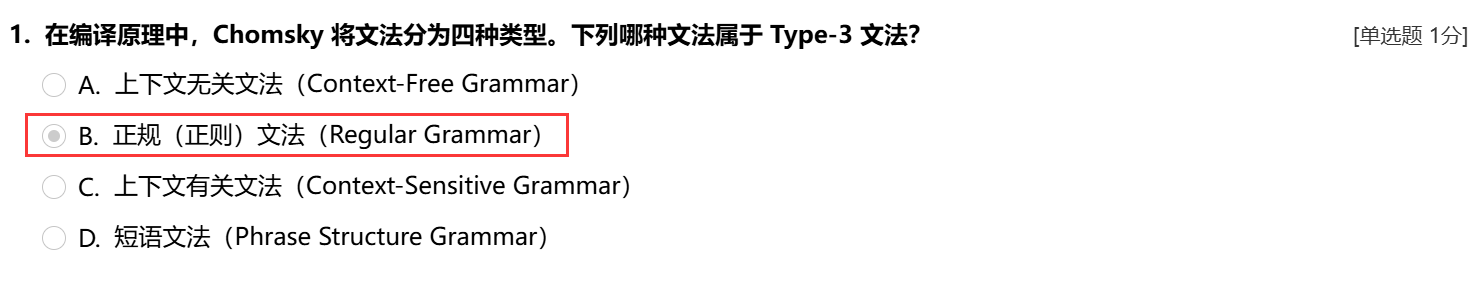
## 编译原理小雅——课堂练习/作业/测验

### 课堂练习

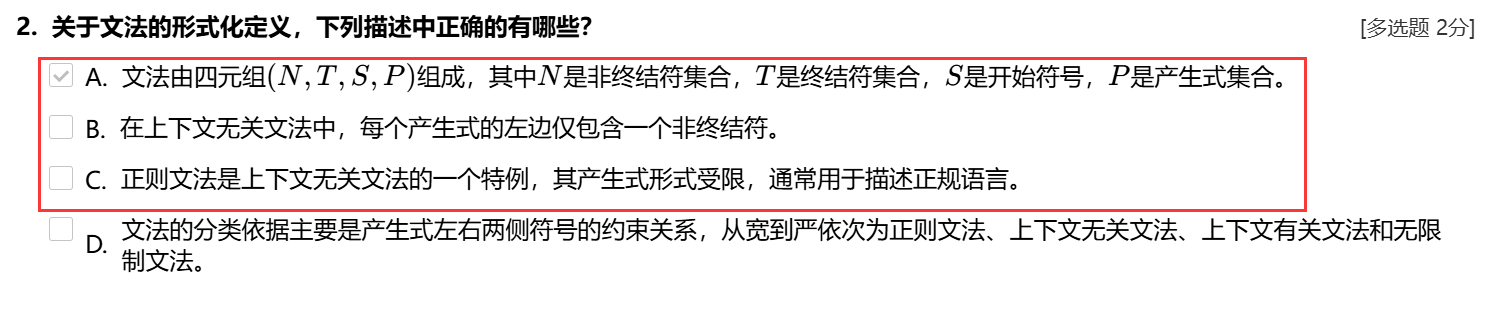
#### 第二章习题（文法与语言）



**解答：**乔姆斯基将文法分为四类，分别是0型，1型，2型，3型文法

（下表中高级文法的特点需首先满足低一级文法的特点）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 文法 | 别名 | 自动机 | 规则 | 解释 |
| 0型文法 | 无限制文法/  短语结构文法 | 图灵机 | a->b | 无限制 |
| 1型文法 | 上下文有关文法 | 线性有界非确定图灵机 | a1Aa2->a1Ba2  (B!=ε) | |A|<|B|除S->ε外  A只有出现在a1和a2的上下文中才能允许使用B替换 |
| 2型文法 | 上下文无关文法 | 非确定下推自动机 | A->γ | 左侧为一个非终结符 |
| 3型文法 | 正规文法 | 有穷自动机 | A->cB / A->c | 在2的基础上进一步  限定产生式的形式 |

