## 安徽大学图标

## 编译原理实验报告

## 实验名称：LL(1)文法判断和预测分析法实现

## 姓名： 何铭春

## 学号： E21514033

## 一.实验内容

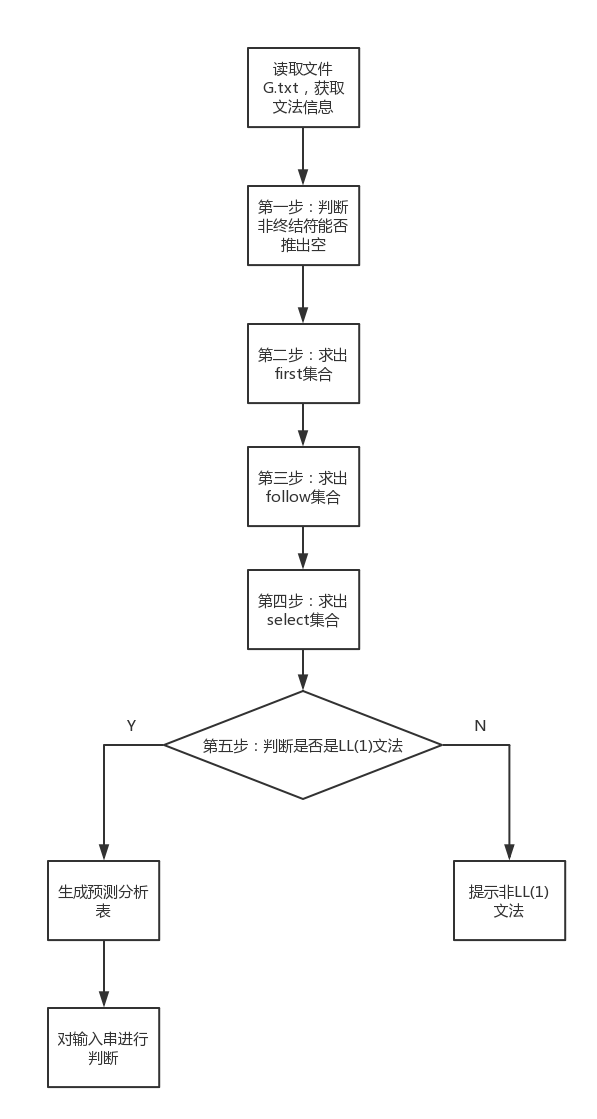
LL(1)五步判别

构造预测分析表

对输入的句子进行语法分析

使用词法分析器将代码段转为输入串

## 二.基本思想（流程图）



## 三.核心算法(流程图)

### 1.LL(1)五步判别法①判断非终结符能否推出空

数据结构设计：

保存非终结符和终结符对应数字

noterminal={'E':0,'A':1,'T':2,'B':3,'F':4}

terminal={'i':0,'(':1,')':2,'#':3,'+':4,'\*':5}

保存语法式子

G=["E->TA","A->+TA|&","T->FB","B->\*FB|&","F->i|(E)"]

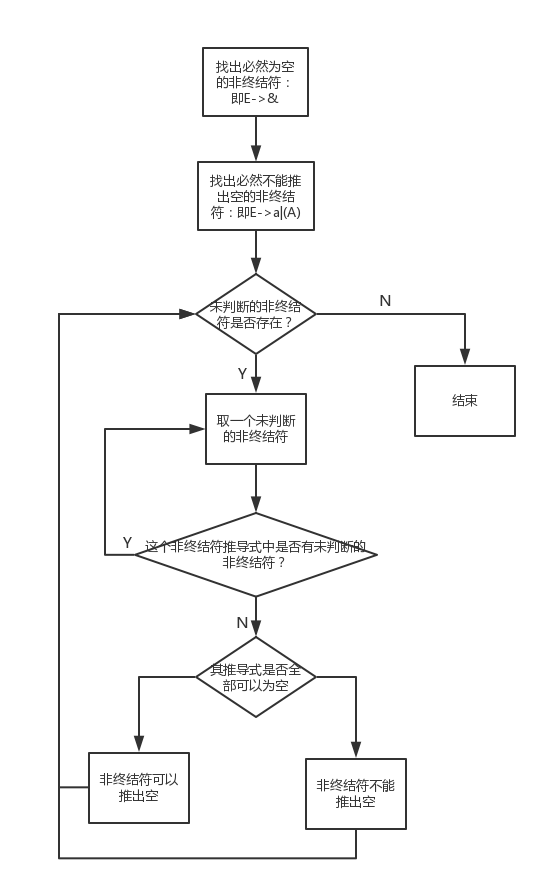
三个集合，分别保存未判断的非终结符集合、已经被判断能推出空的集合、一定不能推出空的集合。

inDoubt=set(list(noterminal.keys()))

null=set([])

unNull=set([])

