**介绍**  
在计算机科学中，序列和数组的项用于枚举组合结构，这类结构在多个领域有着广泛的应用。例如，在分析算法的复杂度时，我们需要了解（或估算）这种整数值函数。常见的例子包括关于 n 的多项式、指数函数（基数为整数，指数为关于 n 的多项式）、阶乘和二项式系数。而较为不熟悉的则是斯特林数（Stirling numbers），它们在本书的其他部分中已有介绍。这里将讨论的正整数序列被称为“括号、满二叉树、有效的变量乘积、栈排列”和“凸多边形的三角剖分”。

在通过公式明确定义Catalan数后，我们将通过组合论的方式展示它们如何计数平面中的非负路径。随后，我们将通过建立与先前展示过的结构集之间的一一对应关系，证明所有后续考虑的结构集的大小都为Catalan数。

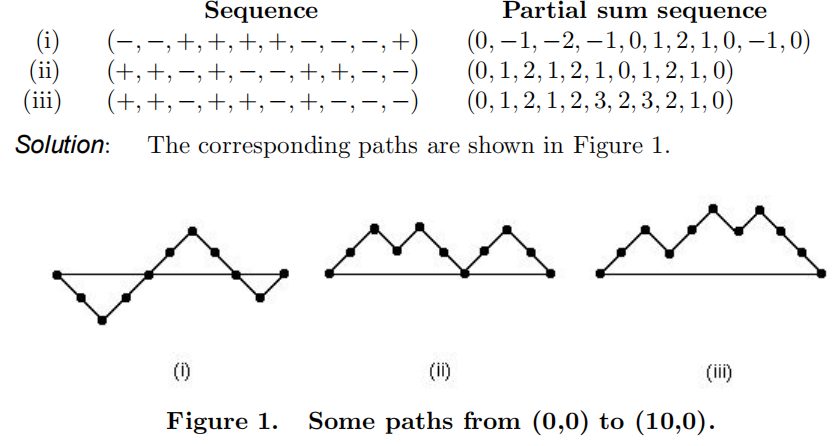
通过递推关系和生成函数，我们将采用另一种方法证明某些结构集（例如有效的变量乘积）的大小是Catalan数。由于已经建立的一一对应关系，所有被考虑的结构集最终都将通过Catalan数来枚举。接着，我们将研究序列的渐近行为，并获得第 n 个Catalan数的数量级。

**路径与Catalan数**  
假设在一次选举中，共投出了 m=a+b 张选票，其中候选人A得到了 a 票，候选人B得到了 b 票。选票按某种随机顺序逐一计数，产生一个由 a 个A和 b 个B组成的序列。可能的选票序列的数量是 C(m,a)，即从集合 {1,2,…,m}中选取 a 个元素的子集数，因为每个这样的子集表示哪 m 张选票是投给了A的。如果假设所有的序列出现的概率相同，那么问题转化为求出A在整个计票过程中一直领先的概率。

每一个选票序列可以表示为一个有序的 m-元组 (x1,x2,…,xm)，其中 xi=1表示第 i 张选票投给了A，xi=−1表示投给了B。每当计数到第 i张选票时，第 i 次部分和 si=x1+x2+⋯+xi（其中 s0=0）表示A相对于B的“领先”情况。如果我们用 P(a,b)来表示所有部分和 si>0 的序列数，那么A在整个计票过程中始终领先的概率就是 P(a,b)/C(m,a)。

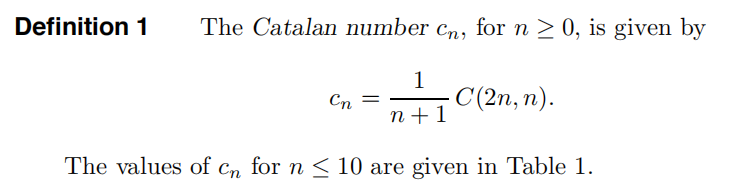
Catalan数出现在当 a=b 的情形，假设 a=b=n，那么 m=2n。我们现在考虑求出A在整个计票过程中始终不落后（即从未被B领先）的概率。有 C(2n,n) 种可能的序列，它们包含 n 个 1 和 n 个 -1。我们要求的是那些部分和 si≥0 的序列，即对于所有 i=1,2,…,2n−1，都有 si≥0。

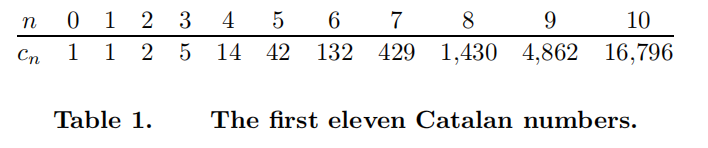
我们可以通过路径来表示序列 (x1,x2,…,x2n)，这个路径从原点 (0,0)到点(2n, 0)，路径的每一步是从 (i−1,si−1) 到 (i,si)的线段，其中 i=1,2,…,2n，路径的第 i 步的斜率为 si−si−1=xi∈{1,−1}。

**例子 1**  
假设 n=5，则路径从原点到点 (10,0)。我们展示了三种这样的序列及其部分和序列。为了清晰起见，序列中的 1用加号表示，-1 用减号表示。接下来我们绘制这些对应的路径。  
注意在序列 (i) 中，部分和序列既有正数也有负数，所以路径会在 x-轴的上方和下方都有分布。而在序列 (ii) 中，所有部分和都是非负的，所以路径始终不低于 x-轴。但在序列 (iii) 中，对于所有 i=1,2,…,2n−1，我们都有 si>0。因此，除了两个端点之外，路径完全位于 x-轴的上方。

我们称，如果所有 si≥0，从 (0,0)到 (2n, 0) 的路径为**非负路径**；如果对于所有 i=1,2,…,2n−1，都有 si>0，称为**正路径**。因此，非负路径对应于一个选举序列，其中候选人A和候选人B各获得了 n 票，但A从未落后。而正路径则表示A在整个过程中始终领先，直到最后一票使得两者平局。