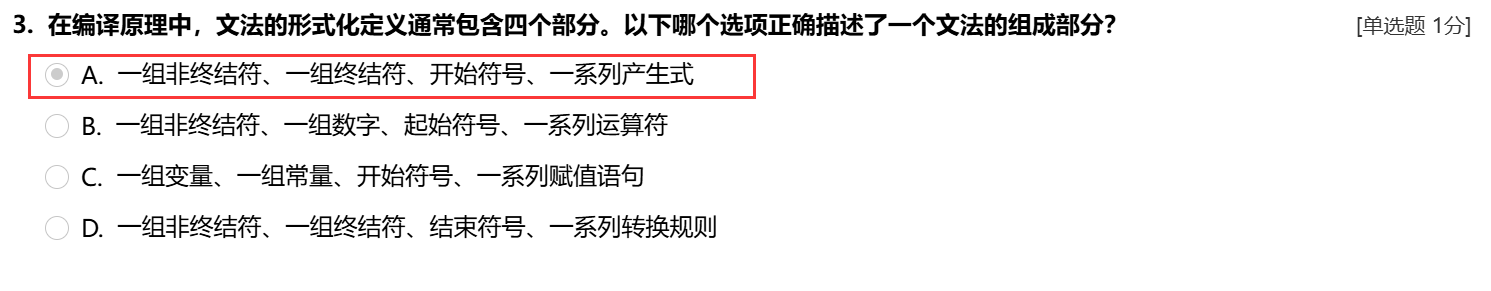


**解答：**

A.B.C均正确

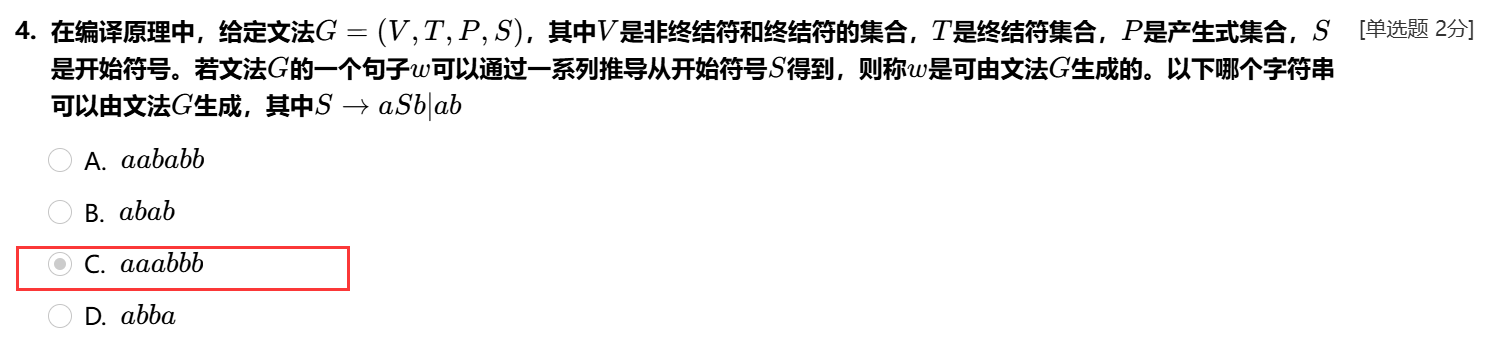
D.按照对产生式两侧符号的约束关系从宽到严进行排序：

无限制文法->上下文有关文法->上下文无关文法->正则文法



**解答：**

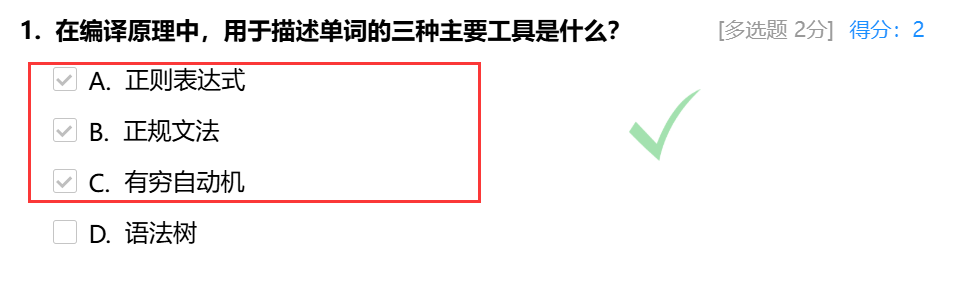
参看文法的形式化定义，由四元组 (N,T,S,P) 组成，可知A正确。



**解答：**

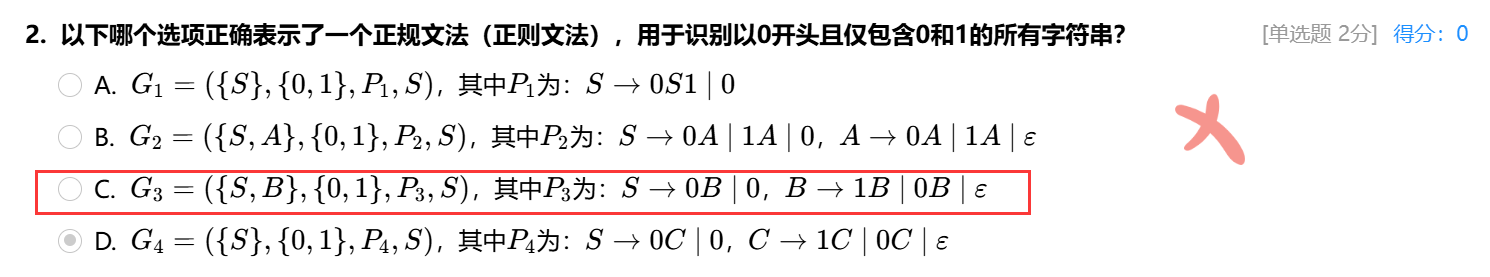
根据文法G的产生式S->aSb|ab可以推得，S->anbn，选项C满足题意

#### 课堂测练1



**解答：**

词法分析使用3型文法，即正规文法，其工具包括正规文法，正则表达式与有穷自动机



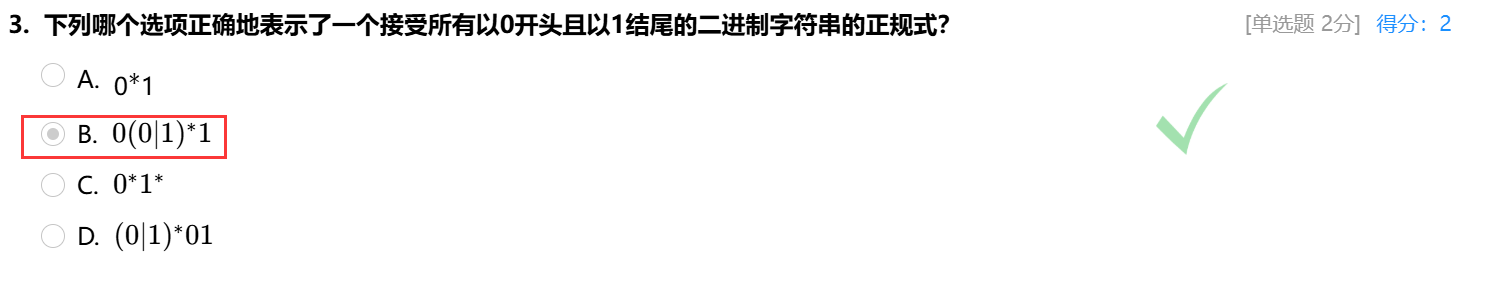
**解答：**

要识别以0开头且仅包含0和1的所有字符串，则第一个非终结符必须为0，排除B；

A选项描述的是0n1m（n>=1,n>=m>=0）的字符串，不满足题意；

D选项的产生式虽然满足条件，但在四元式的非终结符集里漏掉了非终结符C；

C选项满足条件。



**解答：**

A选项无法满足0开头，且只有一个1结尾；

C选项无法满足至少包含一个0开头，至少一个1结尾；

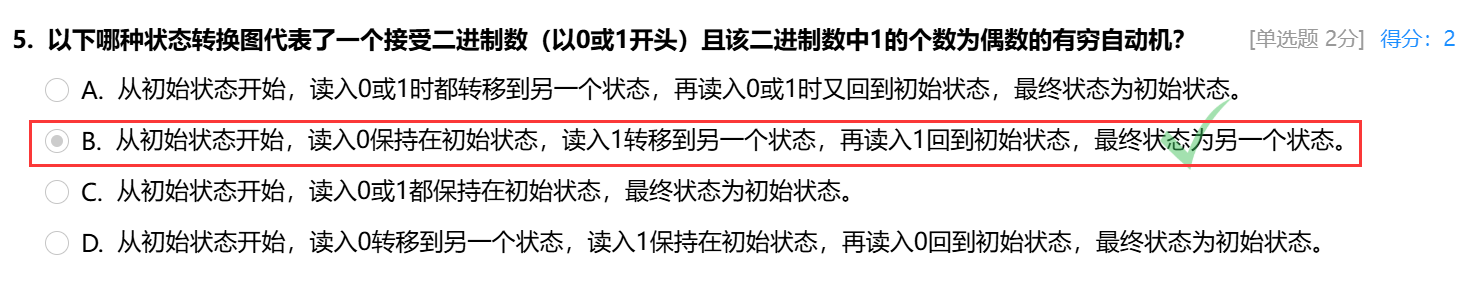
D选项无法满足1开头

B选项描述了一个以0开头1结尾，且中间为一个0或1的闭包，满足题目要求。



**解答：**

标识符通常由字母或下划线开头，后跟零个或多个字母、数字或下划线。因此，有穷自动机（DFA）用于识别标识符时，应从字母或下划线开始，然后连续读取字母、数字或下划线，直到遇到不属于这些类别的其他字符。这样可以确保正确地识别程序中的标识符。



解答：

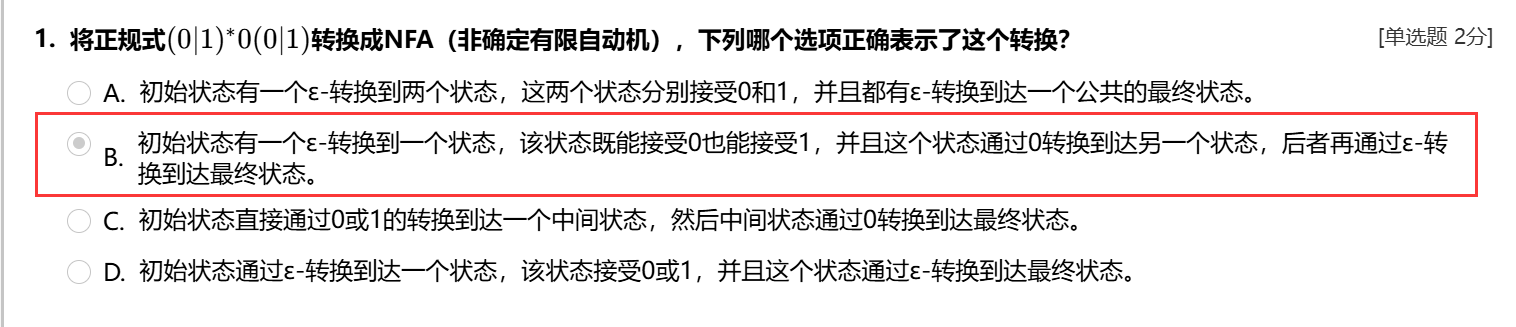
A选项无法保证1的个数为偶数：初态接受1进入X态，X态接受0进入终态；

B选项正确；

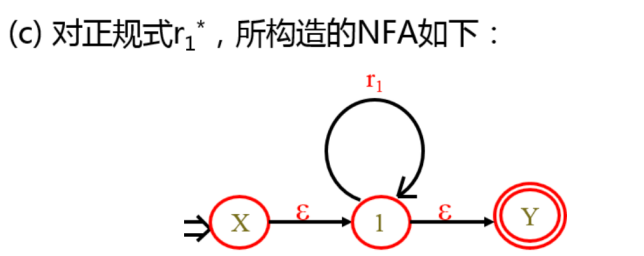
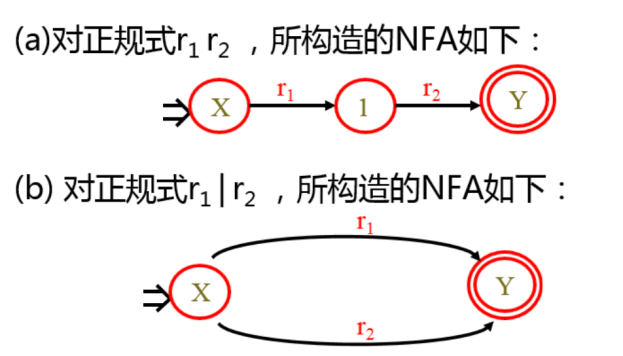
C选项同A；