流程图：



### 2.LL(1)五步判别法②求first集合

数据结构设计：

保存非终结符和终结符对应数字

noterminal={'E':0,'A':1,'T':2,'B':3,'F':4}

terminal={'i':0,'(':1,')':2,'#':3,'+':4,'\*':5}

保存语法式子

G=["E->TA","A->+TA|&","T->FB","B->\*FB|&","F->i|(E)"]

三个集合，分别保存未判断的非终结符集合、已经被判断能推出空的集合、一定不能推出空的集合。

inDoubt=set(list(noterminal.keys()))

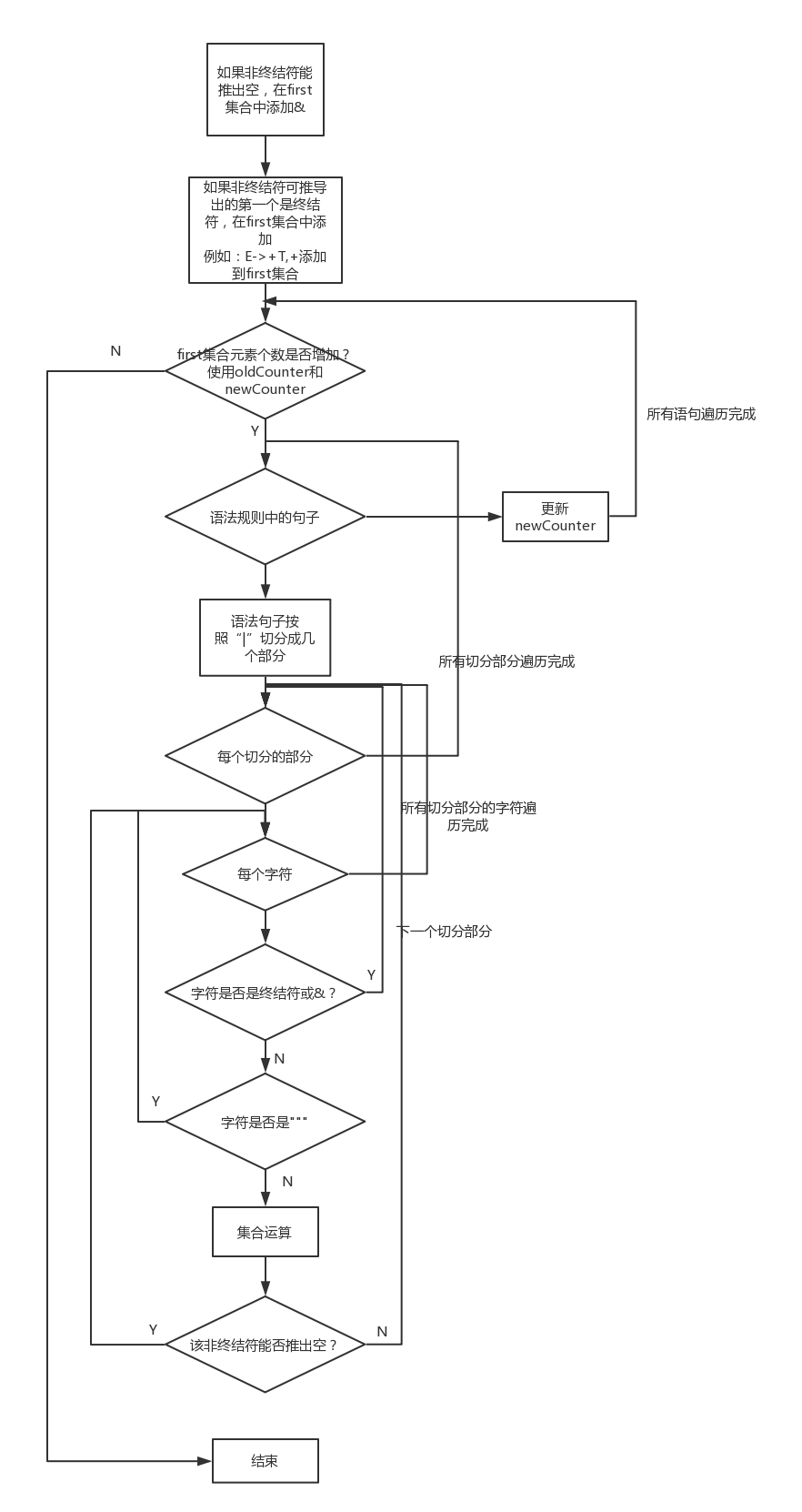
null=set([])

unNull=set([])

