
SAB)A      )+5*7$      A->epsilon
SAB)      )+5*7$
SAB      +5*7$      B->epsilon
SA      +5*7$      A->+TA
SAT+      +5*7$
SAT      5*7$      T->FB
SABF      5*7$      F->num
SAB5      5*7$
SAB      *7$      B->*FB
SABF*      *7$
SABF      7$      F->num
SAB7      7$
SAB      $      B->epsilon
SA      $      A->epsilon
S      $      Accept
请按任意键继续. . .

```

说明：该测试样例输入为正确的符号串序列。预测分析程序进行语法分析，给出相应的输入缓冲区、符号栈变化及产生式序列。

2. 测试用例 2

输入及运行结果如下：

```

*****
>>                                     分析表                                     <<
*****
+      -      *      /      (      )      num      $
-----
E      error      error      error      error      E->TA      synch      E->TA      synch
A      A->+TA      A->-TA      error      error      error      A->epsilon      error      A->epsilon
T      synch      synch      error      error      T->FB      synch      T->FB      synch
B      B->epsilon      B->epsilon      B->*FB      B->/FB      error      B->epsilon      error      B->epsilon
F      synch      synch      synch      synch      F->(E)      synch      F->num      synch
*****
请输入待分析字符串:
(9-3)*6+(2/1
*****
>>                                     分析过程                                     <<
*****
栈      输入      输出
-----
$E      (9-3)*6+(2/1$      E->TA
$AT      (9-3)*6+(2/1$      T->FB
$ABF      (9-3)*6+(2/1$      F->(E)
$AB)E(      (9-3)*6+(2/1$
$AB)E      9-3)*6+(2/1$      E->TA
$AB)AT      9-3)*6+(2/1$      T->FB
$AB)ABF      9-3)*6+(2/1$      F->num
$AB)AB9      9-3)*6+(2/1$
$AB)AB      -3)*6+(2/1$      B->epsilon
$AB)A      -3)*6+(2/1$      A->-TA
$AB)AT-      -3)*6+(2/1$
$AB)AT      3)*6+(2/1$      T->FB
$AB)ABF      3)*6+(2/1$      F->num
$AB)AB3      3)*6+(2/1$
$AB)AB      ))*6+(2/1$      B->epsilon
$AB)A      ))*6+(2/1$      A->epsilon
$AB)      ))*6+(2/1$
$AB      *6+(2/1$      B->*FB
$ABF*      *6+(2/1$
$ABF      6+(2/1$      F->num
$AB6      6+(2/1$
$AB      +(2/1$      B->epsilon
$A      +(2/1$      A->+TA
$AT+      +(2/1$
$AT      (2/1$      T->FB
$ABF      (2/1$      F->(E)
$AB)E(      (2/1$
$AB)E      2/1$      E->TA
$AB)AT      2/1$      T->FB
$AB)ABF      2/1$      F->num
$AB)AB2      2/1$
$AB)AB      /1$      B->/FB
$AB)ABF/      /1$
$AB)ABF      1$      F->num
$AB)AB1      1$
$AB)AB      $      B->epsilon
$AB)A      $      A->epsilon
$AB)      $      error
*****
请按任意键继续. . .

```

说明：该测试样例输入为错误的字符串序列。预测分析程序进行语法分析，遇到错误时输出提示信息，并退出。

四、实验结论和心得

1. 实验结论

通过此次实验，进一步了解了语法分析程序的工作原理，重点解析了 LL(1) 文法，并对其中一些关键点，如分析表的构造、预测分析过程等，有了更深刻的理解和更理性的认识。

2. 实验心得

在本次实验中，遇到的主要问题有以下五点：

- a) 文法符号的映射。本实验利用二维数组作为分析表，课本所讲分析表均以非终结符为行，终结符为列，这显然与数组下标不符。后构造 `map()` 函数，进行文法符号至下标的映射，该问题得到解决；
- b) 栈的操作。本实验利用字符数组作为符号栈，起初困惑如何实现弹栈操作。经思考，将序列末尾的文法符号赋为 `'\0'`，等价于弹出栈顶符号，该问题得到解决；
- c) 输入符号串的操作。本实验输入符号串为 `string` 类型，要实现算法中的向前指针前移，并不好操作。经查阅网上资料，利用 `substr()` 函数，该问题得到解决；
- d) 对于终结符 `num` 的处理。文法中 `num` 所指为 `0~9` 数字，但分析表表项中所填为 `'->num'`，无法与其余产生式采用相同的方法入栈，需要进行特殊处理，即将向前指针所指数字直接入栈；
- e) 莫名的 `bug`。在运行过程中一直出现数组下标越界，后经调试发现：粗心大意将 `'->'` 也倒序入栈，修改后，该问题得到解决。

实践出真知，本次语法分析程序的设计与实现实验是对课堂和书本所学知识的补充。语法分析的实际情形与已了解到的原理大体一致，但又复杂许多。通过自己动手、亲力亲为编写代码，加深了对预测分析程序的理解和记忆，收获颇丰。