例题 2.1 是非题

1．因名字都是用标识符表示的，故名字与标识符没有区别。（ ）

2．正规文法产生的语言都可以用上下文无关文法来描述。（ ）

3．上下文无关文法比正规文法具有更强的描述能力。（ ）

分析

1．标识符是高级语言中定义的字符串，一般是以英文字母开头（包括大小写字母）的，由数字、字母和一些特殊字符（如下划线等）组成的一定长度（不同的高级语言对标识符的长度的规定不同）的字符串，它只是一个标志，没有其它含义。名字是用标识符表示的，但名字不仅仅是一个字符串，它还具有属性和值，就象一个人的名字不仅仅是一个符号，对于认识这个人的人来说，它还代表着这个人，因此本题错。

V_T^*

2．乔姆斯基把文法分成 4 种类型，从 4 种文法的形式化描述来看，它们的差别主要在对规则形式的约束上。0 型文法的规则形式是：每个产生式$\alpha\rightarrow\omega$都满足，α（VN∪VT）*且至少含有一个非终结符，ω （VN∪VT）*，0 型文法也叫短语文法。1 型文法的规则形式是：每个产生式$\alpha\rightarrow\omega$均满足$|\alpha|\leq|\omega|$，仅仅 S→例外，且 S 不得出现在任何产生式的右部，1 型文法也称上下文有关文法，从两种文法的规则形式来看，1 型文法对规则形式的约束比 0 型文法强，1 型文法能描述的语言 0 型文法也能描述，而 0 型文法能描述的语言 1 型文法不一定能够描述，1 型文法是 0 型文法的特例。2 型文法的规则形式是 A→ω，其中 A∈VN，ω （VN∪VT）*，2 型文法也称上下文无关文法，分析 2 型文法的规则形式不难发现，2 型文法是 1 型文法的特例，也是 0 型文法的特例。3 型文法的规则形式是 A→αB 或 A→α，其中α ，A、B∈VN，3 型文法也称正规文法，可以看出 3 型文法是前面 3 种文法的特例。从上面的分析可以，正规文法是上下文无关文法的特例，可以用正规文法描述的语言，其正规文法描述的形式也是上下文无关文法的描述形式，即可以用上下文无关文法描述，因此本题对。

3．上下文无关文法是 2 型文法，正则文法是 3 型文法，从上题的分析可以看出，3 型文法是 2 型文法的特例，3 型文法可以描述的语言都可以用 2 型文法来描述，而 2 型文法可以描述的语言则不一定能用 3 型文法来描述。即 2 型文法比 3 型文法的描述能力强，因此本题对。

例题 2.2 填空题

1．程序语言是由（ ）和（ ）两方面定义的。

2．常用了参数传递方式有（ ）、（ ）和（ ）。

3．文法 G 所产生的句子的全体的（ ），将它记为（ ）

4．一个上下文无关文法包含四个组成部分是（　　）。

解答

1．如同自然语言一样，程序语言主要由（语法）和（语义）两个方面定义，有时，语言定义也包含语用信息，语用主要是有关程序设计技术和语言成分的使用方法。

2．参数传递主要有三种不同的途径，（传地址（call by reference））、（传值（call by value））和（传名（call by name）），传名也常称为换名。

3．假定 G 是一个文法，S 是它的开始符号。如果 $S \overset{*}{\Rightarrow} \omega$，则称ω是一个句型。仅含终结符号的句型是一个句子。文法 G 所产生的句子的全体是一个（语言），将它记为（L（G））。

4．一个上下文无关文法 G 包括四个组成部分：（一组终结符号），（一组非终结符号），（一个开始符号），（一组产生式）。

说明: 高级语言参数传递的方式通常有 4 种，传地址、得结果、传值和传名。早期的 FORTRAN 语言通常采用传地址或得结果的方式来实现参数的传递，这两种方法的不同给 FORTRAN 的标准化增加了难度，相比来说，传地址是一种常用的参数传递方式，而得结果则不是。