我们将看到，从原点到点 (2n,0)的非负路径和正路径的数量都是Catalan数。在进一步讨论这个选举问题之前，我们先定义Catalan数。





**例子 2**  
从原点到点 (6,0)的所有非负路径。哪些是正路径？

**解答**：  
有 C(4,2)=6种长度为6的斜率序列，从1开始，以-1结束。在这些序列中，有五个是非负路径 — 序列 (+,−,−,+,+,−)不是。为了检查是否是非负路径，我们计算它们的部分和，并发现非负路径的斜率序列如下：

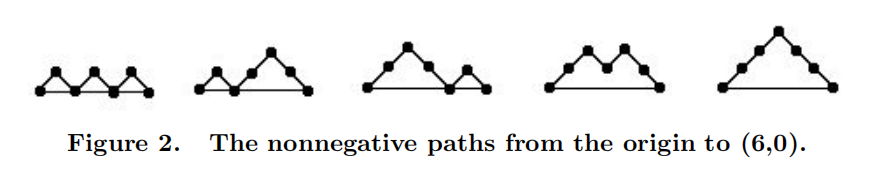
(+,−,+,−,+,−),(+,−,+,+,−,−),(+,+,−,−,+,−),

(+, −, +, −, +, −),(+, −, +, +, −, −),

这些路径的部分和序列分别是：

(0,1,0,1,0,1,0),(0,1,0,1,2,1,0),(0,1,2,1,0,1,0),

(0, 1, 0, 1, 0, 1, 0),(0, 1, 0, 1, 2, 1, 0),

这些路径如图2所示。

现在我们来证明非负路径和正路径的数量都是Catalan数。

**定理 1**  
从原点到 (2n,0) 的路径数量：  
(i) 正路径的数量是Catalan数 cn−1，  
(ii) 非负路径的数量是Catalan数 cn。

**证明**：  
我们首先建立正路径与非负路径之间的一个一对一对应关系。设 (s0,s1,…,s2n)是正路径 P 的部分和序列。则 s0=s2n=0，并且对于 i=1,2,…,2n−1 都有 si≥1。设 x=(x1,x2,…,x2n) 是对应的斜率序列，因此 xi=si−si−1∈{1,−1}。由于 s1≥1 且 s0=0，我们必须有 s1=1，因此路径 P 必定经过点 (1,1)。类似地，由于 s2n−1≥1 且 s2n=0，我们有 s2n−1=1，因此路径 P 必定经过点 (2n−1,1)。

如果我们从斜率序列中去掉第一个和最后一个项，就得到了一个序列x'，它有 n−1 个1和 n−1个-1。进一步，x′ 的部分和满足：

si′=x1′+x2′+⋯+xi′

=x2+x3+⋯+xi+1

=si+1−1≥0 对于 0≤i≤2n−2

因此，从原点到 (2n,0)的正路径 P 对应到从原点到(2n-2, 0) 的非负路径 P'。几何上，路径 P′ 是通过将路径 P 从 (1, 1) 到 (2n-1, 1) 这段部分，按新坐标系平移得到的，其中新坐标系的原点是 (1,1)。

由于路径 P 的第一项和最后一项必须分别是1和-1，因此从路径 P 到路径 P' 的操作是可逆的，即可以通过在路径 P' 前加上1，在末尾加上-1来恢复路径 P。因此，从原点到 (2n, 0) 的正路径数量 an 与从原点到 (2n−2,0)的非负路径数量 bn之间满足关系：

an=bn−1

**部分 (ii) 的证明：**  
接下来，我们需要证明 an=cn−1，这将使得 bn=an+1=cn。

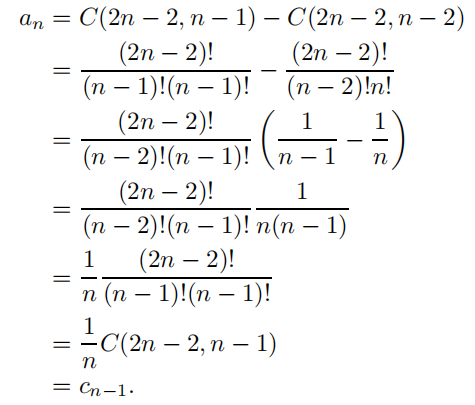
从原点到 (2n,0)的正路径 P 由它的从 (1,1) 到 (2n−1,1)的部分 P' 唯一确定，而这部分路径完全位于 x-轴之上。路径 P' 的数量等于从 (1,1)到 (2n−1,1)的总路径数减去从 (1,1)到 (2n−1,1)的那些与 x-轴相交的路径数量。

我们通过计数与 x-轴相交的路径来计算P' 的数量。设 Q 是从 (1,1) 到 (2n−1,1)的一条路径，且它首先在点 (k,0)与 x-轴相交。则 k≥2，因为路径 Q 从(1, 1)开始。如果我们将路径 Q 从 (1,1) 到 (k,0)这一段关于 x-轴反射，就得到了路径 Q' 从 (1,−1)到(k, 0)。然后，我们将 Q′的后一段（从 (k,0)到(2n-1, 1)）与反射后的部分拼接，得到路径 Q∗从 (1,−1)到 (2n-1, 1)。

每个这样的路径 Q∗必定与 x-轴相交，因此我们可以通过将路径 Q∗ 从 (1,−1)到 x-轴的第一个交点处反射，得到路径 Q。因此，路径 Q 与路径 Q∗一一对应。

对于路径 Q∗，我们知道它的增加步骤比减少步骤多两个。它有 n 个增加步骤，和 n−2个减少步骤，因此路径 Q∗的数量为 C(2n−2,n−2)，这也是路径 Q 的数量。

最后，正路径的数量 an就是从 (1,1)到 (2n-1, 1) 的正路径数量，等于总路径数 C(2n−2,n−1)减去与 x-轴相交的路径数 C(2n−2,n−2)，因此：



从而证明了正路径的数量是 cn−1，并得到了定理的结论。

**括号的合法序列**  
卡塔兰数计算了计算机科学中许多重要的结构。首先让我们考虑合法括号序列的集合，它可以递归定义（见《离散数学及其应用》第4.3节）。

**定义2** 如果且仅如果一个括号序列可以通过以下有限规则衍生出来，则该序列是合法的：

1. 空序列是合法的。
2. 如果 A 是合法的，则 (A) 也是合法的。
3. 如果 A 和 B 都是合法的，则 AB 也是合法的。

我们称第二条规则中跟随 A 的右括号关闭了前面 A 的左括号。

**例子3** 找到一个长度为 n=8的括号序列，其中一个是不合法的，一个是合法的。

**解答**：序列( ( ) ) ) ( ( ) 是不合法的，因为第三个和第四个左括号中只有一个可以被后面跟随的右括号关闭。但序列( ( ) ) ( ) ( ) 是合法的，因为每个左括号都被紧跟其后的第一个右括号关闭，且这些右括号不会关闭它们之间的左括号。

显然，每个合法的括号序列必须包含相等数量的左括号和右括号。此外，在任何合法括号序列的前 i 个括号（总长度为 2n）的初始字符串中，左括号的数量必须至少与右括号的数量相等。因此，如果我们在合法括号序列中将左括号替换为 1，右括号替换为 -1，我们就得到了一个序列 (x1,x2,…,x2n)∈{1,−1}^2n，其所有部分和 si≥0，因此它是从 (0,0)到 (2n, 0) 的非负路径。

反过来，任何从 (0,0)到 (2n, 0) 的非负路径都能生成一个合法的括号序列。根据定理1及其一一对应关系，我们可以得出以下结论：

**定理2** 长度为 2n的合法括号序列的数量是卡塔兰数 Cn。

**例子4** 找出长度为 2n=6 的合法括号序列。

**解答**：因为 n=3，根据定理2，正好有 C3=5 个这样的序列。它们分别是：

( ) ( ) ( ) , ( ) ( ( ) ) , ( ( ) ) ( ) , ( ( ) ( ) ) , ( ( ( ) ) ).

