**LL（1）设计说明文件**

姓名：叶倩琳

班级：软工1706班

学号：201706061330

提交日期：2019年12月9日

**一、实验目的**

构造LL(1)文法的预测分析表。

**二、实验要求**

1. 用C++ 实现；

2. 编程构造如下LL(1)文法的预测分析表，文法G[A]：  
 (0) A→baB (1) A→ε (2) B→bN (3) B→a (4) N→aBbb (5) N→b

3. 预测分析表输出到output.txt文件。

4. 提供名为LL1.exe的执行文件实现上述功能，还需提供源文件以及名为设计说明.doc的说明文件。

**三、实验过程**

**（1）流程图**

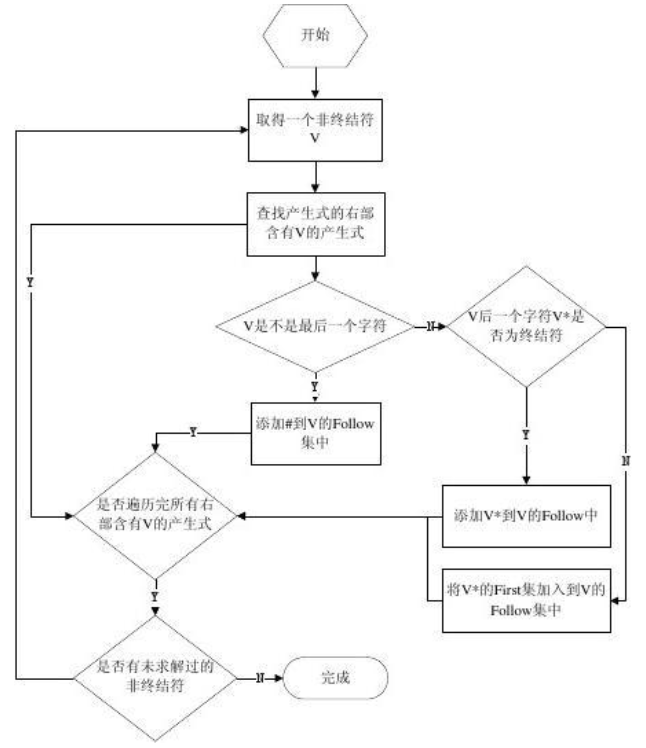


图3-1-1 LL(1)构建预测分析表流程图

1. **概念介绍**

LL（1）第一个“L”, 代表从左向右扫描单词;第二个“L”,代表产生的是最左推导,“1

”代表向前查看（lookahead）一个单词。要推导预测分析表就要引入First集合和Follow集合。

1. **First集合**

【概念】

设 G =（VT，VN，P，S）是上下文无关文法，对a∈(VT U VN)\*，

First（a）= { a->\* aβ, a∈VT,β∈(VT U VN)\*,

或者 a->\*ε 时 a =ε}

【计算】

对所有 x∈VN U VT U {ε} U{v | A->u∈P, 且v是u的后缀}, 则对 x∈V U {ε}，置 First(x)={x}；对其它x，置 First(x)=∅，重复如下过程，直到所有 First 集合没有变化为止：

(1) 对于 A->ε ∈P, 置 First(A) = First(A) ∪ {ε}.

(2) 若A->Y1Y2…YK∈P，置First(A)=First(A) ∪ First(Y1Y2…YK).

(3) 对于 Y1Y2…YK∈{v | A->u∈P, 且v是u的后缀}, 其中 K>=1，Yj∈VN∪VT（1<=j<=K）, 若任意j:1<=j<=i-1(ε∈First(Yj)) ∪ε∉First(Yi), 其中1<=i<=K ，则令 First(Y1Y2…YK) = U First(Yj) - {ε}，否则，若任意j:1<=j<=K(ε∈First(Yj)), 则令First(Y1Y2…YK) = U First(Yj).

1. **Follow集合**

【概念】

设 G =（VT，VN，P，S）是上下文无关文法，对每个A∈VN，

Follow(A) = { a | S# =>\*αAβ# 且 a∈First(β#)，α,β∈(VT∪VN)\* } （# 代表输入单词序列右边的结束符）

【计算】

置 Follow(S) = {#}，置所有其它的 Follow 集合为∅；重复如下步骤，直至所有Follow集不再变化为止：

对于 X->αAβ∈P，

(1) 把 First(β) - {ε} 加至 Follow(A);

(2) 若有ε∈First(β)，则把 Follow(X) 加至 Follow(A) 中．

1. **Select集合**

【概念】

预测集合。

【计算】

设 G =（VT，VN，P，S）是上下文无关文法。对任何产生式 A->α∈ P，其预测集合Select (A->α) 定义为：

如果ε∈first(α)，那么Select (A->α) = first (α)；

如果ε∈first (α)，那么Select (A->α) = ( first (α)–{ε} )∪ follow(A)

NOTE：定义表明 a∈VT∪{#}

意义：下一字符属于Select (A->α)时，可以选用A->α.

