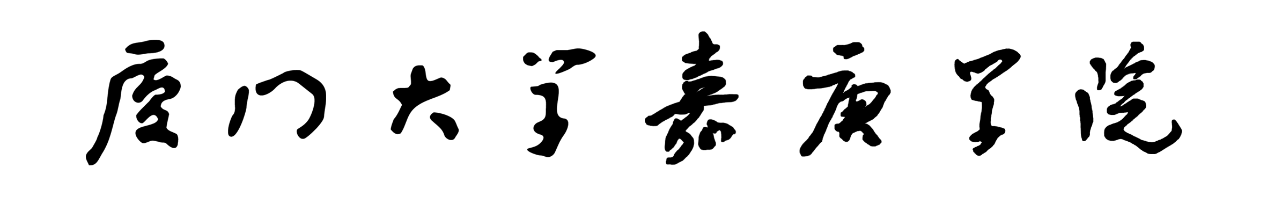
****

**(论文首页纸)**

**2018—2019 学年第 一 学期**

**课程名称： 语法分析的高级语言实现**

**任课教师： 郭一晶**

**成员：**

**SWE16003徐麒涵**

**SWE16004黄潇**

**SWE16024连嘉华**

**SWE16035蔡志鹏**

提交日期：2018年12月25日

1. **实验目的**

加深对语法分析器工作过程的理解；能够采用一种编程语言实现简单的语法分析程序；能够使用自己编写的分析程序对简单的程序段进行语法分析。

1. **实验分工**

徐麒涵：负责编写实验报告

黄潇：负责使用C++实现

连嘉华：负责使用C++实现

蔡志鹏：负责使用C++实现

1. **程序说明**

Analysis.h和analysis.cpp实现词法分析

Grammer.h和Grammer.cpp 存储文法的数据结构

RemoveRecursion.h和RemoveRecursion.cpp实现消除直接递归、间接递归、回溯

LL.h和LL.cpp实现LL(1)分析法

Main.cpp为程序的入口

在main函数中首先调用RemoveRecursion类的remove方法进行消除直接递归、间接递归、回溯；然后调用LL类的init方法进行LL(1)分析表构建。

1. **其他说明**

程序输入的文法必须从S开始

$代表空字

在RemoveRecursion中通过removeIndirectRecursion消除间接左递归，通过simplify化简，与课本上的消除间接左递归的方式不同，对所有文法循环，循环过程中先提取了公共左因子，然后内层循环进行循环，之后再次提取公共左因子，最后才消除直接左递归

大概的伪代码如下

For 文法 i=n-1 to 0

提取公共左因子

For 文法 j=n-1 to i+1

文法j入文法i

提取公共左因子

消除直接左递归

在LL中通过init方法构建LL(1)表，先调用getFirst方法获得first集，然后通过getFollow获得follow集。再通过getTable求得LL(1)表。

获得first集的操作如下：

对文法进行dfs，对每条产生式碰到了终结符直接加入，非终结符则对这个非终结符优先dfs，然后去掉空字加入

Dfs的伪代码

For 文法的所有产生式 i=0 to n-1

For 产生式 i=0 to n-1

If 终结符

加入first

Break；

Else

Dfs（这个非终结符）

加入first

If 有 Epsilon

Continue；

Else

Break；

全都有Epsilon则Epsilon加入first

1. **运行结果截图显示**

输入

14

S->H(){L}|H(){}

L->LG|G

G->C|I|W

C->E;|;

I->J(E)GKG

W->U(E)G

E->V=E|B

V->D|D[E]