**堆栈排列**  
在《计算机程序设计艺术》第一卷的《基础算法》中，唐纳德·克努斯提出了一个问题，即计算在计算机中出现的某一类型的排列数量。堆栈是一种只能在一个特殊端（称为堆栈顶部）进行插入和删除的列表。当字符被一个一个地插入和删除时，最后插入的必须是最先删除的（后进先出，LIFO）。插入到堆栈顶部的操作称为推送（push），从顶部删除的操作称为弹出（pop）。我们可以将堆栈看作一堆盘子，其中推送操作表示将盘子放到堆栈顶部，而弹出操作表示从顶部移除一个盘子。

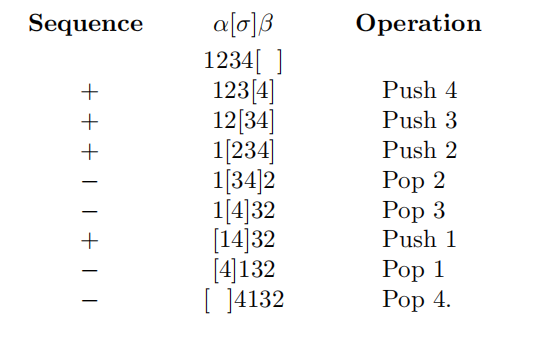
如果一组推送和弹出操作的序列具有相等数量的推送和弹出，并且在每个阶段，序列中push的数量至少与pop的数量相等，则该序列是可接受(admissible)的。如果我们将推送用 1 表示，弹出用 -1 表示，则一个可接受的序列对应于从原点到点 (2n,0) 的非负路径。因此，按照定理1，长度为 2n 的可接受推送和弹出序列的数量是卡塔兰数 Cn。

当在计算机中应用时，长度为 2n 的可接受推送和弹出序列将输入字符串的长度为 n 转换为一个输出字符串，该输出字符串包含相同的符号，但顺序可能不同。假设我们有一个初始输入字符串为集合 N={1,2,…,n} 的标准排列 123…n，以及一个长度为 2n 的可接受推送和弹出序列。每一次push操作将输入字符串的最后一个元素转移到堆栈top，每一次pop操作将堆栈顶部的元素转移到输出字符串的开头。在完成 n 次push和 n 次pop后，输出字符串将是 N 的一种排列，称为堆栈排列。

**例子5** 设 n=4，并考虑以下可接受序列：

(+,+,+,−,−,+,−,−)，其中加号表示推送，减号表示弹出。找出由这个推送和弹出序列产生的堆栈排列。

**解答**：我们将用一个形式为 α[σ]β 的字符串表示每一个可接受的推送和弹出操作的结果，其中 α\alpha 是当前输入字符串，β\beta 是当前输出字符串，[σ][\sigma] 是当前堆栈，堆栈顶部的元素是 σ\sigma 的第一个元素。然后，给定的可接受序列执行如下操作，产生堆栈排列 4132。



一个堆栈排列是指从 1,2,…,n 中通过一组可接受的推送和弹出操作得到的排列。但我们已经看到，长度为 2n 的可接受推送和弹出序列与从原点到点 (2n,0)的非负路径之间存在一一对应关系。堆栈排列可以通过对应的非负路径在平面上找到，具体如下：

1. 让 i 为横轴的索引，s 为纵轴的索引，假设路径通过点 (i,si)，其中 i=0,1,2,…,2n。让 k 为路径的最大纵坐标 si。
2. 绘制方程 s=j 的横线段，其中 j=1,2,…,k。由这些线条围成的区域将被称为第j 盒子。
3. 从左到右标记路径中的每个上升步骤为 n,n−1,…,1。
4. 对于 j=1,2,…,k，在盒子 j 中标记每个下降步骤为它前面的上升步骤的标签。

堆栈排列就是从右到左读取每个盒子中的下降步骤标签的顺序。

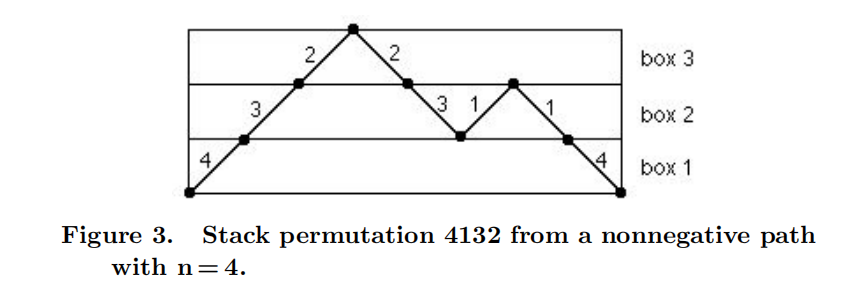
由于路径从原点开始，永不穿越 i-轴（方程 s=0），并且在 (2n,0)结束，因此每个盒子 j 必须包含相同数量的上升和下降步骤。此外，它们必须在盒子 j 中交替排列，从左到右分别是上升和下降步骤，且从上升步骤开始。因此，步骤（2）中的上升步骤和步骤（3）中的下降步骤的标记建立了上升步骤和下降步骤之间的一一对应关系，其中对应的步骤具有相同的标签。