一个列表，每个非终结符对应其中一个集合

first=[set([]),set([]),set([]),set([]),set([])]

流程图：



### 3.LL(1)五步判别法③求follow集合

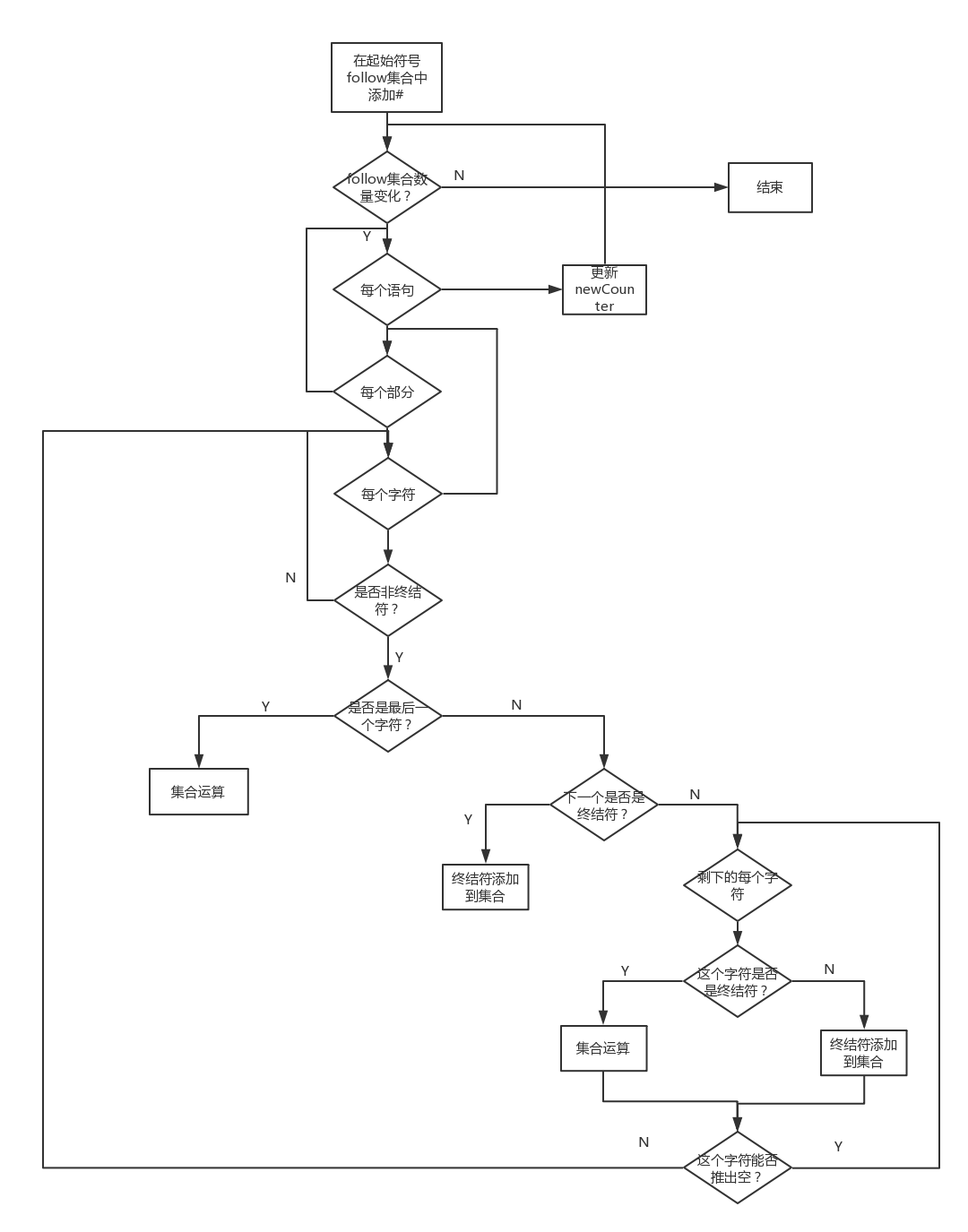
数据结构设计：

新增：start=”E”保存开始符号

一个列表，内有非终结符个数集合，每个非终结符对应一个集合。

follow=[set([]) for i in range(0,len(noterminal))]

