**2017年《编译原理》第2次小测验解析**

**1. 文法 G1 的开始符号是 ① ，非终结符包括 ② ，**

**终结符包括 ③ 。 该文法表明运算‘=’比‘+’的优先级 ④ ，**

**运算‘=’的结合性是 ⑤ 结合的。**

M→ M + A | A

A→ V = A | V (G1)

V→ x | y | z

**答：**① M ② M, A, V ③ +, =, x, y, z ④ 高 ⑤ 右。

**解析：**

本题考察的第一个知识点：上下文无关文法（CFG）的基本概念。

在任何正确的CFG定义中，只有非终结符才需要、必须用产生式来定义其所代表的语法结构，所以可按以下规律来判别文法的终结符和非终结符：

1. 非终结符**绝对会**出现在产生式左边：

因此，上述文法G1的非终结符包括 M, A, V 。

1. 终结符**只能**出现在产生式右边（绝不会出现在产生式左边）。

还有一个约定：第一个产生式左部非终结符就是文法开始符号。

本题考察的第二个知识点：文法符号（只考虑***终结符***）的优先级。

在学习消除文法二义性的第一种方法时曾提到：“越接近文法开始符号S的文法符号的优先级越低”。这个陈述中涉及一个基础性概念：

某文法符号 X 与开始符号 S 的距离。

该距离指：从S要推导出包含 X 的文法符号序列，所需**最少直接推导**步骤数。如文法G1中，

M 与 开始符号M 的距离为0，因为 M 可0步推导出其自身；

+ 与 M 的距离为1，因为 M => M+A，需要最少一步直接推导；

= 与 M 的距离为2，因为 M => A => V=A，需要最少两步直接推导；

因此，文法表明了 = 的优先级高于 +。

本题考察的第三个知识点：文法符号（只考虑***终结符***）的结合性。

在学习消除文法二义性的第一种方法时曾提到：“对于A→αAβ，其右部中，若A在终结符a左边出现（即β中包含a），则终结符a具有左结合性质；……”。这句话除了强调【 “结合性”必须通过“递归定义非终结符A” 】之外，还强调 a 和 A在产生式右部中的相对位置。为便于理解，可将A的产生式简单地写成两种情况：

Case 1： A 🡪 … A … a … ，且 a 右边无 A

该产生式表明终结符 a 是左结合的，如课堂例子中 E 🡪 E + T；

Case 2： A 🡪 … a … A … ，且 a 左边无 A

该产生式表明终结符 a 是右结合的，如课堂例子中 F 🡪 - F；

不满足上述任何条件者，则表明不了a的结合性，或者表明 a 无结合性，如

A 🡪 a A a A 🡪 A a A 等。

**2. 对于文法 G1 产生的句子 z + x = y ，画出该句子的 (a) 分析树、 (b)语法树。**

**解析：** 本题考察推导、两种树的基本概念。

M→ M + A | A

A→ V = A | V (G1)