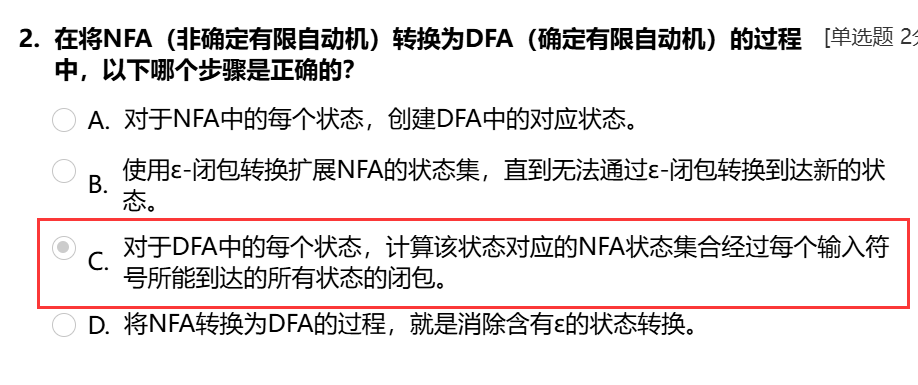
D选项只能保证0的个数为偶数。

#### 课堂测练2



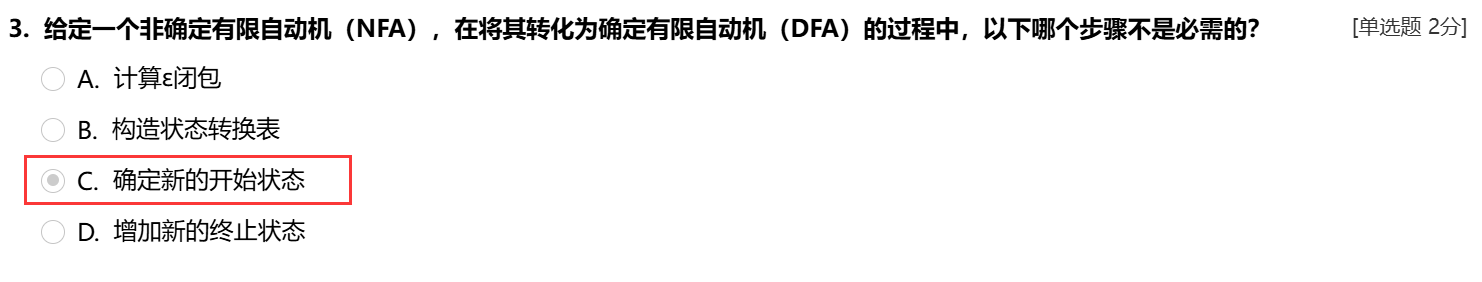
**解答：**具体转换过程参考下图规则



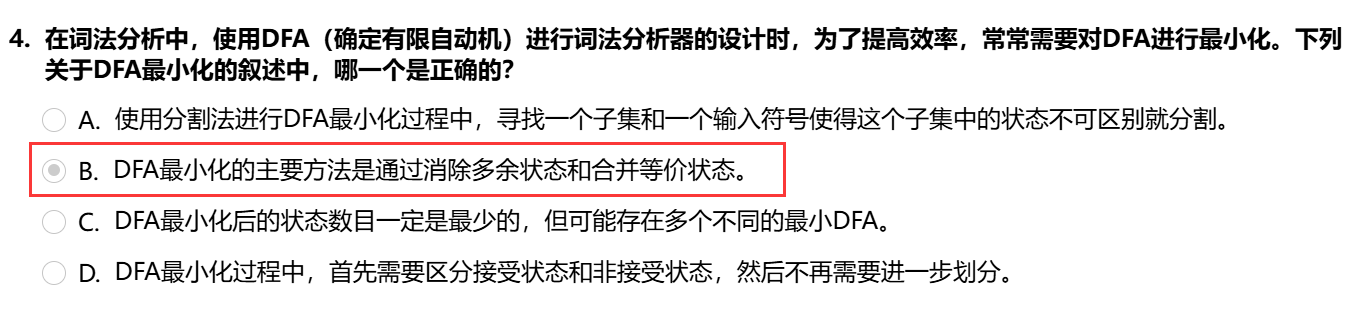


**解答：**

1. DFA的每个状态都对应NFA中的一个状态集，而非一一对应关系；
2. 需要先按照输入符号求可达状态，在求可达状态的ε-闭包；
3. 正确；
4. 消除含有ε的状态转换只是NFA确定化的一个步骤，还包括构建初始状态，确定状态转移等。

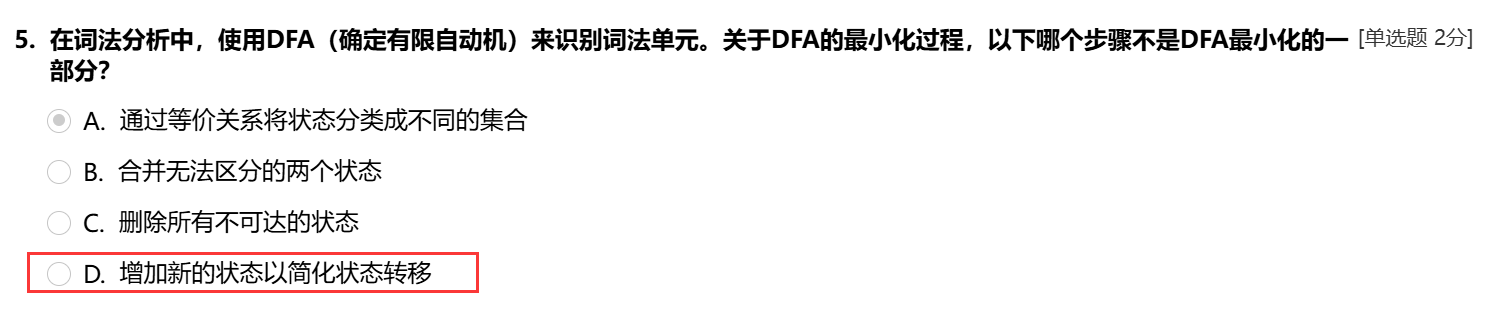


**解答：**



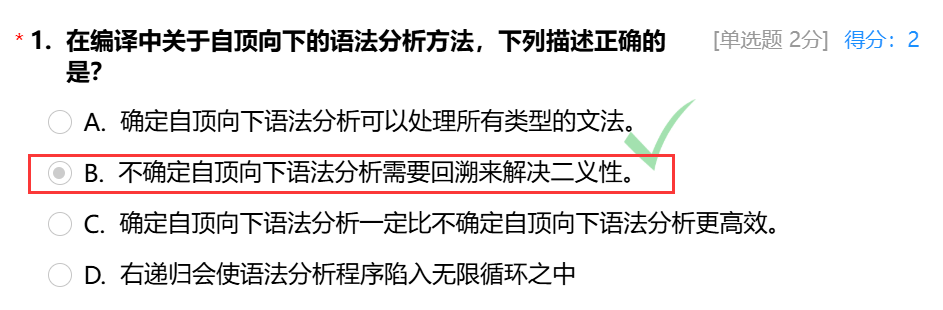
**解答：**

1. 在使用分割法进行DFA最小化时，当存在一个输入符号能够使得一个子集中的状态可以区分，就可以进行分割；
2. 正确；
3. 一个DFA最小化后的状态数目一定最少，且一定对应一个确定的最小DFA；
4. 在DFA最小化过程中，确实首先需要区分接收状态和非接受状态，但是还需要进一步进行划分。



**解答：**D与最小化DFA相悖

#### 课堂测练



**解答：**

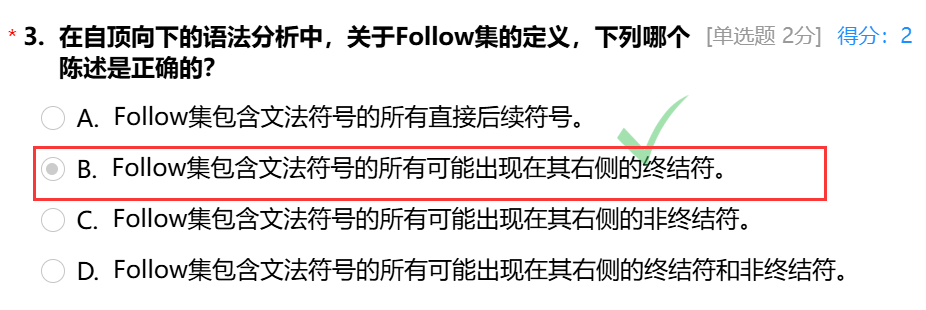
1. 自顶向下语法分析采用LL(1)分析，需要文法满足LL(1)文法，其需要满足：

一个上下文无关文法（2型文法）其各个产生式的可选集互不相交；

1. 正确；
2. 通常情况下确定的自顶向下语法分析的效率会更高，但是如果文法较简单两种方法的效率可能相同（不确定性失效），或者在进行确定自顶向下语法分析时，求解可选集的这一过程可能会造成计算量的增加从而导致效率变低；
3. 左递归会使语法分析程序陷入无限循环之中。



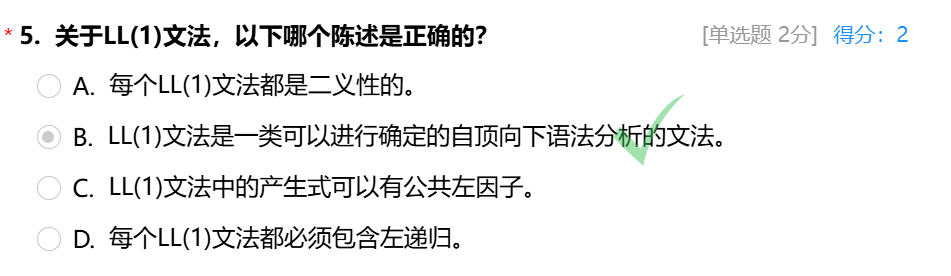
**解答：**根据定义，A项正确；



**解答：**根据定义，B项正确；



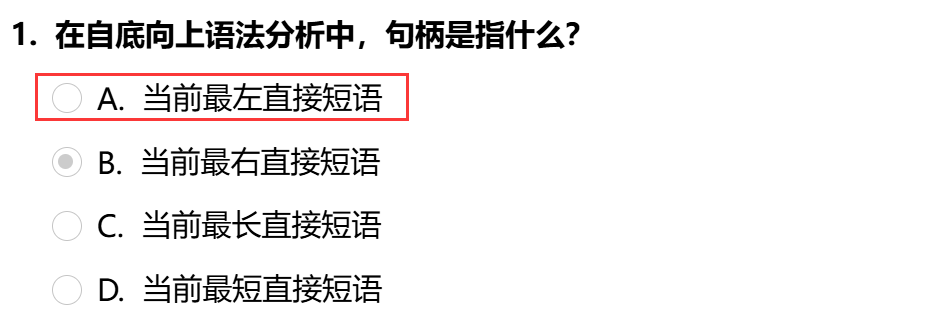
**解答：**根据定义，B项正确；



解答：

1. LL(1)文法是非二义的，因为 LL(1) 分析器在每一步都能唯一确定下一个要匹配的符号。
2. 正确
3. 如果一个文法有公共左因子，那么它通常不是 LL(1) 文法。公共左因子会导致在解析过程中出现歧义，无法通过 LL(1) 分析器唯一确定下一步的操作。
4. LL(1) 文法不允许直接或间接的左递归。左递归会导致自顶向下分析时无限循环，因此 LL(1) 文法必须消除左递归。