流程图：



### 4.LL(1)五步判别法④求select集合

数据结构设计：

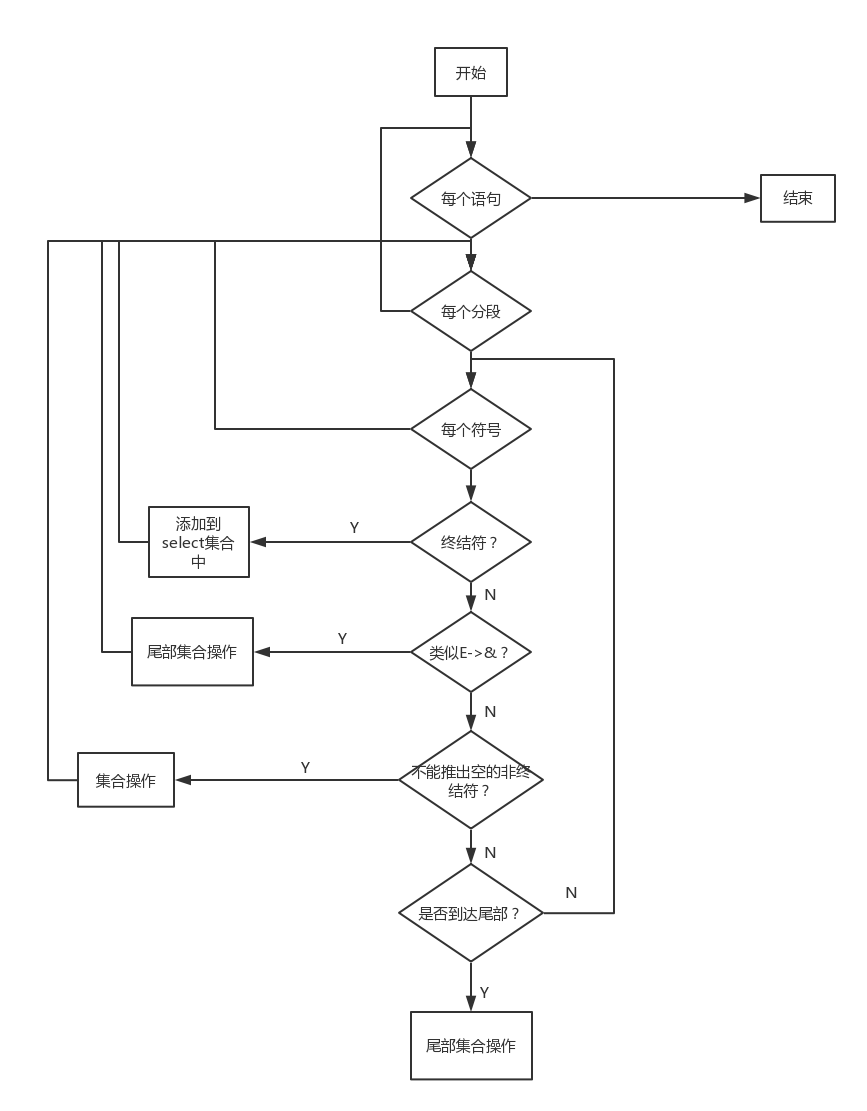
Select集合，保存形如E->AT={a,b}的字符串

select={}

原本语法中F->i|(E)分开为F->i和F->(E)

GStr=[]

流程图：



### 5.LL(1)五步判别法⑤判断是否为LL(1)文法

数据结构设计：

承接上面数据结构不变

主要思路：

取相同左部的select集合做列表的合并，列表的合并是保留重复元素的。

最后将这个列表转化成集合，集合不保留重复元素，比较前后变化。如果变化，那就不是LL(1)文法，反之是LL(1)文法。

### 6.生成预测分析表

数据结构设计：

建立一个二维数组：

M=[(["0"]\*len(terminal)) for i in range(len(noterminal))]

代码实现：

简单根据select集合就能完成填充

def getM(terminal,noterminal,select,M):

for i in select:

line=i.split('->')[0]

for column in select[i]:

M[noterminal[line]][terminal[column]]=i.split('->')[1]

### 7.分析过程表

数据结构设计：

堆栈

stack="#E"

保存输入的字符串

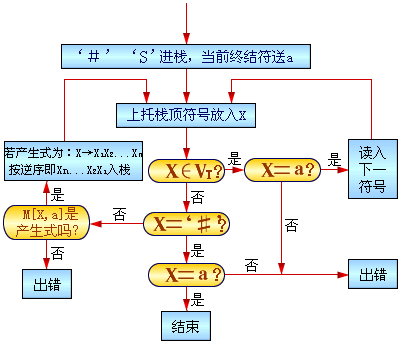
strFor=input("请输入要判别的字符串:")

strFor=strFor+'#'