在标记过程中，每个标记为 l 的上升步骤表示在该时刻被推送到堆栈中的元素 l，该元素只有在路径首次通过盒子 j 时，才会被弹出。

### 示例 6

画出示例 5 中给出的可接受序列所生成的非负路径，并通过标记步骤找到对应的堆栈排列。

**解答：** 路径的纵坐标序列是从可接受序列（斜率序列）(+,+,+,−,−,+,−,−)计算出的部分和序列：  
s=(0,1,2,3,2,1,2,1,0)

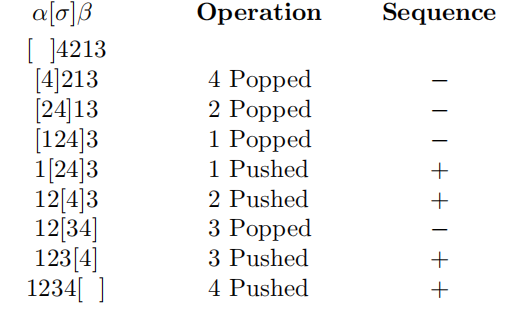
此路径的最大纵坐标 k=3，如图 3 所示，路径中包含 s=1,2,3的横线段（部分线段）。n=4 的上升步骤从左到右标记为 4, 3, 2, 1。然后，根据（3）对下降步骤进行标记。当从右到左读取这些下降步骤的标签时，生成的堆栈排列是 4132。

一个堆栈排列只能由一个唯一的可接受序列生成，因此由一个唯一的**非负路径**生成。我们可以通过以下方式从堆栈排列恢复可接受序列，从而恢复路径：  
从堆栈排列开始，按照示例 5 中的方式，在前面加上空的输入字符串 α和空的堆栈 [σ]。然后，开始构建一个空的推送-弹出序列。接下来，给定当前输入、堆栈和输出的内容 α[σ]β，其中 σ\sigma 和 β\beta 至少有一个非空，重复执行以下步骤：

1. 如果堆栈为空或 σ\sigma 和 β\beta 都非空，并且 β\beta 的第一个元素 b 小于 σ\sigma 的第一个元素 s，则将 b 转移到 σ\sigma 的左侧（b 刚刚被弹出），并在推送-弹出序列的右侧添加 −。
2. 如果输出为空或 σ\sigma 和 β\beta 都非空，并且 β\beta 的第一个元素 b 大于 σ\sigma 的第一个元素 s，则将 ss 转移到 α\alpha 的右侧（s 刚刚被推送），并在推送-弹出序列的右侧添加 +。

当输入字符串为 α=12⋯n，σ\sigma 和 β\beta 都为空时，生成堆栈排列的可接受序列就是构建的推送-弹出序列push-pop sequence。

### 示例 7

找出生成堆栈排列 4213 的可接受推送和弹出序列admissible sequence of pushes and pops。

**解答：** 从第三列的序列向上读取可得到可接受序列为 (+,+,−,+,+,−,−,−)。

从以上可以看出，堆栈排列的集合和非负路径的集合之间存在一一对应关系，或者堆栈排列的集合和可接受序列的集合之间也存在一一对应关系。

根据定理 1，我们可以得到以下定理：

### 定理 3

一个 n 元素集合的堆栈排列数量是卡塔兰数 Cn。

### 良好括号化的乘积Well-parenthesized Products

考虑一个具有二元运算的代数结构 S，我们称该运算为乘法。通常情况下，我们可以用 xy 来表示 x,y∈S 的乘积。

假设该运算是**非交换**的，也就是说，一般情况下 xy≠yx。

当乘法是结合律associative的，即对于所有 x,y,z∈S，有 (xy)z=x(yz)，那么 n+1 个元素 x1x2⋯xn+1 的乘积是定义良好的welldefined。

但如果假设乘法在 S 中不是结合的，那么乘积 x1x2⋯xn+1只有在适当插入括号后，才能递归地确定要乘的元素对。我们将没有括号的序列 x1x2⋯xn+1 简单地称为**乘积(product)**。

我们将确定如何为乘积 x1x2⋯xn+1插入括号。假设二元运算是非交换且非结合的，这允许我们将此数量解释为通过括号化乘积，得到的 S 中不同元素的最大数量。然而，我们也可以考虑 xi 是实数，二元运算是普通乘法。在这种情况下，我们寻求通过连续的两个数相乘来计算乘积 x1x2⋯xn+1 的不同方式。这正是卡塔兰最初对这个问题的表述。

### 示例 8

找出乘积 x1x2x3x4 的不同括号化方式。

**解答：** 每当左括号由右括号闭合，并且我们已经完成了由括号内嵌的成对元素定义的乘积时，我们必须有一个恰好包含两个元素的乘积。乘积 x1x2x3x4的良好括号化方式为：

((x1(x2x3))x4),(x1((x2x3)x4)),((x1x2)(x3x4)),

(x1(x2(x3x4))),(((x1x2)x3)x4)

注意，在示例 8 中，我们使用了 n=3对括号来括号化每个乘积。虽然不是必要的，但包括外部括号是方便的，其中左括号排在前，右括号排在后。

### 定义 3

如果一个乘积可以通过以下规则的有限序列递归地获得，则该乘积是良好括号化的well-parenthesized：

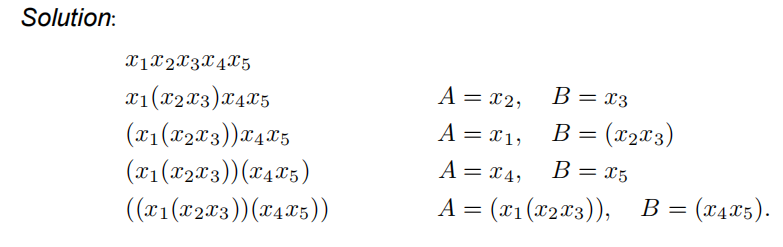
1. 每个单一项 x∈S 都是良好括号化的。
2. 如果 A 和 B 都是良好括号化的，则 (AB) 是良好括号化的。

注意，良好括号化的乘积well-parenthesized product（除了 S 的单一元素之外）包括外部括号。因此，(xy) 是良好括号化的，而 xy 不是。

我们可以使用数学归纳法证明，为了形成一个良好括号化的乘积，必须在 x1x2⋯xn+1中添加 n 对括号。

如果从一个良好括号化的乘积中删除 n+1 个变量 x1,x2,…,xn+1，则剩下的 n 对括号必须是一个良好形成的括号序列。但并非每个良好形成的括号序列都可以通过这种方式得到。例如，( ( ) ( ) ( ) ) 是 n=4对括号的良好形成序列，但由于该运算是二元的，外部括号要求在内部括号中计算的三个元素的乘积是未定义的。