#### 课堂测练

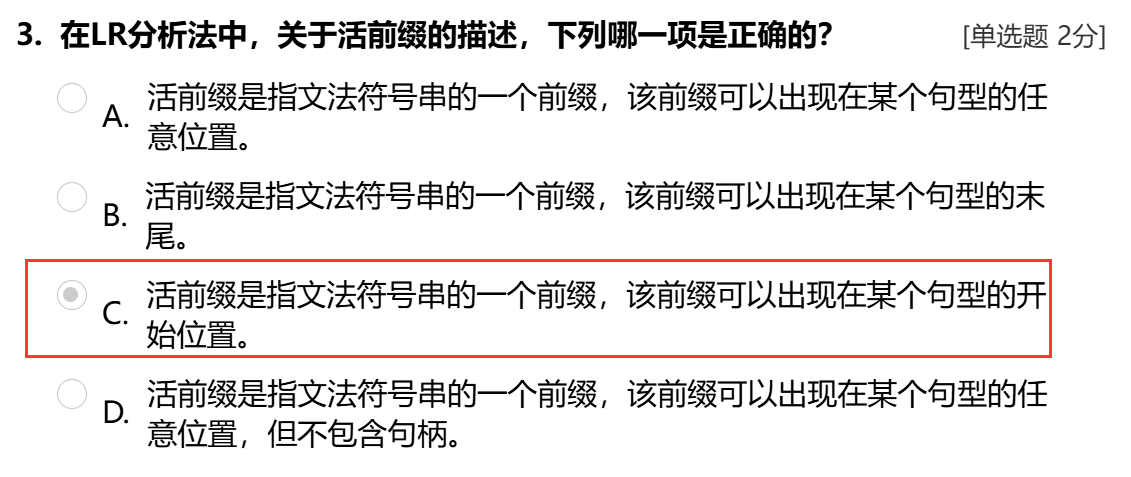


**解答：**由句柄的定义可知，A项正确

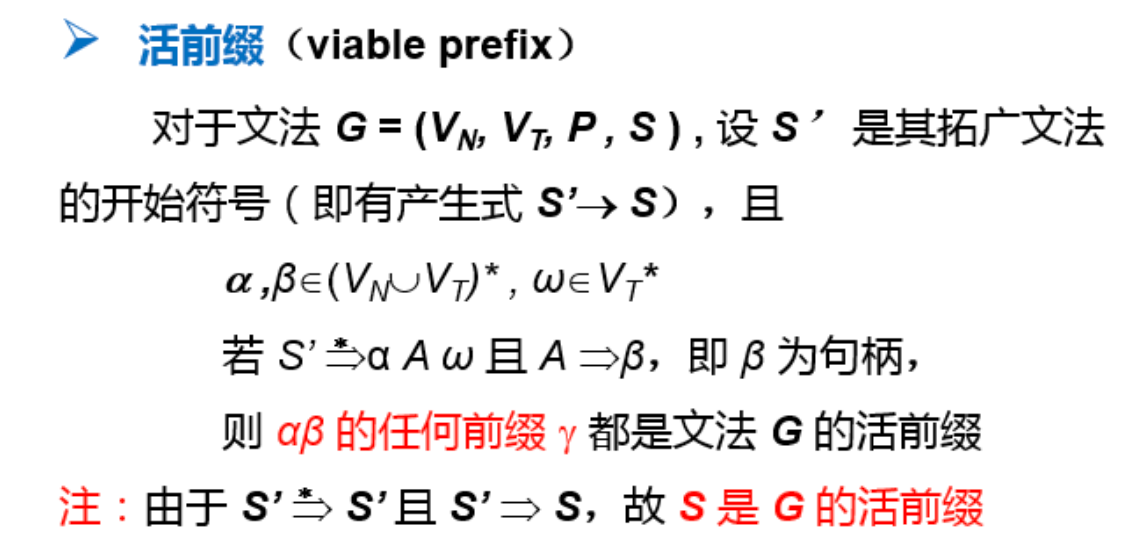


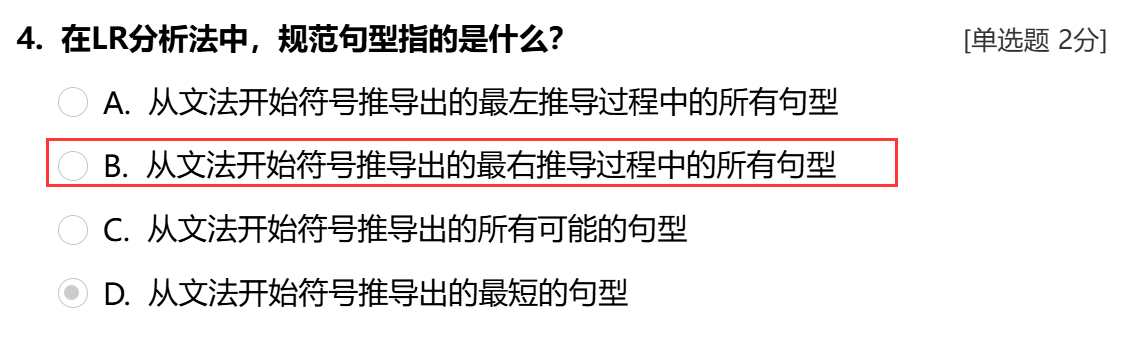
**解答：**最左素短语的定义（短语&至少包含一个终结符&不包含其他素短语）

素短语是规约的基本单位，最左素短语即句型的最左边开始的第一个可规约串

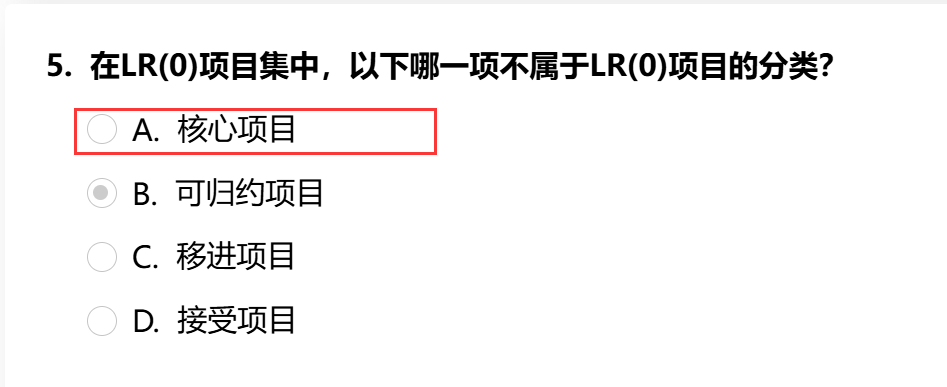


**解答**：参看下方对于活前缀的定义：

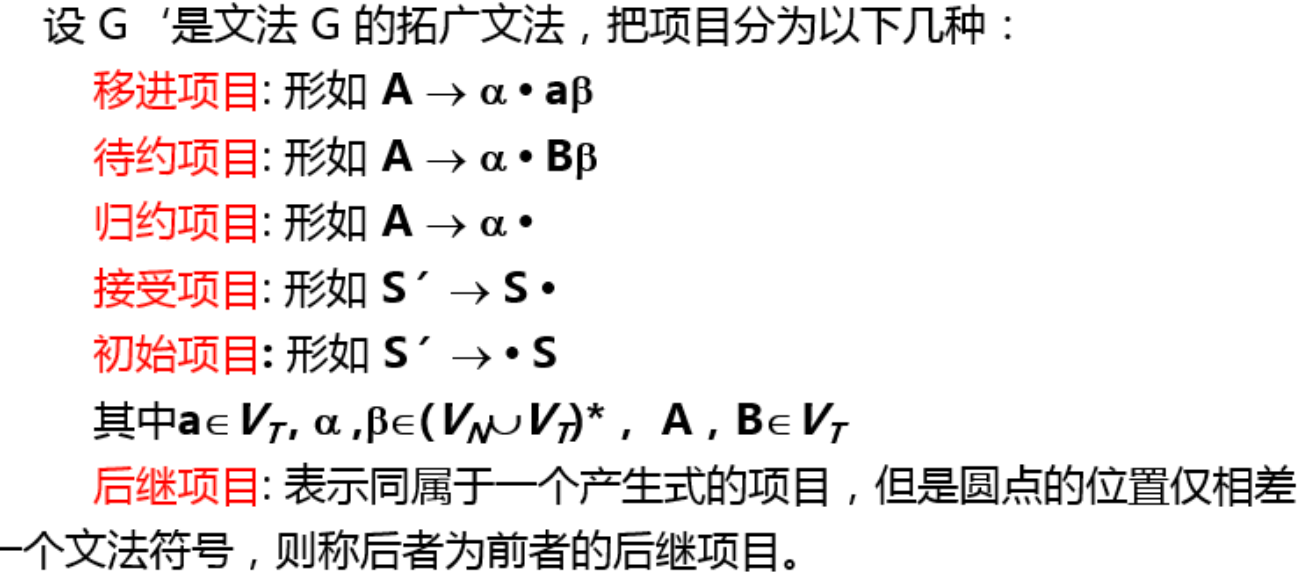




**解答：**LR分析法：L指从左到右扫描，R指最右推导，即规范推导，规范推导得到句型为规范句型。

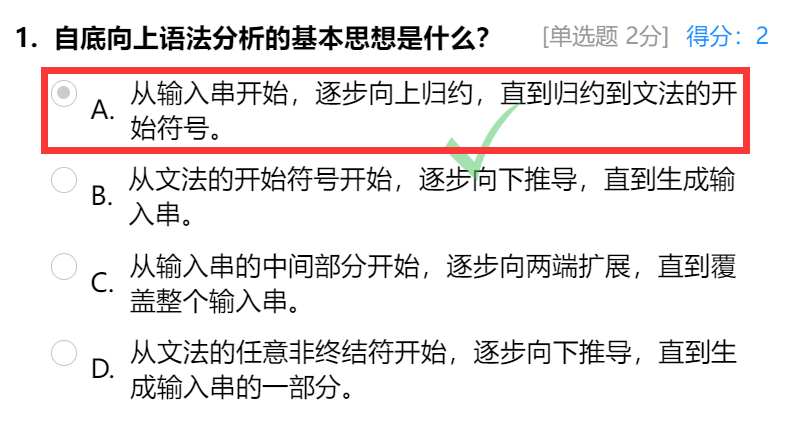