V→ x | y | z

（即使你不知道 + 和 = 的优先级、结合性，也不影响本题的解答，除非你连树的概念都不知道或未理解）

分析树：可（用构造过程）直观地表示推导过程，也可（用树形）表示句型结构。

本题给定句子的最**右**推导过程为：

M => M + A => M + V=A => M+V= V => M+V= y

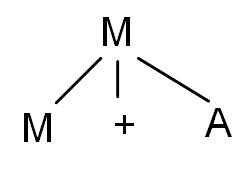
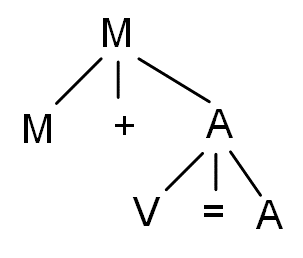
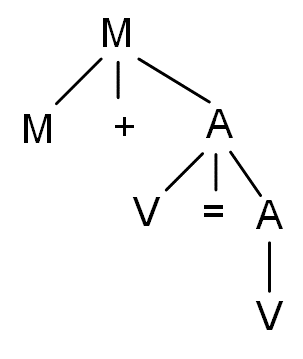
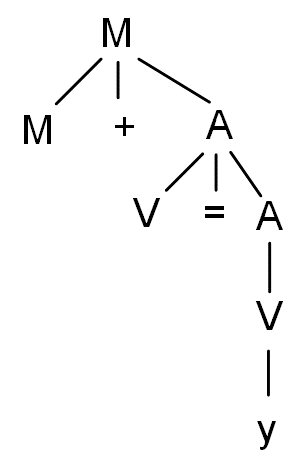
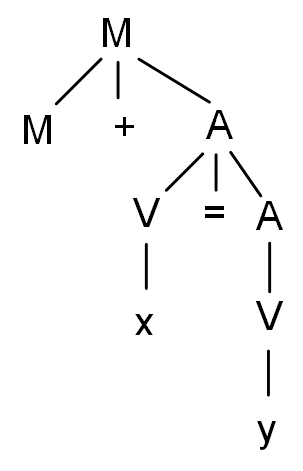
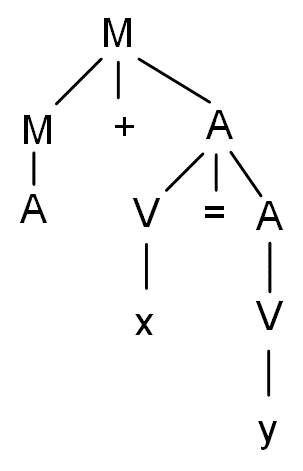
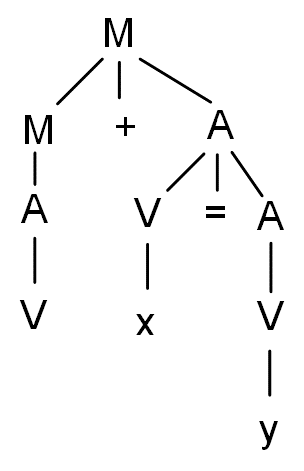
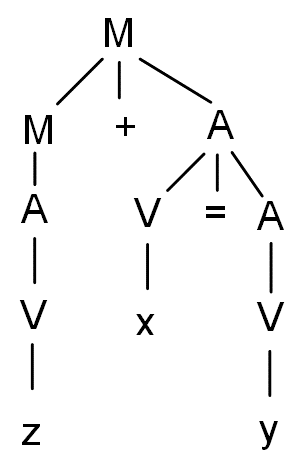
=> M+ x =y => A +x=y => V +x=y => z + x = y

注：下滑线标记了各右句型中将被展开的非终结符，矩形边框包围了前一句型最右边的非终结符被展开的结果。

据此，分析树的构造过程如下列各图所示。

**注1：请关注上述推导的每个句型与分析树的对应关系**。

**注2：根据本题要求，只需给出最终的分析树即可。**



[ 也可用最左推导构造分析树 ]

语法树：可表示句子结构，但表示不了推导过程。这类树有多种构造过程。

无论如何构造，牢记语法树的根本：父节点表示运算、儿子节点表示相应的操作数。

构造方法1：若你根据文法确定出了优先级和结合性，则可据此 *(递归地)* 分解句子结构。并将分解结果（直观地）表示为语法树即可。

根据题目1的答案，文法G1表明了 = 的优先级高于+，所以句子 z +x=y 的最后一个运算是 +， 第一个运算是 =，即整个句子结构就是：z + (x=y)。这样最终语法树的树根就是 +，其左孩子（操作数）就是z，右孩子（操作数）就是 x=y。

构造方法2： 观察分析树表示的句子结构（因为两种树都表示了句子结构，所以可采用“句子结构”为桥梁）

对于最终的分析树，采用“***从上往下***”看 (第2层) ，可知句子整体结构是二元加，这必然是句子中最后一个计算，所以语法树的树根用 + 标记，其操作数为：

1. 左操作数为 z（分析（子）树 M-A-V-z的叶子）;
2. 右操作数是一个赋值表达式，因此该子树的树根为 =，其操作数为：
3. 左操作树为 x（分析树 V-x 的叶子）；
4. 右操作树为 y（分析树 A-V-y 的叶子）

综上即可得到完整语法树。

构造方法3：借助最终分析树，执行LR 分析的驱动器算法，并且： “归约”时，构造产生式右部对应的语法树（子树），须遵守操作符为父亲节点，操作数为儿子节点。【具体过程请阅读P170~171给出的树的语法制导翻译】

如本题中的句子 z + x =y，其分析过程中的几个格局之分析栈及树节点如下：

1. 移进z并归约为M，移进 + x，x归约为V，再移进 =y，y归约为A之后的分析栈：

(栈底) M + V = A (栈顶，下同)