记录当前步骤数

counter=1

流程图：



## 四.实验结果

按书上格式输出了五步判别法中的数据。

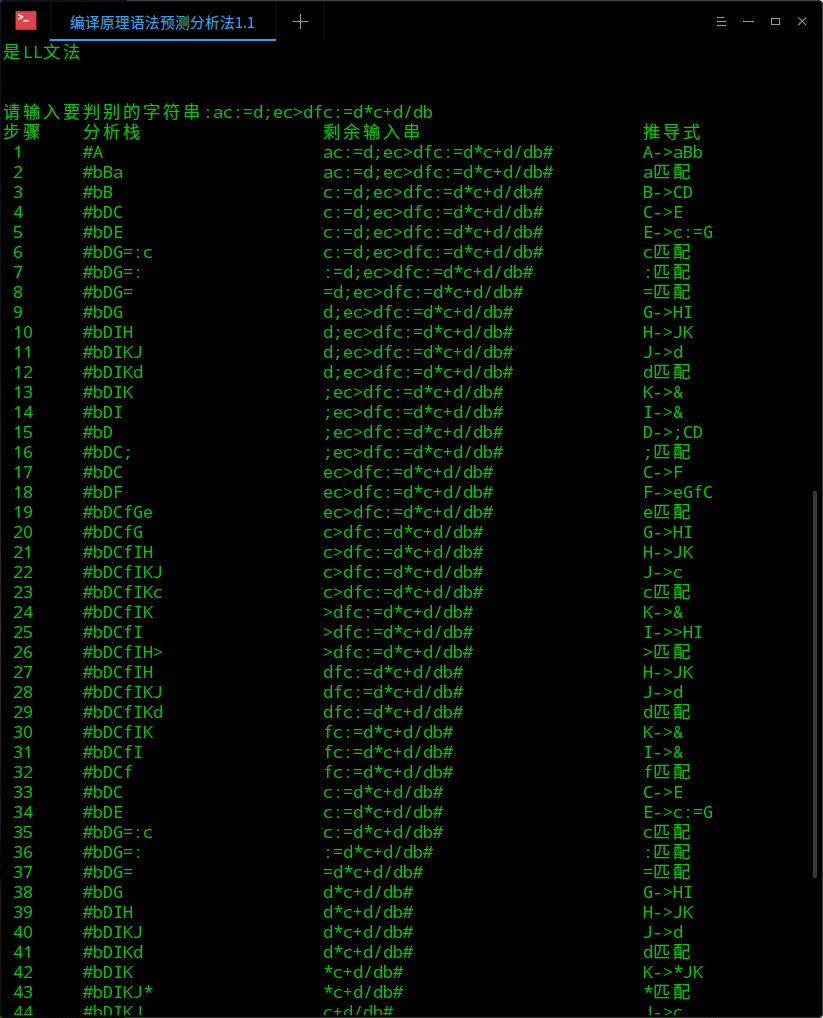
然后输入判别字符串，进行预测分析法的输出。

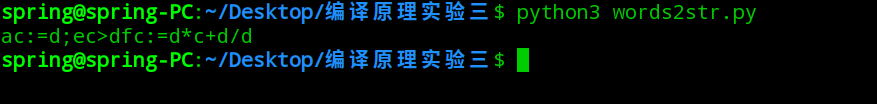
书本上词法分析实例：



PPT上实例：







## 五.实验心得

1.选择合理的数据结构是程序成功的一半。我在长时间思考之后选择使用Python字典（类似C++枚举类型）来让非终结符以及终结符和数字对应起来。用于first和follow以及select、M数组下标。有了这个就能够将非终结符相对应的数组更易于连接起来。

2.在判断是否能推出空以及求first和follow集合时，我研究书上算法，先找出能判断和找出first以及follow集合的非终结符。例如E->&，那么E明显可以推出空。如果T->aB|cA，T明显不能推出空。然后在循环对未判断的非终结符进行判断。

3.算法分模块实现，对于算法的每个步骤，我都设定一些值让这个单一步骤能够运行，便于开发和调试。并且模块中不定义数据，只在主模块中定义数据，其他模块只负责函数调用。

4.模块功能实现时，有些步骤过于复杂，需要在编写代码前认真思考如何实现这个循环和选择。

5.将词法分析和语法分析联系在一起，能够起到判断代码段是否符合词法和语法规则。