**解答：**参考下图，答案为A核心项目

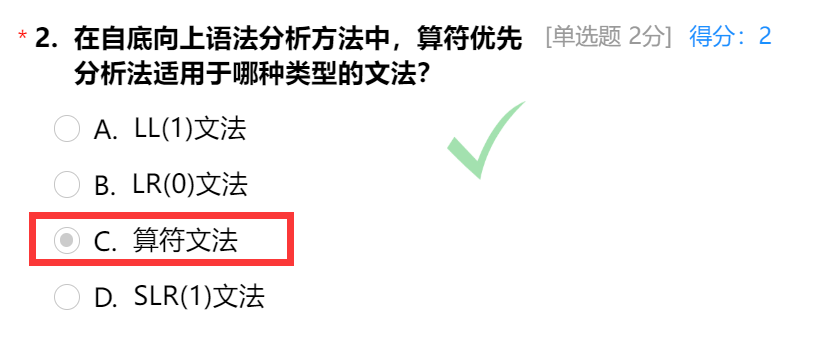


### 测验

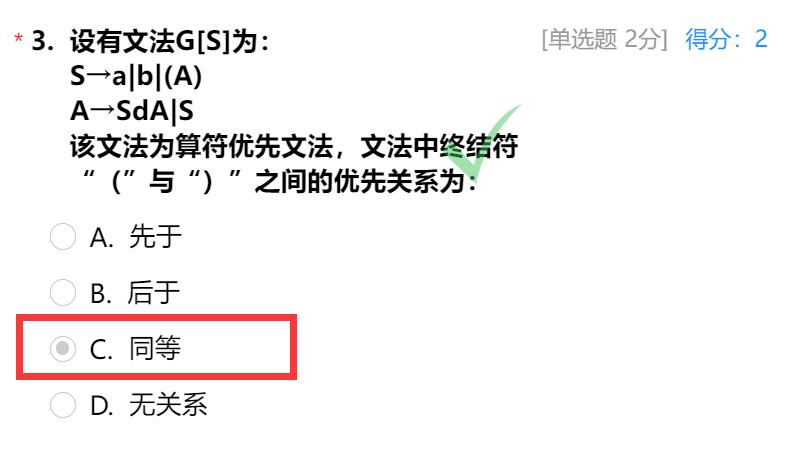
#### 线上自学单元测验（第五章）



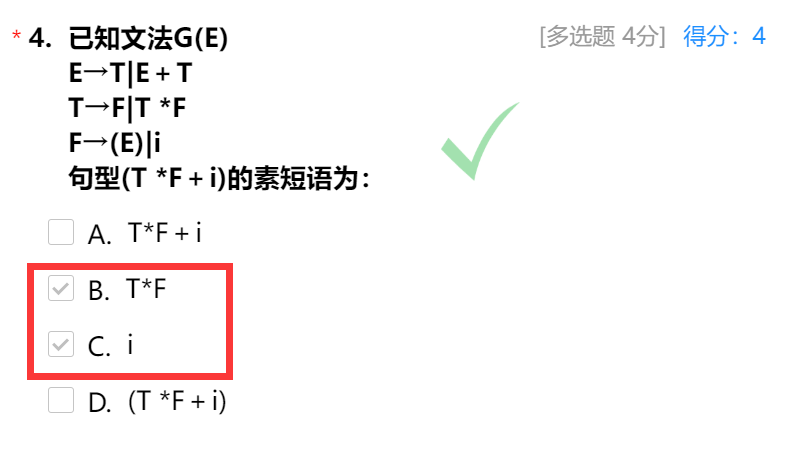
**解答**：参看自底向上语法分析的定义，可知A正确

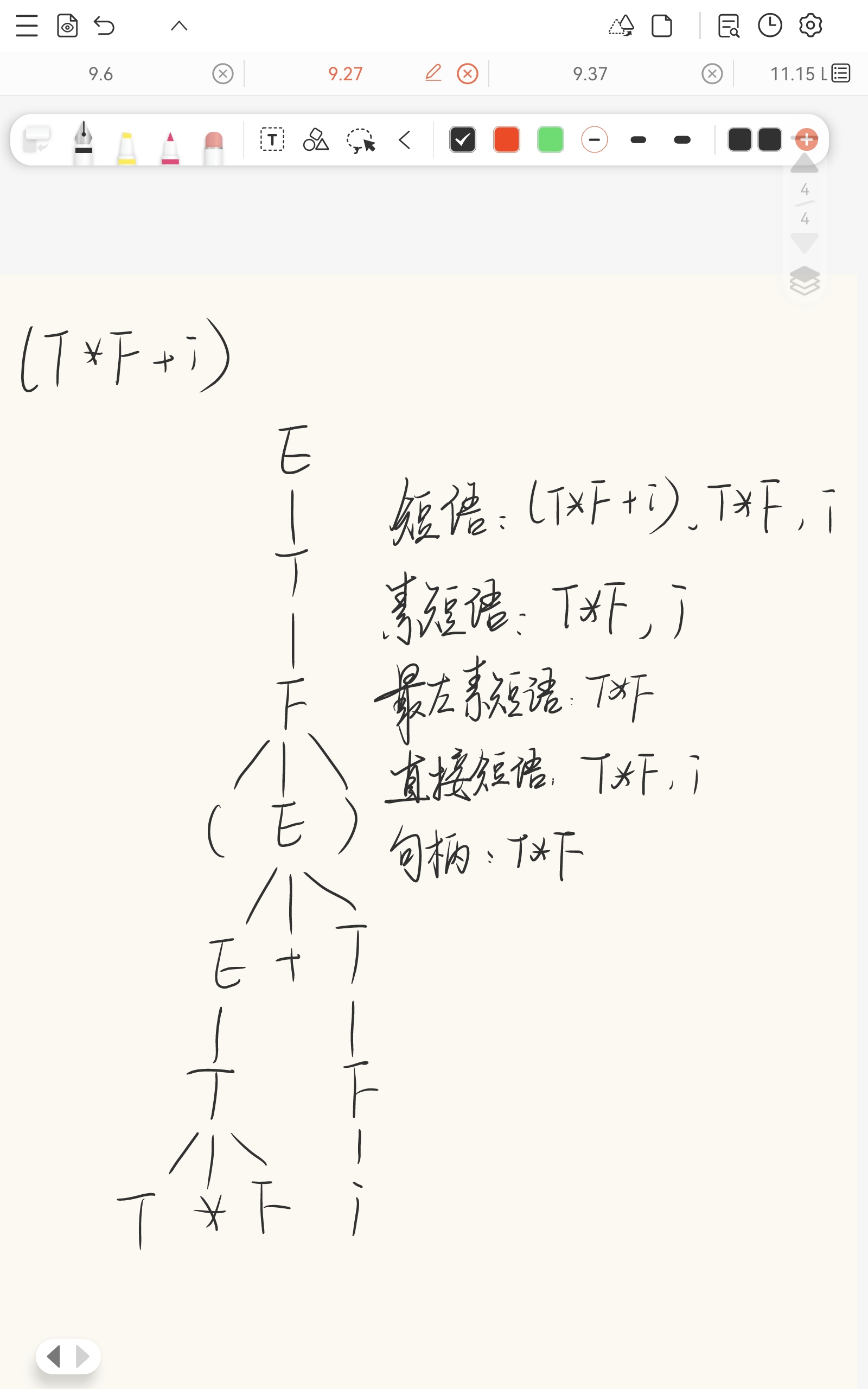


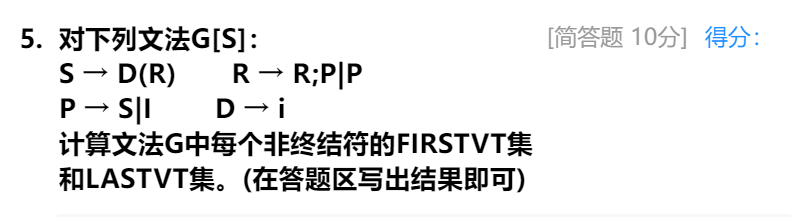
**解答：**算符优先分析法是一种自底向上的语法分析方法，特别适用于算符文法。算符文法是指不含ε产生式且每个产生式的右部不含两个相邻的非终结符的文法。因此，算符优先分析法适用于算符文法。