### 示例 9

通过在每一步识别A和B，证明 ((x1(x2x3))(x4x5)) 是通过递归定义从积 x1x2x3x4x5获得的，因此它是一个良好括号化的积。

我们现在将展示良好括号化的积 x1,x2,…,xn+1和从原点到点 (2n,0) 的非负路径之间的唯一对应关系。一个良好括号化的积形成一个长度为 3n+1 = n+1+2n 的字符串 p，其中有三种类型的字符：n个左括号，n+1个变量，以及n个右括号。非负路径的斜率序列从原点到点 (2n,0)仅包含两个不同的数字：1和-1，每个数字出现n次。

为了从字符串p中获得一个结构与斜率序列相似的序列，我们形成一个长度为 2n 的字符串 q=S(p)，通过删除最后一个变量 xn+1 和n个右括号。

### 示例 10

找出字符串 q，如果 p=((x1(x2x3))(x4x5))是例子8中的良好括号化积。

**解答：** 删除 x5和右括号后，字符串 p 变为：

q=((x1(x2x3(x4

接下来我们分析这个从良好括号化积 p 中获得的字符串 q=S(p)的性质。我们将证明，给定这样的字符串 q，我们可以从它中**恢复出积 p**。根据 q 的定义，首先我们注意到以下几点：

**引理1：** 字符串 q=S(p) 包含n个左括号，并且按顺序包含变量 x1,x2,…,xn。

由于在 p 中，xn 和 xn+1之间不可能有左括号，因此字符串 q 的最后一个字符必须是 xn。这个结论实际上是由更一般的性质所隐含的，类似于非负路径所满足的性质，我们在以下引理中给出该性质。

**引理2：** 设 q=S(p)，其中 p 是由变量 x1,x2,…,xn+1 构成的良好括号化积。对于 i≤2n，字符串 q1q2…qi 中的左括号数目至少与变量的数目相同。

**证明：** 我们将通过数学归纳法证明该引理。首先，当 n=1 时，我们必须有 p=(x1x2)，因此 q=(x1结论成立。

假设对于任意 n≥2，结论在变量数目为 k≤n时成立。现在考虑由 n+1个变量构成的良好括号化积 p。根据递归定义，p=(AB)，其中 A 和 B 是两个非空的良好括号化积。p 中的第一个左括号出现在AB 之前，因此在 p 中 B 前面每个字符的左括号数目比在 A 中多一个。但是，如果 xk 是 A 中的最后一个变量，那么它在 S(A) 中不出现，而是在 q=S(p) 中出现在 xk−1 之后。因此，直到达到 xk，q 中左括号数与变量数之间的差距比在 S(A) 中大1，而当达到 xk 时，差距为0。之后，B 中每个字符的差距与在 p 中的差距相同。由于 A 是一个由 k≤n 个变量组成的良好括号化积，所以根据归纳假设，字符串 q1q2…qi对于 i≤2n中的左括号数目至少与变量数目相同。

我们说，满足引理1和引理2的字符串q 是合适**suitable**的。

### 定理 4

长度为 2n 的suitable字符串集合 和 长度为 3n+1的良好括号化积集合之间存在一一对应关系。

**证明：** 我们将展示函数 S 是一一对应的。设 q 是一个合适的字符串。我们需要证明，可以从字符串 q 中恢复出良好括号化积 p，使得 q=S(p)。首先，我们将 xn+1 附加到 q 的右边，并将新的字符串记为 q′=**q**xn+1。根据引理1，我们得到以下结论：

**引理1'：** 字符串 q' 包含n个左括号，并且按顺序包含变量 x1,x2,…,xn+1。

由于 q 和 q' 在前 2n 个位置相同，我们将表示 q' 中第 i 个字符的字符记为 qiq\_i，对于 i≤2ni \leq 2n。然后 q′q' 满足引理2的结论。

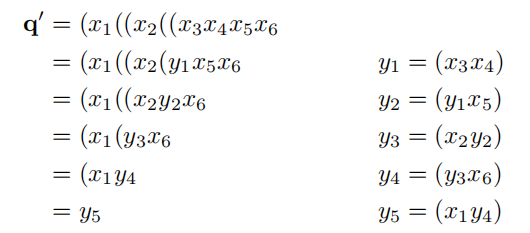
接下来，我们通过**数学归纳法**证明定理。如果 n=1，则 q′=(x1x2，因此我们必须有 p=(x1x2)。

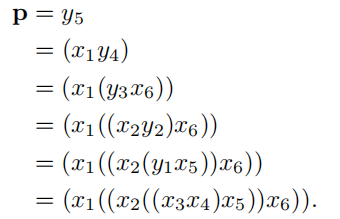
假设对于 n≥2 且定理对于 n−1 成立时也成立。根据引理1和引理2，q 的最后两个字符是 xn−1 xn 或者是 (xn，因此 q′的最后三个字符要么是 xn−1xnxn+1，要么是 (xnxn+1。根据引理1'，无论哪种情况，q' 都会有三个连续字符形式为 (xjxj+1)（对于某个 j≥1）。设 j1 为最小的这样的 j。当向 q' 中插入右括号形成良好括号化积时，右括号必须紧跟在 (xj1xj1+1) 之后，因此 p 中将包含 (xj1xj1+1)。将 (xj1xj1+1) 替换为一个新变量 y1后，得到新的字符串 q1。那么，字符串 q1 包含 n−1个左括号和 n 个变量，且满足引理1'的结论，变量数为 n−1。根据归纳假设，我们可以恢复出良好括号化积 p1。将 (xj1xj1+1) 替换为 y1 后，得到一个良好括号化积 p，使得 q=S(p)。由于 SS 是一一对应的且可逆，因此它是一个一一对应关系。

### 示例 11

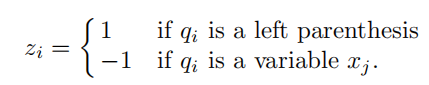
从适合的字符串  
q=(x1((x2((x3x4x5恢复出正确配对的乘积 p，使得 q=S(p)。

**解答：**  
我们通过将 x6 加到 q 的右边，构成 q′=(x1((x2((x3x4x5x6。  
然后，在每一步 j 中，找到第一个紧跟着两个变量的左括号，并用新变量 yj替换这个长度为 3 的子串，其中 yj等于该子串加上一个右括号作为第四个字符。