语法树的节点：(z) (+) (x) (=) (y) 括弧表示节点或子树，此时树高均为1。

(z归约得到的) M 的属性就是 z 对应的语法树节点，其他类似。

 (b) V=A归约为A后的分析栈： M + A

语法树的节点：(z) (+) ( (=) (x) (y) ) 此时 (=) 为子树的根，(x) 和 (y) 为其儿子

(c) M+A归约后的分析栈： M

语法树的节点：( (+) (z) ( (=) (x) (y) ) )

此时 (+) 为树根，(z)为左儿子， (=)为右儿子

{实际上，利用扩充了语义分析的LR分析器的话，语法树节点就是“语义”，用“属性”表示，并存储在语义栈中! 如右侧三个图所示。}

**解答：**【根据题目陈述可知：解答中无需上述各步骤，只需给出最终的树即可】

分析树 语法树

**3.右图的分析树使用了某文法 G2 的全部产生式，**

**(a) 请写出完整文法 G2（注意：D1 与 D2 对应同一个  
文法符号 D，其他类似）；**

**(b) 写出分析树表示的句型；**

**(c) 指出该句型的所有短语、直接短语、句柄；**

**(d) G2 是否为 LL(1) 文法？请给出理由；**

**(e) 若 G2 不是 LL(1) 文法，请给出等价的LL(1)文法。**

**解析：** 本题考察分析树与文法的关系、短语（直接短语/句柄）、LL(1)文法的概念。

问题(a):

根据分析树的概念可知，任何一颗只有父子关系（树高为2）的子分析树，对应一个产生式，且父节点就是产生式左部，其**所有儿子节点**从左到右依次排列就构成了产生式右部。

据此，对整个分析树扫一遍，将每个树高为2的子树都写成对应的产生式，去掉重复的产生式，就可得到对应的文法 G2 如下：

S 🡪 a D B

D 🡪 D d | d

B 🡪 b B | b

另外，题干中指明这个树使用了G2的“全部”产生式，所以G2的全部产生式如上。

注意1：因为可用产生式集合表示CFG，所以非终结符集合、终结符集合、开始符号无需另外专门指出。

注意2：只写文法时，文法符号不用编排序号！！！

问题(b):

每一颗的分析树均对应（表示）一个句型 并表示了其结构，其中：

* （从上往下看）树的形状表示了结构，
* 所有叶子节点（的文法符号序列）（从左到右依次写出来）就是句型。如果均为终结符，则该句型就是句子(如本题就是)。
  + **回忆 句型 的概念！！！**

所以这颗树表示的句型为：addbb （携带下标 ad1d2b2b1也可）。

有同学将句型写成正规式： ad\*b\*，ad+b+  ☹，这是错误的、绝不允许的。因为任何分析树所表示的句子、句型一定是长度、符号均确定的文法符号序列。

问题(c): 考察短语们在分析树上的表现形式。

短语：以某个非终结符为根的子树（含整颗树）的所有**叶子**，从左到右依次写出来所得的符号序列。

据此，在观察分析树后，可找到的短语有：

[**注意：短语、直接短语、句柄都不能写成产生式**]

ad1d2b2b1 相对于树根S 的短语

d1 相对于D2 的短语

d1d2 相对于 D1 的短语

b2b1 相对于 B1 的短语

b1 相对于 B2 的短语

直接短语：所有短语中，位于分析树末端、且只有父子关系的那些子树（树高为2）的  
所有叶子，从左到右依次写出来所得的符号序列.

据此，在观察分析树后，可找到上述短语中的直接短语有：

d1 相对于产生式D🡪d 的直接短语

b1 相对于产生式 B🡪 b 的直接短语

句柄：最左边的直接短语，在树上表现为 **深度优先遍历**过程中，**第一个**遇到的直接短语。

据此，句柄为 d1。

应注意的概念问题：

* 直接短语 一定 是短语， 但短语 不一定 是直接短语!
* 句柄 一定 是直接短语，但直接短语 不一定 是句柄!
* 每个句型的句柄是**唯一的**，但直接短语、短语**至少**有一个!

问题(d)： 考察 LL(1)文法的定义和判别方法.