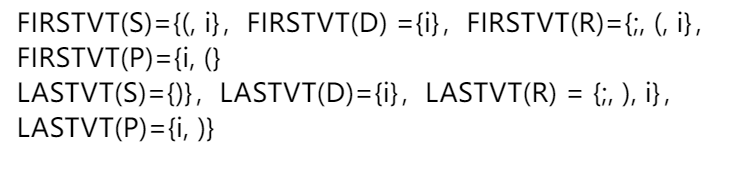
**解答：**（A），左括号与右括号之间只相差一个非终结符，因此两者的的优先关系同等。



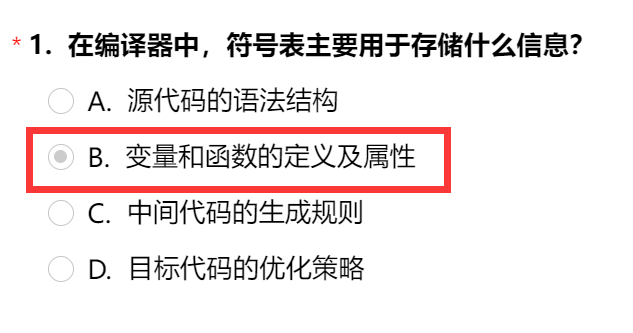


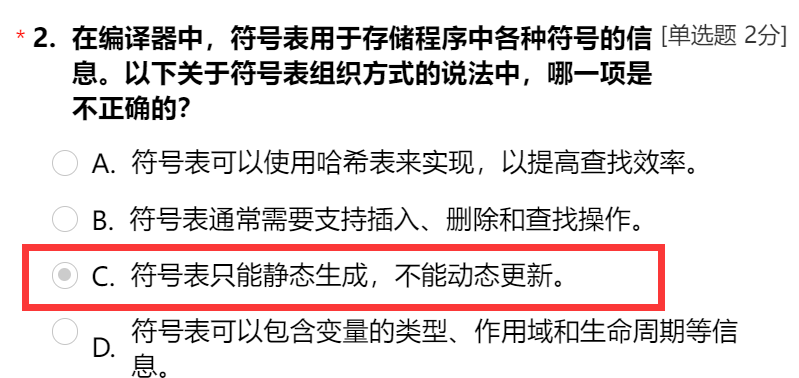


**解答：**



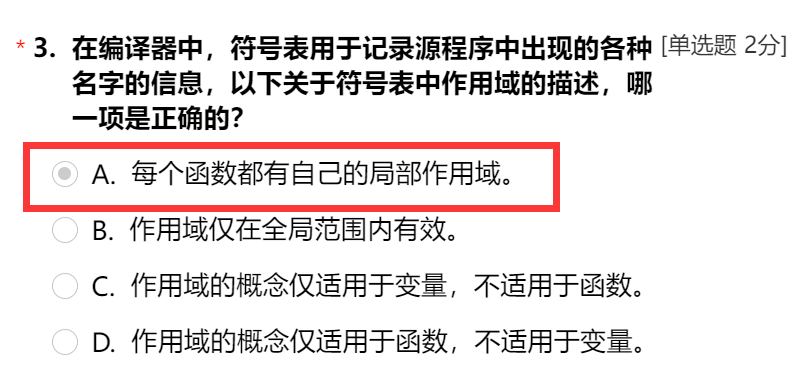
#### 单元测练 第八章



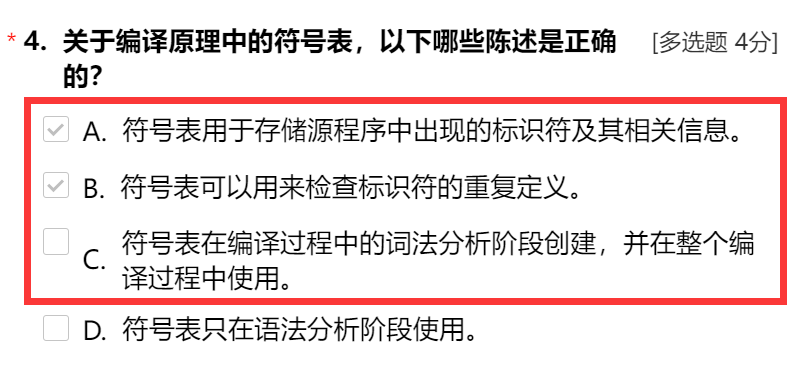


**解答：**

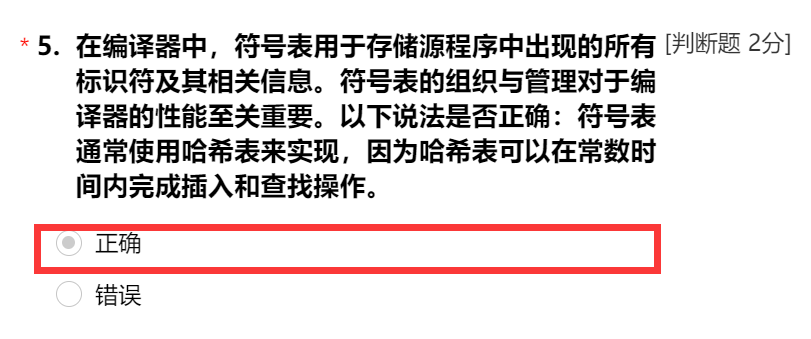
1. 符号表的组织形式：线性/排序/散列（哈希）
2. 符号表支持创建，释放，增删改查操作
3. 符号表支持增删改查，能够动态更新
4. 符号表可以用来存放变量和函数的定义，属性；体现作用域和可见性信息等。



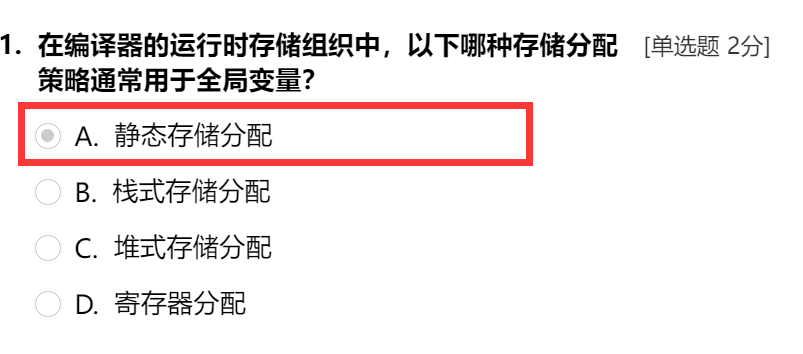
**解答：**作用域的概念适用于函数和变量；可分为局部作用域和全局作用域



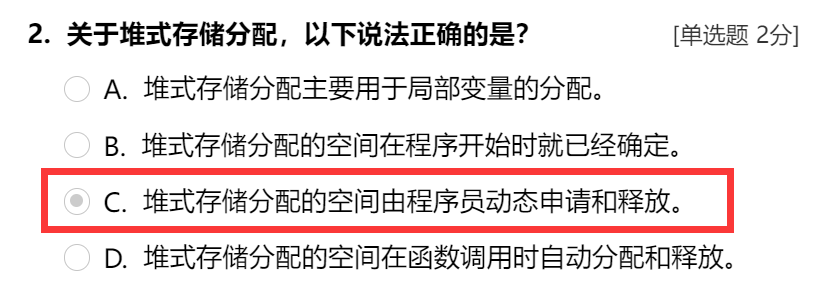
**解答：**符号表在整个编译过程中均可使用，故选ABC



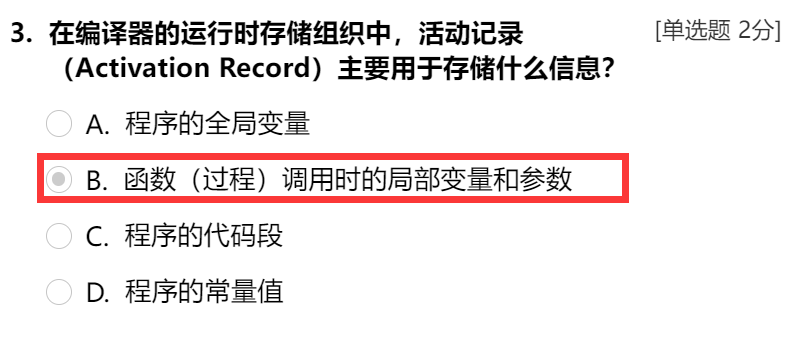
#### 单元测练 第九章



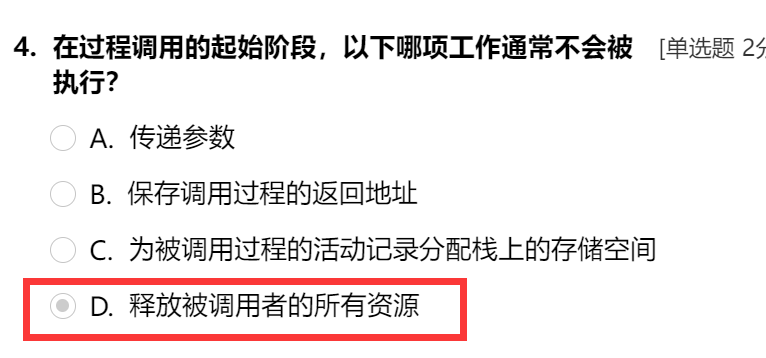
1. 静态存储通常存储全局变量或常量
2. 栈式存储通常存储函数调用时的参数、返回地址以及函数内部的局部变量
3. 堆式存储通常进行动态内存分配
4. 寄存器是CPU内部的高速存储单元，存储频繁使用的数据单元



