**编译原理实践第4次课**

**（自顶向下的语法分析）**

1. 给定下面的文法:

E → TG

G → +TG | ε

T → FS

S→ \* F S| ε

F → (E) | **id**

其中E为开始符号.

输入字符串: id + id \* id (假设各个词法单元以空格隔开, 不会出现id+id\*id情况)

请使用递归下降法生成语法树，并将语法树打印成字符串形式，语法树的打印成字符串使用中括号嵌套的形式. 例如id + id的语法树可以为字符串:

E->TG

T->FS

F->id

S->e

G->+TG

T->FS

F->id

G->\*FS

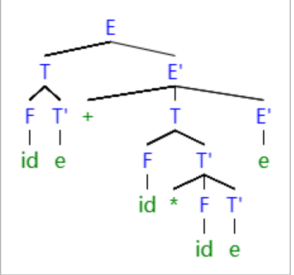
F->id

S->e

G->e

[E [T [F [id]] [S[e]]] [G [+] [T [F [id]][G[\*][F [id]] [S[e]]]] [G[e]]]]

其对应的树状图如下所示:



注意: 1)使用e表示ε，并假设文法中不会出现符号e和中括号.

2) <http://mshang.ca/syntree/> 可以使用该网站, 能够将字符串形式生成树状图, 以检验是否正确.

**提交代码与readme.pdf（readme.pdf里需要有对于代码的使用，以及给定某个例子，其运行结果截图）**