判别方法1**：**

构造文法的预测分析表，看看是否存在多重定义（即填写了两个或以上的产生式右部）的条目（单元格）。

\*\* 这是基本方法，但必须构造出预测分析表，计算量大，麻烦。

判别方法2**：**

根据推论3.2判断，但只需考察那些有 两个 或更多个候选项的产生式即可。若***每个***此类产生式***均***同时满足推论3.2的三个条件，则文件是LL(1)的，否则不是。

【注意：对于原因不能简单地回答“不符合推论3.2”，而是应明确地给出事实证明：哪个非终结符的产生式不符合该推论的哪个/些条件，只要一条即可.如下面这样】

对于文法G2，D和B的产生式右部各存在2个候选项，先考察D的两个候选项，存在以下两个推导：

D => Dd => dd （先用候选项1推导，再用候选项2推导）

D => d （用候选项2推导）

二者都推导出了 d 开始的序列，不满足推论3.2的条件1。所以：G2不是LL(1)文法。

提示1：既然已经得出了结论，就无需再B的产生式推导+判断了。

提示2：推论3.2的前两个条件的本质：确定同一非终结符的任意两个候选项的 First 集合是否相交？若相交，则文法不是LL(1)。

判别方法3**：** 还有几类特殊文法，一定不是 LL(1) 文法，包括：

* 二义文法
* 有左递归的文法
* 有左因子的文法

考虑文法 G2 的 D 产生式右部，存在（直接）左递归，而B的产生式右部存在左因子，所以 G2 不是 LL(1) 文法。

自己思考：若一个文法不是二义的、没有左递归、没有左因子的话，它是LL(1)文法吗？

问题(e)： 文法 G2 不是二义文法，所以改写的内容就是消除左递归、提取左因子。

当这两种现象同时存在时，一般先消除左递归。当然也可以先提取左因子，但所得文法（可能）会有不同，改写过程也稍啰嗦。

对于文法G2，消除 D 产生式中的直接左递归，提取B产生式的左因子，可得左下文法。该文法中，D、B的产生式也可以写为右边的形式。

D的产生式也可写成：

D 🡪 d D’

D’ 🡪 D | ε

S 🡪 a D B

D 🡪 d D’

D’ 🡪 d D’ | ε

B 🡪 b B’

B’ 🡪 B | ε

B的产生式也可写成：

B 🡪 b B’

B’ 🡪 bB’ | ε

实际上，D描述的是“由d构成的非空串”，B描述的集合类似。

【小测验的解析结束】

【下面给出学习、做题过程中值得注意的若干问题，务必关注并理解】

附录：一些值得注意的问题/错误

* 表示推导的符号 是 =>，不是箭头 🡪 。
* 我们这个课程中，产生式左部与右部之间的连接符是(单横线)箭头 🡪，

既不是等号，也不是 => ，更不是冒号 。

* 分析树/语法树上的父子关系用**实线**连接，**没有箭头 。**
* 文法的开始符号也是一个非终结符 。
* 一个特定文法的开始符号不一定叫 S。
* 产生式的书写中，只有连接产生式‘左部’和‘右部’的箭头🡪、连接候选项的 ’|’ **不是**文法符号，其他符号都是文法符号（要么是终结符，要么是非终结符）
  + 如 +, = 等 **运算符** 是终结符。
* 紧跟在文法符号后面的序号（即下标 0，1，2，…）何时使用？
  + 在分析树上：目的是为了区分同名但不同位置的符号，**且**用于在短语/直接短语/句柄中以示区分。
  + 在产生式右部：目的是便于在语义规则中区分同名但不同位置的符号。
  + 其他情况下无需序号!!!
* 分析树、语法树表示的句子/句型结构：
  + 从树根开始，由上往下看，你能看到每颗子树表示的结构！
  + 若你为同一个句子画出了两种树，可根据上述特点观察两颗树上的每层的结构是否一致！若不一致，则肯定至少有一颗画错了。
* 分析树是本课程非常重要的概念，在语义分析/中间代码生成过程中反复被使用。所以必须熟练地构造（绘制）、熟练地剪句柄。
* 语法树也是一个重要概念，特别是在：
* 写出一个表达式的后缀式时、
* 为正规式构造NFA时，

用语法树表示表达式、正规式的结构（即你对它们中包含运算的分解结果）。

建议：看完并理解后重新做做，直到完整正确为止!

=============================