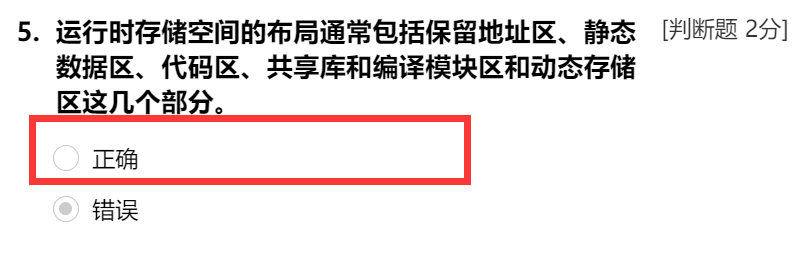
**解答：**堆式存储是由程序员动态申请与释放的



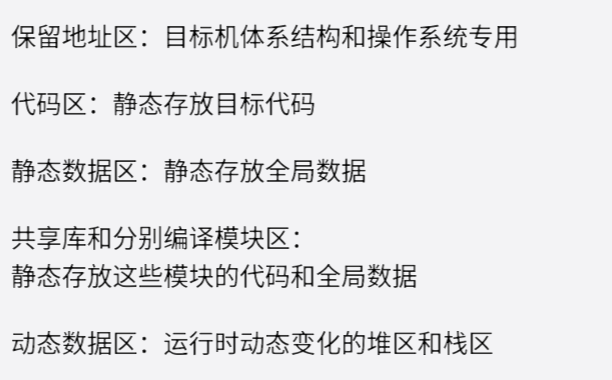
**解答：**活动记录用于储存局部变量，函数实参，临时值，等数据信息以及必要的控制信息

