**国家人才培养模式创新实验区、云南省教学团队、云南大学软件学院教育创新基金联合资助项目**

**实验二  
判定LL(1)文法**

姓 名： 何志颖

学 号： 20151120237

专 业： 软件工程

任 课 教 师 ： 柳青

编 制 时 间 ： 2018年6月26日

成 绩：

云南大学软件学院

2018年6月制表

目录

[1、 实验目标 3](#_Toc1372635984)

[2、 实现方案 4](#_Toc1932019031)

[1）从键盘读入某个文法所有产生式。 4](#_Toc1589359860)

[2）判断该文法是否是LL（1）文法。 4](#_Toc822965561)

[3） 识别所有终结符和非终结符， 4](#_Toc860198840)

[4）从键盘读入要识别的符号串。 4](#_Toc1703492901)

[5）由程序根据分析表进行预测分析得出结果。 4](#_Toc1289889341)

[3、 实现步骤 5](#_Toc660571884)

[4、 结构算法 6](#_Toc1647824495)

[计算first集 7](#_Toc2065609028)

[计算follow集 8](#_Toc1293240799)

[计算select集 9](#_Toc1404127617)

[判断是否LL(1) 10](#_Toc101194397)

[5、 测试用例 12](#_Toc1586358844)

[6、 运行结果 12](#_Toc340542637)

[7、 实验体会 14](#_Toc869667481)

1. **实验目标**

理解LL(1)文法的判定以及构造LL（1）预测分析表。

语法分析的作用是识别由词法分析给出的单词符号序列是否是给定文法的正确句子（程序）。

常用的语法分析分两大类：

一、自顶向下分析

二、 自底向上分析

其中自顶向下分析，就是从文法的开始符号出发企图推导出与输入的单词串完全匹配的句子，若输入串是给定文法的句子，则必能推出，反之必然出错。

当我们需选用自顶向下分析技术时，首先必须判别所给文法是否是LL（1）文法。因而我们对任给文法需计算FIRST、FOLLOW、SELECT集合，进而判别文法是否为LL（1）文法。

1. **实现方案**

1）从键盘读入某个文法所有产生式。

2）判断该文法是否是LL（1）文法。

若不是则做消除左递归和提取左公因子。若仍不是LL（1）文法则该文法不能用预测分析法来分析。

1. 识别所有终结符和非终结符，

构造所有非终结符的First和Follow集，由程序自动构造LL（1）分析表 。

4）从键盘读入要识别的符号串。

5）由程序根据分析表进行预测分析得出结果。

1. **实现步骤**

（1）FIRST集的计算

（2）FOLLOW集的计算

（3）SELECT集的计算

（4）LL（1）预测分析表的构造

1. **结构算法**

文法分析中最的用的技术是求以下种集合(终结符集)的技术，这三种集合的具

体定义如下(其中β ∈ (VN ∪VT) )\* ，A ∈ VN):

First (β)={a ∈VT|β \*  a…} ∪ {if β \* λthen {λ } else Ø}

Follow(A)={a ∈ VT |S + …Aa…}∪{if S \*…A then {#} else Ø }

Select(A → β)=First (β)， 当First(β)不含有 λ

=First(β)-{λ}∪Follow(A)，当First(β)含有 λ

其中#表示输入的结束符号，first(β)表示β串所能推导的终极符串的头终结符集，如果β能推导出空串，则令First(β)包括 λ。 follow(A)表示所有那些终结符的集合,这些终极在某个句型中出现在A的紧后面。

## 计算first集

1.初始化所有的非终结符号的first集为空;

2.置changed=1

3.如果changed为1,转4,否则转结束.

4.changed=0;i=0;

5.如果i<rules.GetSize()转6,否则转3

6.从左到右分析第i个非终结符号的每个规则的右部。如果遇到一个终结符号,则把这个非终结符号加到第i个非终结符号的first集中,加入后若改变这个非终结符号first集则置changed为1,接着分析其它规则;如果遇到一个非终结符号则把这个非终结符号的first集并到第i个非终结符号的first集当中,此时若改变了第i个非终结符号的first集置,changed为1,若这个规则可以导出空,则继续分析这个规则的后面的符号。

7.i=i+1,转5

## 计算follow集

1.初始化所有的非终结符号的follow集为空;

2.置changed=1

3.如果changed为1,转4,否则计算结束.

4.changed=0;i=0;

5.如果i<rules.GetSize()转6,否则转3

6.从左到右分析第i个非终结符号的每个规则的右部。对于在这个规则右部的每一个非终结符号 1)计算这个符号后面符号串的first集,把它并入到这个符号的follow集,如果并运算改变了这个符号的follow集,则置changed为1; 2)计算这个符号后面的符号串能否推出空,若能推出空,则把第i个非终结符号的follow集并入到这个非终结符号的follow集,如果改变了这个非终结符号的follow集,

则置changed为1.