然后，设 p=y5，并依次替换新变量 y5,y4,…,y1 的值，得到：

**推论 1**  
n+1 个变量的正确配对乘积的数量是卡塔兰数 Cn。

**证明：**  
根据定理 4，从 x1x2…xn+1到每个正确配对的序列对应一个长度为 2n 的适合字符串 q，其中有 n 个左括号和 n个变量 x1,x2,…,xn。定义一个序列 z=(z1,z2,…,z2n)，其中：

根据引理 2，序列 z 的部分和 si 非负。因此，对应于 q 的是从 (0,0) 到 (2n,0)的非负路径。根据定理 1，正确配对的乘积的数量是 Cn。

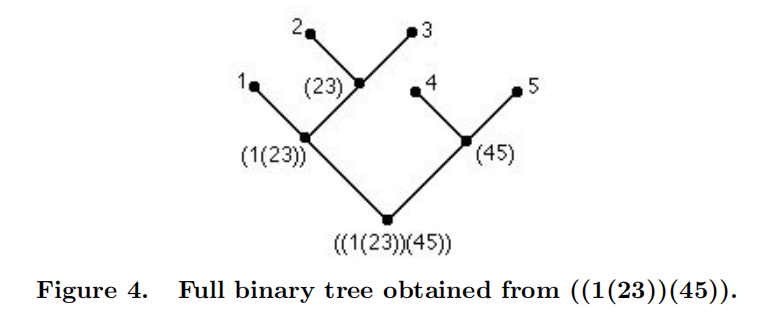
满二叉树Full Binary Trees  
回顾（见《离散数学及其应用》第 10.1 节），满二叉树是一个根树，每个内部顶点internal vertex恰好有两个子节点。因此，具有 n 个内部顶点的满二叉树有 2n 条边。由于树比边多一个顶点，具有 n 个内部顶点的满二叉树 T 有 2n+1 个顶点，因此有 n+1 个叶子。假设我们沿横截面（前序遍历、中序遍历或后序遍历，见《离散数学及其应用》第 10.1 节）标记树的叶子，标记为 x1,x2,…,xn+1。然后，树递归地定义了 x1,x2,…,xn+1 的正确配对乘积，规则如下：

**标记规则：**  
如果 v 是一个内部顶点，其左子节点为 a，右子节点为b，并且分别标记为 A 和 B，则将 v 标记为 (AB)。  
树的根节点上的标签将是well-parenthesized product该正确配对的乘积。

相反，给定一个 n+1 个变量的正确配对乘积 x1,x2,…,xn+1，通过以下步骤确定一个标记的满二叉树：首先将根节点标记为正确配对的乘积，然后从外括号向内移动，向每个标记为 (AB)的顶点添加两个子节点，分别标记为 A 和 B。树的叶子将按照遍历的顺序标记为变量 x1,x2,…,xn+1。因此，n+1个变量的正确配对乘积和具有 n+1个叶子和 n 个内部顶点的满二叉树之间存在一一对应关系。由定理 4，我们得出以下结论：

**定理 5**  
具有 n 个内部顶点的满二叉树的数量是卡塔兰数 cn。

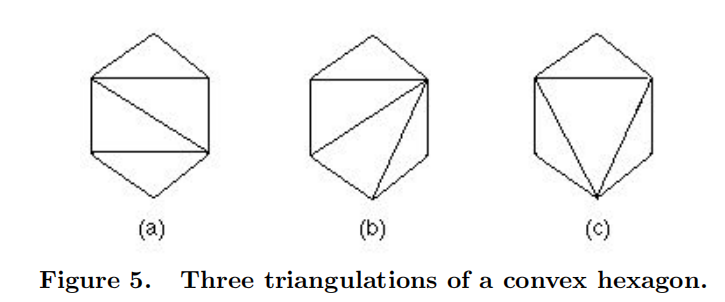
### 示例 12 画出并标记由正确配对的乘积 ((1(23))(45))定义的满二叉树。

**解答：**  
该满二叉树如图所示。

**凸多边形的三角剖分Triangulations of a Convex Polygon**

在本节中，我们考虑卡塔兰数的几何解释。一个平面中的**n-gon**( n-边形（n≥3）)是一个具有 n 个顶点和 n 条边的多边形。设 v0,v1,…,vn−1 为顶点（按逆时针顺序）。我们用 vivj来表示连接顶点 vi和 vj的线段line segment。则 P 的 n条边分别是 si=vi−1vi（对于 1≤i≤n−1）和 s0=vnv0。P 的对角线diagonal是连接两个不相邻顶点的线段 vivj。当 P是凸convex的时，每条对角线都完全位于 P 的内部。

设 D 是一组对角线，满足没有两个对角线在凸多边形 P 内相交，且它们将 P 的内部分割成三角形。三角形的边要么是 D 中的对角线，要么是 P 的边。由此得到的三角形集 T 称为triangulation of P (P 的三角剖分)。图 5 显示了三种凸六边形的三角剖分。



接下来我们将确定形成三角剖分所需的对角线数量。

**引理 3**：一个凸多边形的三角剖分包含 n−2个三角形，且由 n−3 条对角线决定。A triangulation of a convex n-gon has n−2 triangles determined by n − 3 diagonals.

**证明**：我们通过**数学归纳法**来证明。若 n=3，则只有一个三角形，且没有对角线；若 n=4，则有两个三角形，一条对角线。假设 n≥5 且结论对于凸 m-边形的三角剖分成立，其中 3≤m<n。

设 P 为一个凸的 n-边形，且 T 是 P 的三角剖分，包含对角线集合 D。因为 n>3，所以 D 中必定至少有一条对角线，设为 v0vk，其中 2≤k≤n−2。对角线 v0vk将联合形成两个凸多边形：一个是具有顶点 v0,v1,...,vk的凸 k+1-边形 P1，另一个是具有顶点 v0,vk,vk+1,...,vn−1 的凸 (n−k+1)-边形 P2。

对于这两个凸多边形 P1和 P2，它们的三角剖分分别由 D1 和 D2 定义，其中 D1和 D2是对角线集合 D 中去除 v0vk后的子集。由于 k+1≤n 且 n−k+1≤n，我们可以应用归纳假设。因此，T1和 T2 分别包含 k−1 和 n−k−1个三角形，且由 k−2 和 n−k−2条对角线决定。

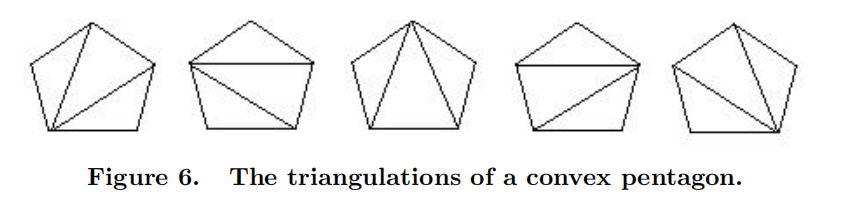
将两者的三角形数相加得：(k−1)+(n−k−1)=n−2个三角形。而对角线的数量为 (k−2)+(n−k−2)=n−4，加上对角线 v0vk，总共有 n−3 条对角线。