例题 2.3

```
program  test (input, output)
    var  i, j: integer;
    procedure  CAL(x, y: integer);
     begin
       y:=y**2;  x:=x-y;  y:=y-x
     end;  {CAL}
    begin  {main}
     i:=2;  j:=3;  CAL(i, j)
     writeln(j:1)
```

end. {main}

假定程序在语法上是正确的，采用哪一种参数传递方式使程序打印 16。

（A）call by reference （B）call by name （C）call by value

（D）前述三种中的二种 （E）都不是（**为乘幂运算）

（上海交通大学 1995 年研究生入学考试试题）

解答

这是一道选择题，要知道正确结果，必须先对三种参数传递方式计算程序的运行结果。

1．传地址：传地址时在子程序中对形式参数的引用实际上是对实参地址的引用，也就是说，对形式参数的值的修改实际上就是对实参值的修改。程序中在调用 CAL 函数之前，首先对 i 和 j 赋值，即 i=2，j=3，函数调用后，函数 CAL 先对 y 进行乘幂运算，也就是执行 j:=j**2，执行后 j 的值为 9，然后执行 x:=x-y，即 i:=i-j，执行后 i 的值为-7，最后执行 y:=y-x，即 j:=j-i，则最后 j 的值为 16。所以采用传地址的参数传递方式时程序打印 16。

2．传名：即名字替换，过程调用的作用相当于把被调用段的过程体抄到调用出现的地方，但把其中任一出现的形式参数都替换成相应的实在参数，那么程序的执行过程实际上是：

```
i:=2; j:=3;

    i:=i**2; i:=i-j; j:=j-i
```

其中，第一行是主程序中的语句，第二行是用子程序 CAL 中的语句替换函数调用 CAL（i，j），然后用实参替换形式参数的结果（即用 i 替换 x，用 j 替换 y）。执行上面的程序语句，我们可以发现最后 j 的值为 16，因此采用传名的参数传递方式程序的打印结果仍然是 16。

3．传值：调用段把实在参数的值计算出来并存放在一个被调用段可以拿得到的地方。被调用段开始工作时，首先把这些值抄入形式单元中，然后就好像使用局部名一样使用这些形式单元。如果实在参数不为指示器，那么，在被调用段中无法改变实参的值。这里，实参都不是指针，采用传值的方式传递参数时，函数的执行不会改变实参的值，程序在函数调用前给 i 赋值 2，给 j 赋值 3，在函数执行完后，i 和 j 的值将仍然是 2 和 3，因此传值时打印值为 3。

通过上面的分析，我们可以看到传地址和传名的打印结果都是 16，正确答案是（D）。

说明: 针对不同的参数传递方式对程序的结果进行计值是一种类型的题，这几年的考题基本上都有一道该类型的题，但在今天，通用高级语言中应用最多的参数传递方式是传值和

传地址，在语言学习中，其它两种参数传递方式已很少有人讨论，但出于应考的目的，掌握它们还是很有必要的。

本题只讨论了三种参数传递方式，如果是得结果呢？每个形式参数都对应两个单元，第 1 个单元存放实参的地址，第 2 个单元存放实参的值。在过程体中对形参的任何引用或赋值都是对它的第 2 个单元的直接访问。但在过程工作完毕返回前必须把第 2 个单元的内容存放到第 1 个单元所指的哪个实参单元之中。本题中的形参 x 和 y 就都有两个单元，它们的第一个单元分别存放 i 和 j 的地址，第二个单元分别存放 i 和 j 的值，进入子程序 CAL 之后，访问的都是 x 和 y 的第二个单元，CAL 执行完后，x 和 y 的第二个单元的值分别变为-7 和 16，然后在返回前将第 2 个单元的内容存放到第 1 个单元所指的哪个实参单元之中，即将-7 抄给 i，将 16 抄给 j，所以本题如果选择得结果的参数传递方式，程序的打印结果仍然是 16。