7.i=i+1,转5

## 计算select集

每个规则的select计算如下:如果这个非终结符号右部的串能导出空,则这个规则的select集为这个规则左部非终结符号follow集并上这个规则右部串的first集,并去掉空串;如果这个规则的右部导不出空,则这个规则的select集为这个规则右部串的first集.

## 判断是否LL(1)

计算任意一个非终结符号的任意两个规则的select集的交集是不是为空。如果存在某一个非终结符号的某两个规则的select集的交集为不为空,则该文法不能用LL(1)方法分析。

**输入的产生式为：**

**S->aB**

**S->c**

**A->bS**

(1)FIRST集

FIRST（S）=[a]

FIRST（aB）=[a]

FIRST（c）=[c]

FIRST（A）=[b]

FIRST（bS）=[b]

1. FOLLOW集

FOLLOW(S)=[#]

FOLLOW(A)=[#]

1. SELECT集

SELECT(**S->aB**)=[a]

SELECT(**S->c**)=[c]

SELECT(**A->bS**)=[b]

那么由于SELECT(**S->aB**)=[a]和 SELECT(**S->c**)=[c]的交集为空，所以此文法是LL(1)文法。

1. **测试用例**

S->aBZ

Z->bBS

Z->空

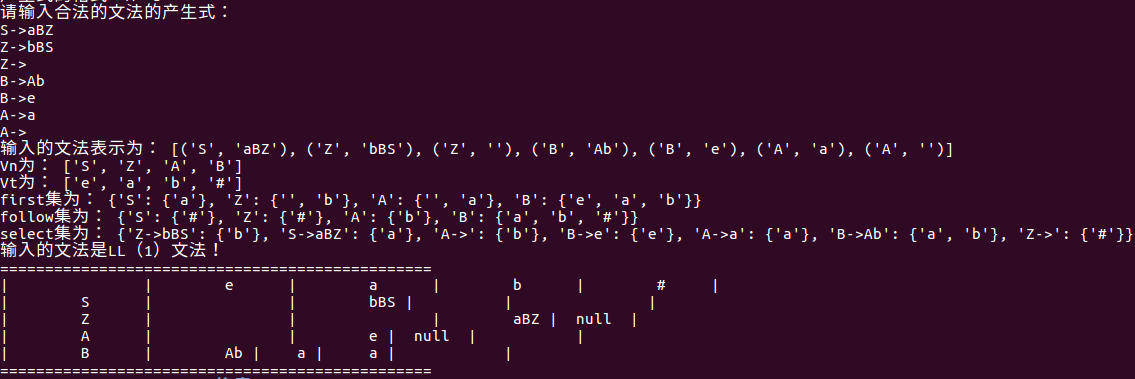
B->Ab

B->e

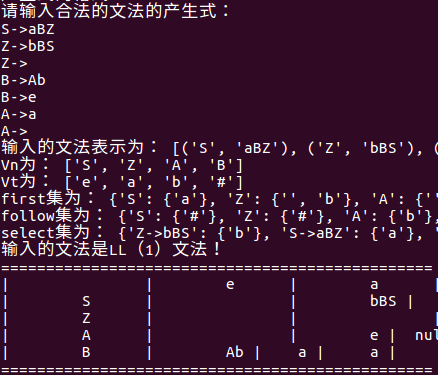
A->a

A->空

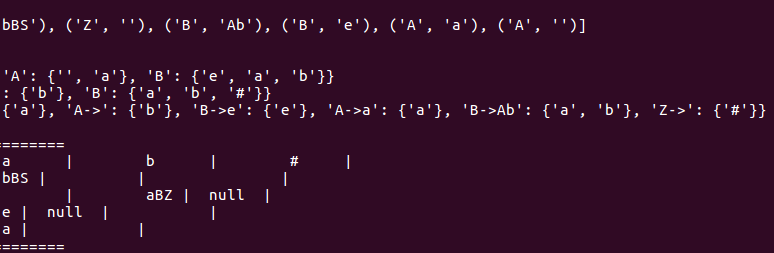
1. **运行结果**

总结果截图

结果左部截图



结果右部截图



1. **实验体会**

通过本次实验，我熟悉了LL（1）文法，掌握了LL（1）文法判定的方法和流程，具体编程实现了FIRST集、FOLLOW集、SELECT集的计算，加深了对于语法分析过程中自定向下分析方法的第一步，即判定给定文法是不是LL（1）文法的理解，为后面的语法分析做了坚实的铺垫，更加深入理解编译原理的语法分析过程中重要的一环。