现在我们可以假设 P 是一个凸的 (n+2)-边形，其中 n≥1，具有顶点 v0,v1,...,vn+1和边 s0,s1,...,sn+1。根据引理 3,P 的每个三角剖分由 n−1 条对角线定义，且有 n 个三角形。

设 tn 为一个凸的 (n+2)-边形 P 的三角剖分的数量。那么显然有 t1=1和 t2=2。

### 例子 13：

画出所有凸五边形的三角剖分，证明 t3=5。

**解**：由于 n=3，每个三角剖分由三个三角形和两条对角线定义。如果 vivj是其中一条对角线，另一条对角线必须与 vi或 vj相交。因此，两条对角线都必须连接共同的端点到一个非邻接的顶点。我们可以找到如图 6 所示的五种三角剖分。

考虑通过按顺序遍历 P 的 n+1 条边（不包括 s0）来形成字符串 s1s2…sn+1。我们将证明，由三角剖分 T 形成的产品 s1s2…sn+1 是良好括号配对的。这里排除了边 s0，使得剩余的边形成一个平面中的开放路径。然后，按路径的顺序，边形成的序列类似于之前考虑的 n+1 个变量的乘积。

设 T 是由 n−1条对角线定义的 P 的三角剖分，其中 n≥4。每条边 si都是 T 中某个三角形的边。T 中包含 P 边的三角形有两种类型：

(i) 外部三角形outer triangle的三条边是 P 的两条相邻边 si,si+1和对角线 vi−1vi+1。

(ii) 内部三角形inner triangle 的三条边是 P 的一条边 si=vi−1vi 和两条对角线 vi−1vj,vivj，其中 j=i−2或 i+1。

例如，图 5 中的六边形（a）和（b）各有两个外部三角形，而（c）有三个外部三角形。为了建立triangulations and well-parenthesized products(三角剖分和良好括号配对产品)之间的单射对应关系，我们必须证明每个三角剖分都有一个外部三角形不包含 s0 作为边。

**引理 4**：如果 n≥2，每个凸的 (n+2)-边形 P 的三角剖分 T 至少有两个外部三角形。

**证明**：**反证法。**假设 T 中至多有一个三角形包含 P 的两条边。设 ni 为第 i个三角形包含的 P 边的数量。那么 ni=1 对于至少 n−1 个三角形，且 ni=2 对于至多一个三角形。当我们将这些 ni相加时，我们得到 P 至多有 n+1 条边。然而P 有 n+2 条边，所以得出矛盾。因此，P 必须有至少两个外部三角形。

### 定理 6：

一个凸的 (n+2)-边形的三角剖分的数量是 Catalan 数 Cn。

**证明**：根据推论 1，我们只需要建立一个凸 (n+2)-边形 P 的三角剖分 与 P 除s0外n+1条边的well-parenthesized products之间的单射对应关系。

**数学归纳法**

设 n≥2，并假设这种单射对应关系对于 (n+1)-边形成立。设 T 为 P 的三角剖分。由于 P 的每一条边都是正好一个三角形的边，并且根据引理 4，至少有两个外部三角形，因此 T 中必定有一个外部三角形没有包含 s0 作为边。这个三角形的顶点是 vi−1,vi,vi+1（其中 1≤i≤n），它的边是P 的两条相邻边 si,si+1和对角线 vi−1vi+1。

将该对角线标记为 (sisi+1)。如果我们删除顶点 vi，并将边 si,si+1 替换为标记为 (sisi+1) 的对角线，那么我们得到了一个凸的 (n+1)-边形 P'。根据归纳假设，我们可以在 ′P' 上建立三角剖分T' 和良好括号配对的产品之间的单射对应关系。但是，P' 的一个边被标记为良好括号配对的产品 (sisi+1)。因此，P′ 的边的良好括号配对产品表示 P 的边的良好括号配对产品。

反之，P 的边的良好括号配对产品中每一对括号内的最内层括号表示该对中的两条边属于一个外部三角形。然后，完成外部三角形的对角线必须包含在 D 中。每次闭括号都会通过此方式添加对角线，直到得到 P 的三角剖分。

### Catalan数所计数的对象总结

我们已经建立了

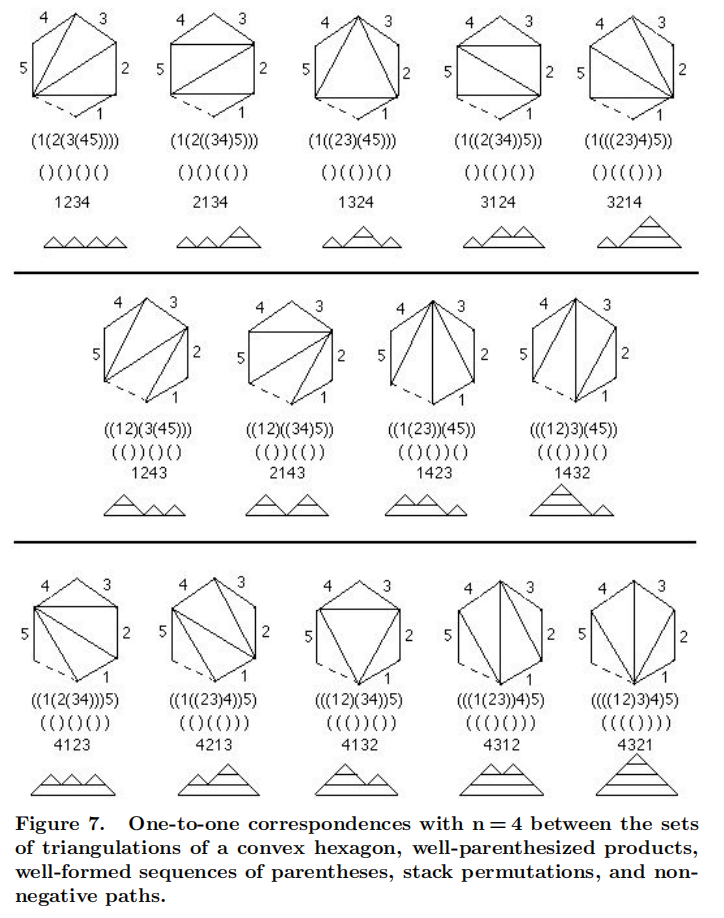
凸的(n+2)-边形的三角剖分集triangulations of a convex (n+2)-gon,、

n+1个变量的良好括号配对乘积well-parenthesized products of n+1 variables、

n对括号的良好排列well-formed sequences of n pairs of parentheses、

12到n的栈排列stack permutations of 12 ···n,

从原点到点(2n,0)的非负路径nonnegative paths from the origin to the point (2n, 0)