1. **程序设计的主要思路**

【变量】

vector<char> finish\_sign;

vector<char> nofinish\_sign;

vector<pair<char, string> > grammer;//文法

map<char, set<char> > first\_set;

map<char, set<char> > follow\_set;

map<char, vector<string> > table;

int n; //文法数

【方法】

bool isNofinish\_sign(char c); 判断是否为非终结符

void input(); 读入

void getfirst(char t); 构造First集合

void getfollow(char t); 构造Follow集合

void settable(); 构建预测分析表（Select集合）

Main函数; 主函数入口

【求FIRST集算法思想】

若产生式右部第一个字符为终结符，则将其计入左部first集；

  若产生式右部第一个字符为非终结符，则：

▲求该非终结符的first集

  ▲将该非终结符的非$first集计入左部的first集

▲若存在$，则将指向产生式的指针右移

▲若不存在$，则停止遍历该产生式，进入下一个产生式

▲若已经到达产生式的最右部的非终结符，则将$加入左部的first集

  处理数组中重复的first集中的终结符。

【求FOLLOW集算法思想】

  对于文法G中每个非终结符A构造FOLLOW(A)时连续使用下面的规则,直 到每个FOLLOW不在增大为止：

▲对于文法的开始符号S,置#于FOLLOW(S)中;

▲若A->aBb是一个产生式,则把FIRST(b)\{ε}加至FOLLOW(B)中;

▲若A->aB是一个产生式,或A->aBb是一个产生式而b=>ε(即ε∈ FIRST(b))则把FOLLOW(A)加至FOLLOW(B)中。

【生成预测分析表的算法思想】

  ▲对文法G的每个产生式A->a执行第二步和第三步;

  ▲对每个终结符a∈FIRST(a),把A->a加至M[A,a]中;

  ▲若ε∈FIRST(a),则把任何b∈FOLLOW(A)把A->a加至M[A,b]中;

  ▲把所有无定义的M[A,a]标上出错标志。

**（4）程序运行顺序**

1）先读入txt文件，读取文法、终结符、非终结符集合；

8

S->AB

A->Da

A->$

B->cC

C->aADC

C->$

D->b

D->$

2）构造First集合：

First(ε) = {ε}

First(Da) = {b, a}

First(aADC) = {a}

First(b) = {b}

Follow(A) = {c,b,a, #}

Follow(C) = {#}

Follow(D) = {a, #}

3）构造Follow集合：

Follow(S) = {#}

Follow(A) = {c,b,a, #}

Follow(B) = {#}

Follow(C) = {#}

Follow(D) = {a, #}

4）构造Select集合并创建预测分析表

Select(A->Da) = {b,a}

Select(A->ε) = {c,b,a, #}

Select(C->aADC) = {a}

Select(C->ε) = {#}

Select(D->b) = {b}

Select(D->ε) = {a, #}

**四、实验结果**

（1）输入文法到inout.txt

8

S->AB

A->Da

A->$

B->cC

C->aADC

C->$

D->b

D->$

（2）运行LL1.cpp

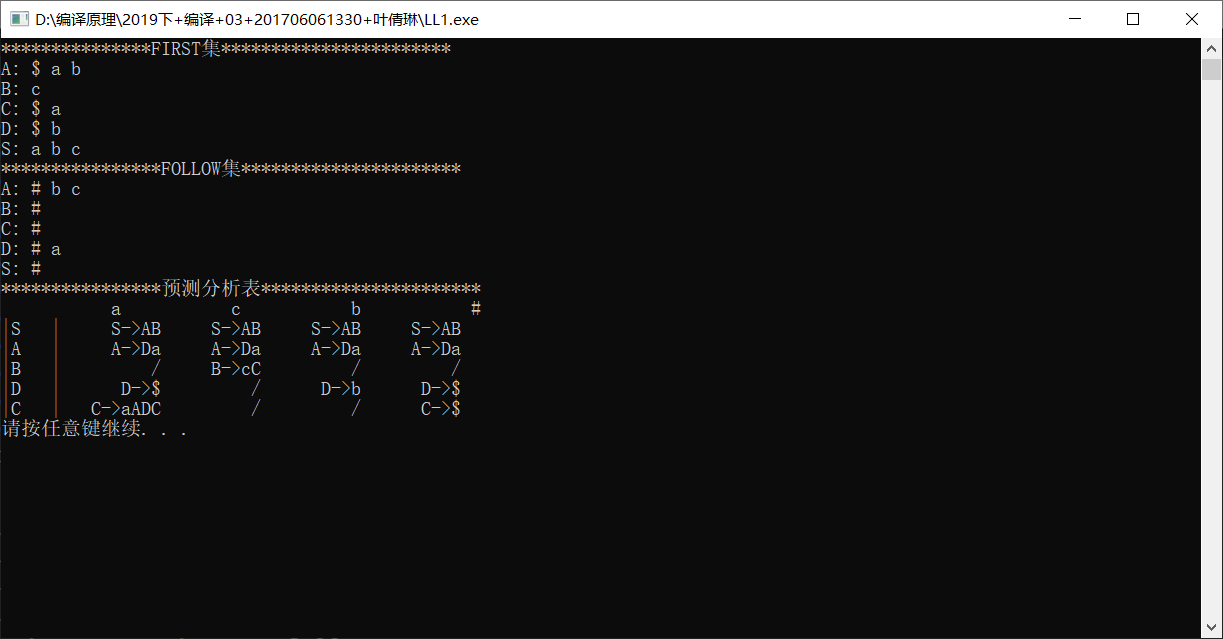


图4-2-1 LL(1)实验结果

（3）程序输出结果到output.txt

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*预测分析表\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

a c b #

S S->AB S->AB S->AB S->AB

A A->Da A->Da A->Da A->Da

B / B->cC / /

D D->$ / D->b D->$

C C->aADC / / C->$

**五、实验体会**

通过本次实验，我熟练掌握了LL（1）分析，对First、Follow、Select集的构造有了更加深入的理解，同时，也能够根据由Select集合推出的预测分析表对输入串进行分析。