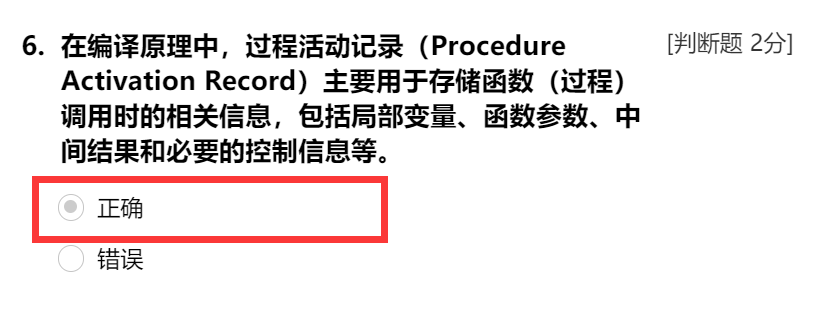


**解答：**释放被调用者资源往往是在过程调用结束的阶段

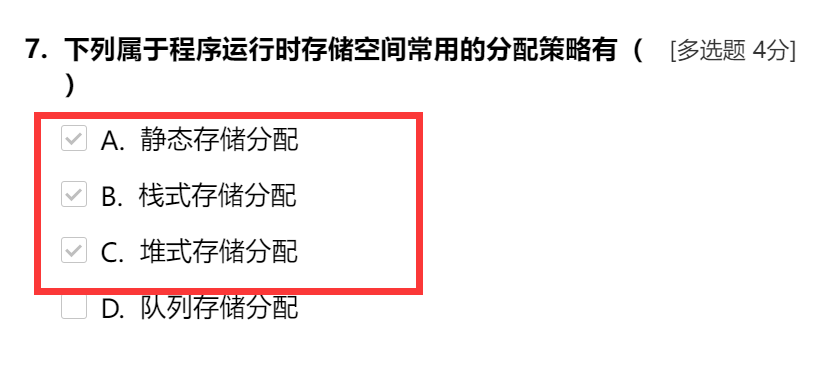


**解答：**

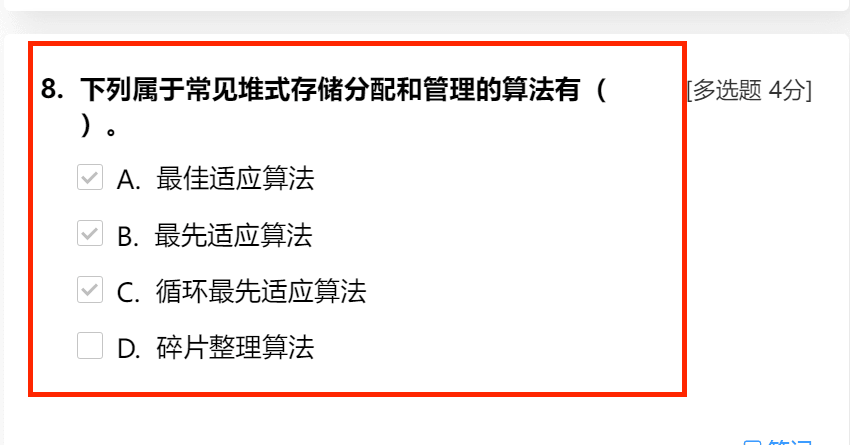


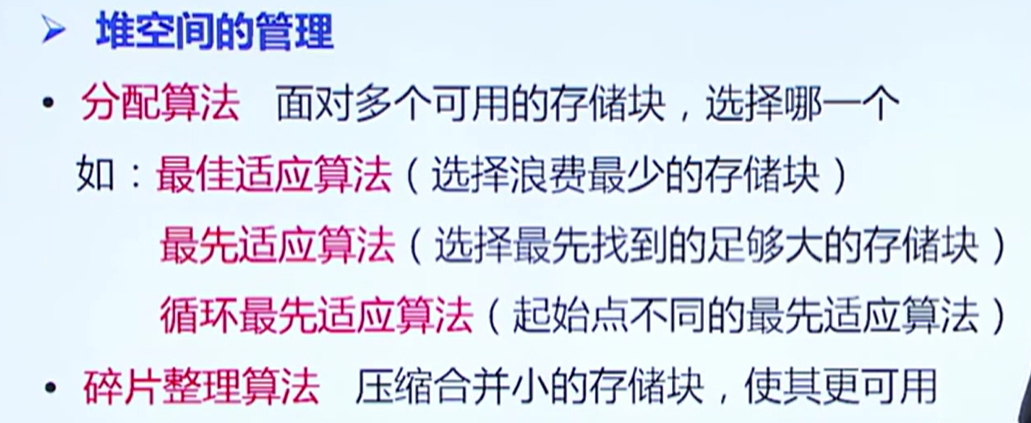


**解答：**原文，表述的非常正确

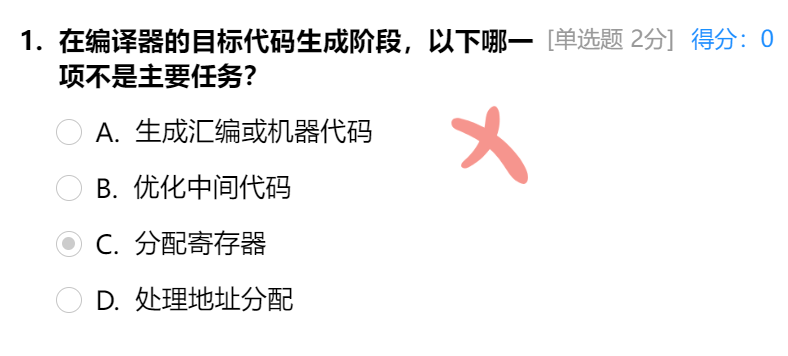


**解答：**运行时存储空间常用的分配策略：静态/动态（栈/堆）

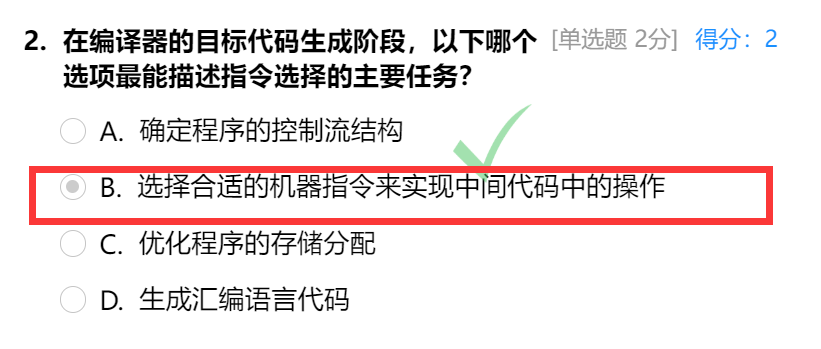




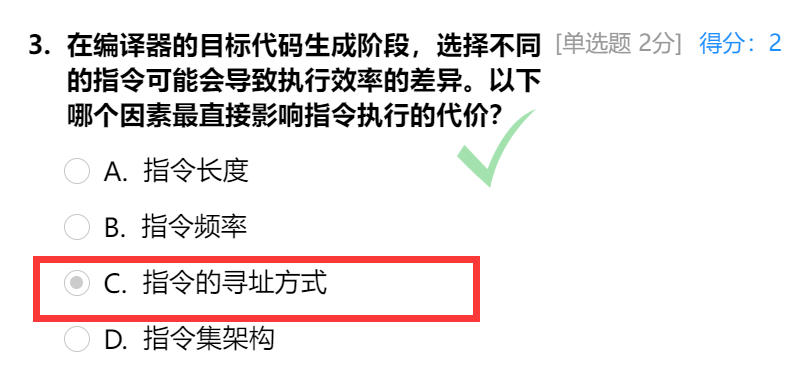
#### 单元测练 第十一章



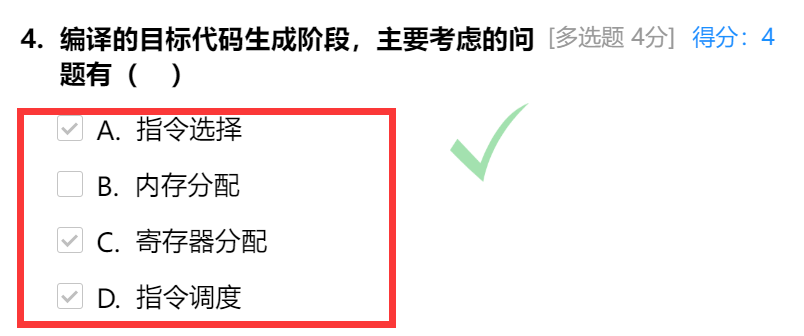
**解答：**优化完中间代码后才到目标代码生成阶段，因此不是该阶段的主要任务



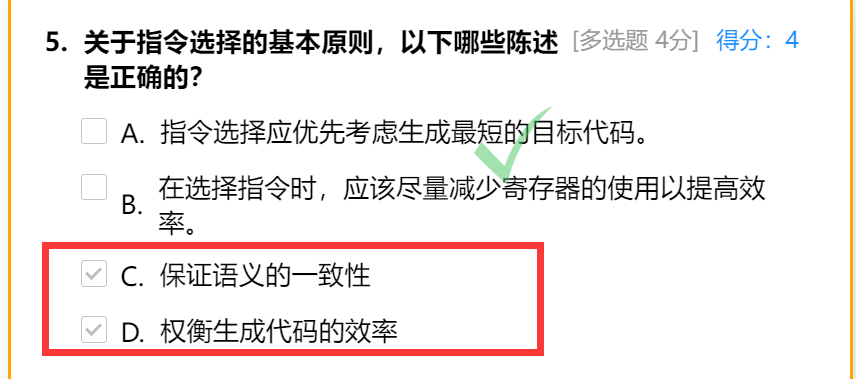
**解答：**指令选择是目标代码生成阶段的一个关键步骤，其主要任务是从中间代码生成对应的机器指令。



**解答：**在目标代码生成过程中，指令的寻址方式会直接影响指令执行所需要的CPU周期。相比之下，指令长度、指令频率和指令集架构虽然也会影响执行效率，但不如指令寻址复杂度直接。



**解答：**目标代码生成阶段的三个方面：指令选择，寄存分配和指令调度



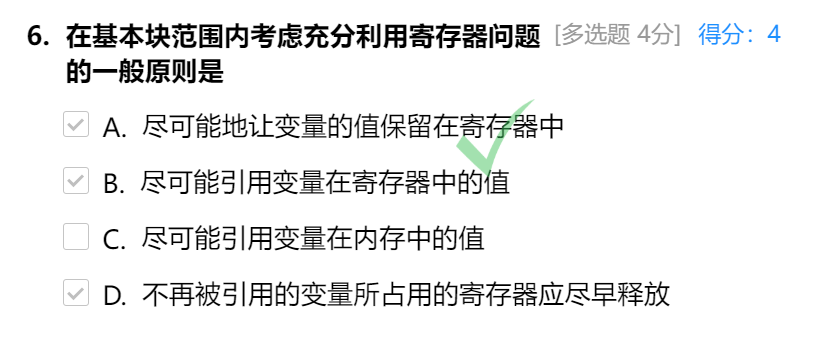
**解答：**

A选项不完全正确，虽然代码长度是一个考虑因素，但并不是唯一的原则。

B选项错误，因为合理使用寄存器可以提高代码效率。

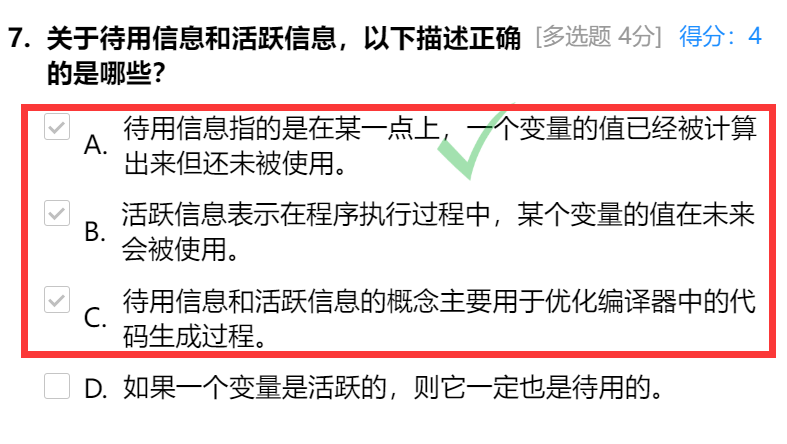
C选项正确，指令选择有保证与中间代码的语义一致。

D选项正确，优化编译器的一个重要目标就是在保证正确性的前提下，使生成的代码执行速度最快，但是这个与语句的上下文和目标机的体系结构有关，需要权衡。

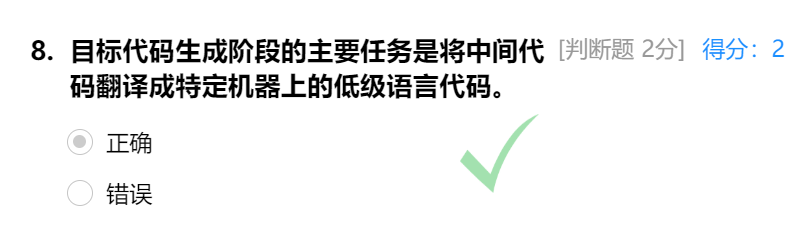


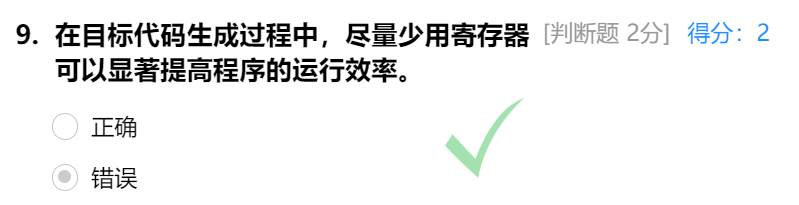
解答：在基本块内充分利用寄存器的一般原则：

1. 尽可能地让变量地值保留在寄存器中
2. 尽可能引用变量在寄存器中的值
3. 不在被引用的变量所占用的寄存器应尽早释放

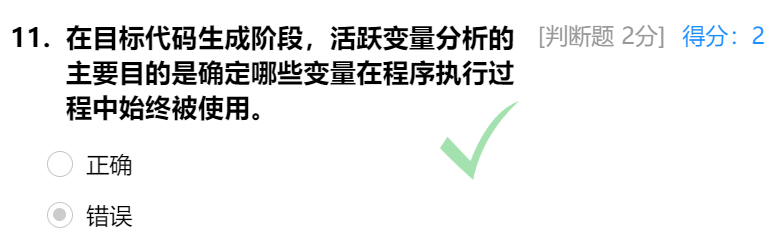
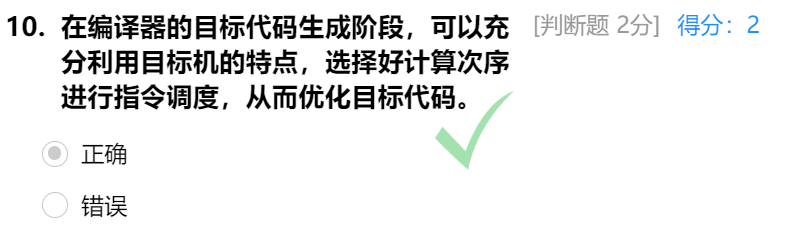


**解答：**根据A,B两个选项，一个活跃信息在未来会被使用，但无法确定其是否已经生成，而待用信息指的是已经被计算出来的值，两者有冲突，因此D错误





**解答：**合理的分配寄存器才能提高程序地运行效率



**解答：**活跃变量是将在未来被使用，而不是始终会被使用