# 文法构造

1. 问题描述

# ·实验目的

设计文法描述的数据结构，实现文法的输入和输出，加深对文法的理解。

# ·实验内容

需要实现的功能：

1）输入文法（文法描述存储在文本文件中，文件名作为命令行输入）；

2）输出文法（标准输出设备）；

# ·实验要求

文法描述文件包含如下内容：

1. 非终结符集合；
2. 终结符集合；
3. 规则集合；
4. 指定开始符号。

## ·文件结构：

1. 非终结符个数；
2. 所有非终结符，空格分隔；
3. 终结符个数；
4. 所有终结符，空格分隔；
5. 规则个数；
6. 所有规则，每行一个规则，规则输入格式：左部，右部符号数，右部符号，空格分隔；
7. 开始符号。

## ·样例

### 描述：

算术表达式文法G =（VN，VT，P，S）其中：

VN = { E, T, F }

VT = { i, +, \*, (, ) }

P = { E → E + T | T  
 T → T \* F | F  
 F → i | (E) }

S = E

### 输入：

3

E T F

5

i + \* ( )

6

E 3 E + T

E 1 T

T 3 T \* F

T 1 F

F 3 ( E )

F 1 i

E

### **输出：**

非终结符集：

E T F

终结符集：

i + \* ( )

产生式集：

0: E -> E + T

1: E -> T

2: E -> T \* F

3: T -> F

4: F -> ( E )

5: F -> i

开始符号：

E

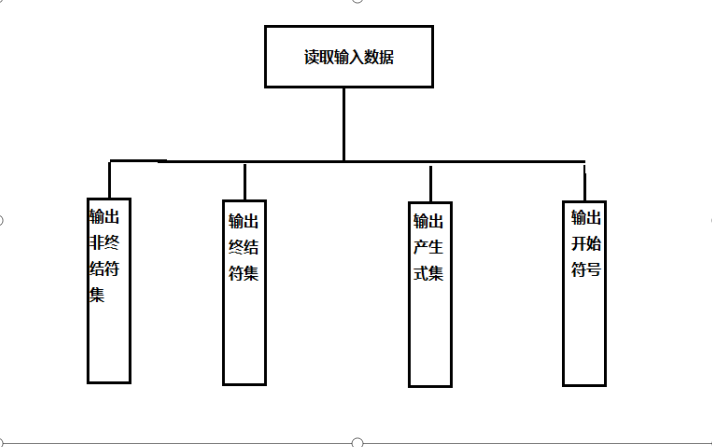
1. 分析模型

这个实验需要我们对文法进行分析，也就是对语法结构的规则进行分析和输出。对于输入的一个文法，可以设计结构体数组分别存储非终结符、终结符以及产生式规则，然后依次输入并按照要求格式输出。再具体来讲，非终结符是用来表示语法范畴的，它是一个类记号而不是一个个体记号；终结符号是组成语言的基本符号，在程序语言中就是基本字、标识符、常数、算符和介符等，从语法分析的角度来看，终结符号是一个语言的不可再分的基本符号；产生式是定义语法范畴的一种书写规则；开始符号是一个特殊的非终结符号，它代表所定义的语言中我们最感兴趣的语法范畴，这个语法范畴通常称为“句子”。

该程序的数据流图即为输入字符串->按照文法存储->按照格式输出

1. 设计模型

分别设计三个结构体数组存储终结符、非终结符以及产生式规则，同时设计一个字符变量存放开始符号；对于每个结构体数组，需要包含终结符或非终结符或产生式规则个数以及存储相应终结符或非终结符或产生式规则的字符数组；特别对产生式规则，可以在输入时进行判断，数字之前为产生式左半部分，遇到数字即加入->，然后接收产生式的右半部分，最后将输入的内容按照实验要求输出即可。



4、系统实现

通过输入函数input把所需要的信息（非终结符集、终结符集、产生式集、开始符号）这些存储在构造的文法结构里。最后通过output函数输出文法结构里的信息。

通过main函数调用input函数，输入的同时对字符串进行处理和存储，再调用output函数输出即可。

1. 测试数据及测试结果

datain.txt内容：



cmd运行命令:wenfa.exe<datain.txt>dataout.txt

输出内容如下：



1. 实验总结

1、文件的重定向，因为第一次接触这个概念，对文件的重定向很陌生，调试程序时是从控制台输入的。上网查询资料后了解其大致用法。编译的exe文件<datain.txt>dataout.txt，可以实现把datain.txt的数据作为程序的输入数据，程序运行结果可以保存在dataout.txt中。

2、结构体的使用。一开始对于非终结符、终结符以及表达式的存储没有太多头绪，感觉无从下手。经过课堂上老师的点拨了解到用结构体存储这些内容，对于程序的规范以及以后的重用都是很有用的。

总之，通过本次实验，设计、编写、调试、运行一个文法构造程序，实现对一个文法的分析与构造，对文法有了更全面深入的了解。