B->AQA|A

A->AOT|T

O->+|-

T->TMF|F

M->\*|/

F->(E)|V|N

Test.txt内容

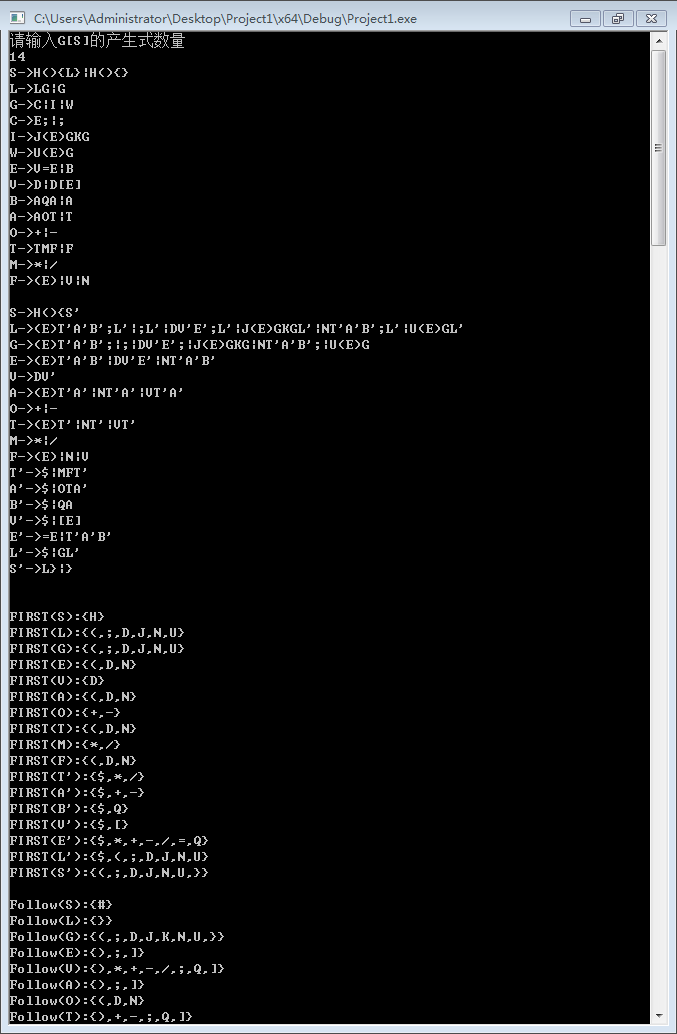
main(){

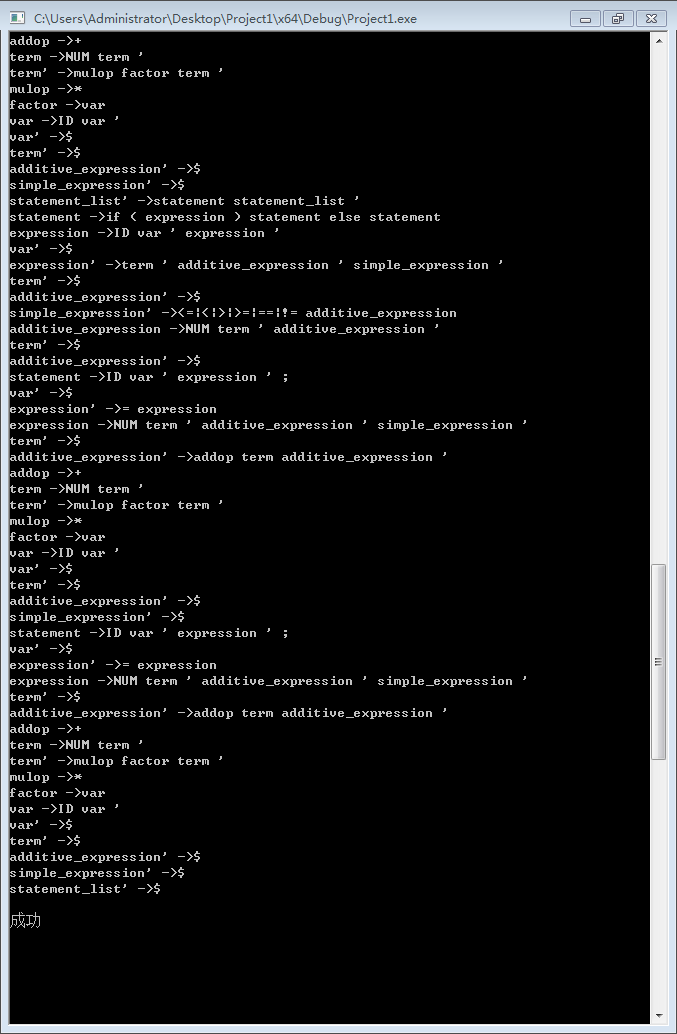
while(a!=0) a= 3+ 21\*a;

a= 3+ 21\*a;

if(a==0) a= 3+ 21\*a; else a= 3+ 21\*a;

}



****

1. **问题**

如果输入文法

14

S->H(){L}|H(){}

L->LG|G

G->C|I|W

C->E;|;

I->J(E)G|J(E)GKG

W->U(E)G

E->V=E|B

V->D|D[E]

B->AQA|A

A->AOT|T

O->+|-

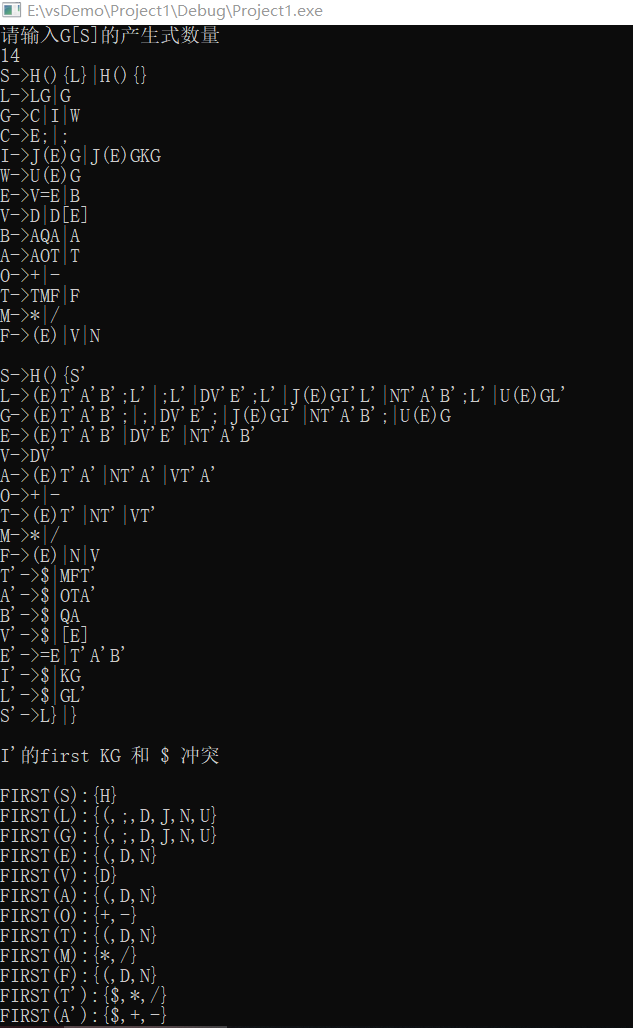
T->TMF|F

M->\*|/

F->(E)|V|N

（识别if else 以及单独的if）

会产生冲突



可以看到

I’的first的KG 和$冲突了

由于水平有限，并未解决