之间的双射关系。对于n=4的情况，这些关系在图7中得到了说明。

在六边形中，未对应于n+1个变量的良好括号配对乘积中的变量的边s0用虚线表示。为了更清晰起见，其他每一条边si，对应于良好括号配对的变量中的一个，简单地标记为i。

### Catalan数的生成函数

我们一开始通过一个公式定义了Catalan数cn，然后通过组合论的论证证明它枚举了平面中的非负路径。随后，我们找到了几个不同类型的组合结构之间的双射关系，从非负路径开始。由此得出，每种类型的结构的数量必须等于非负路径的数量，也即Catalan数。既然已经建立了双射关系，如果我们通过组合论或其他方式展示某一特定类型的结构的数量可以表示为：

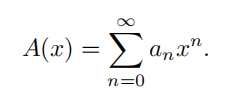
那么就得出了相同的结论。

在本节中，我们将通过递推关系和生成函数来推导上式（1），即n+1个变量的well-parenthesized products的数量（参见《离散数学及其应用》第7.1节和7.4节）。在下一节中，我们将研究Catalan数序列{cn}对于大n值的渐近行为。

如果每个n≥k的项an可以通过递推关系从前k项计算得到：一个序列{an}=a0,a1,…满足线性齐次递推关系（常系数，次数为k）

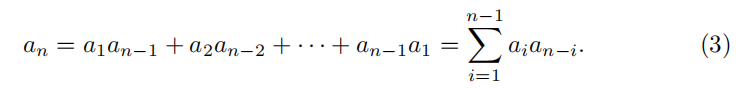
其中Ck=0，且常数Ci满足1≤i≤k。任何满足该递推关系的序列，由其初始的k个值完全确定。

序列{an}的生成函数是幂级数：

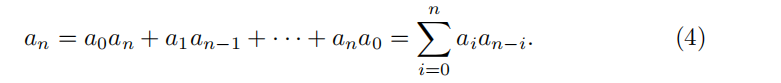


如果该序列满足线性齐次递推关系，并且具有常系数，生成函数将是一个有理函数。分子中的多项式只依赖于初始的k个值，而分母中的多项式则只依赖于递推关系。一旦找到（或通过算法估计）分母中多项式的根，就可以通过部分分式和已知的幂级数计算序列的任何项的值。

假设我们定义an为n个变量的良好括号配对乘积的数量。我们之前已经证明，an+1就是Catalan数cn，由公式（1）给出。但我们将通过递推关系和生成函数来求解an。为了通过递归定义形成一个n≥2个变量的良好括号配对乘积，外层括号是最后添加的，形成(AB)，其中A和B是良好括号配对的乘积，且每个乘积包含至少一个变量。然后，外层括号将包含AB，其中A是前i个变量的良好括号配对乘积，B是后n−i个变量的良好括号配对乘积，满足1≤i≤n−1。然后，通过乘法法则和加法法则，我们得到：

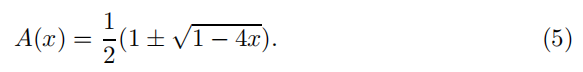
注意，递推式（3）是一个具有常系数的齐次递推关系。然而，它不是线性的，而是二次的。此外，由于确定an的前几项的数量取决于n，递推关系的次数在（3）中并未定义。

如果我们定义a0=0，则可以将项a0an和ana0添加到（3）的右边，而不改变其值，得到：

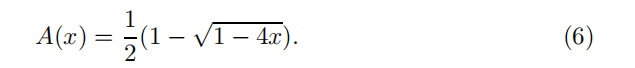
对于n≤1，（4）右边的和为0，但对于n≥2，它是生成函数的平方中x^n的系数。由于a1=1，所以有：

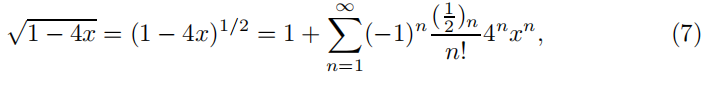


这就是关于生成函数的二次方程。通过二次公式，我们得到：

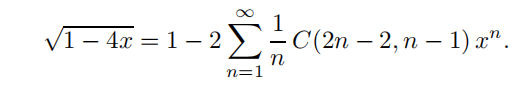


由于A(0)=a0=0，并且当x=0时，右边的带加号的部分为1，我们必须取减号。因此，生成函数为：

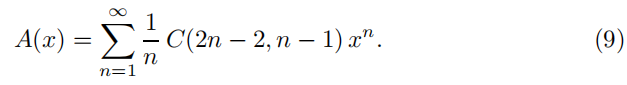


使用二项式定理的广义形式，可以证明：

通过适当的代数变换，可以得到：



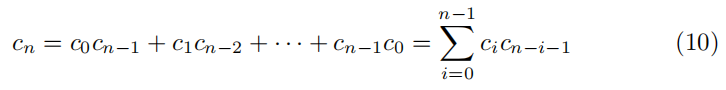
因此，生成函数为：



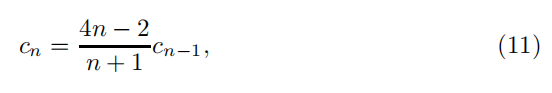
所以，x^{n+1}的系数就是Catalan数Cn，即：

### Catalan数的渐近行为 Asymptotic Behavior

让我们考虑Catalan数序列{cn}对于大n值的渐近行为。在上一节中，我们定义了an为n个变量的良好括号配对乘积的数量，并证明序列{an}满足递推关系（3）。利用生成函数，我们发现an+1等于Catalan数cn。如果我们将cj代入（3）并调整索引变量i的范围，可以得到：

这也是一个二次的齐次递推关系，但是次数未定义。为了更容易地找到序列{cn}的渐近行为，我们需要通过展示它满足一个线性递推关系来得到，它是一次的，但有可变的系数。

通过进一步推导，可以得到：



显然，常系数C会随着n→∞而增大到4。但这并不意味着cn可以通过常数倍的4^n来逼近。

通过斯特林公式的近似，可以证明：

