

我们介绍并研究一个非常重要的组合结构——树。树的计数是计数组合学中一个重要的问题。下面，先给出若干类树的定义。

定义 1. 一个有限顶点集 T 称为是有根树 (rooted tree), 若它满足:

- a. 有一个特殊的顶点 v_0 , 称为 T 的根 (root);
- b. 其余顶点 (除根外) 被划分为 $m \geq 0$ 个不交的非空集合 T_1, \dots, T_m , 每个集合都构成一棵树。树 T_1, \dots, T_m 称为根的子树 (sub tree)。

在一棵有根树 T 上, 其每一个子树 (以及子树的子树, 子树的子树的子树等等) 的顶点称为 T 的一个结点 (vertex)。对于每个结点 $v \in T$ (包括顶点 v_0), 它对应的子树称为以 v 为根的子树或由 v 生成的子树, 若以 u 为根的子树是 v 的子树, 则称 u 是 v 的子结点, v 是 u 的父结点。一个结点 s 的子结点个数成为这个结点的度 (degree), 记做 $d_T(s)$, 在不引起歧义的前提下, 简记做 $d(s)$ 。度为 0 的结点称为叶子点 (leaf)。

若忽略有根树的根的指派, 即只考虑树的顶点之间的连接关系, 则得到一个图论意义上的树, 这里我们称之为“自由树”。在自由树中, 没有根的定义, 自然就没有了父结点和子结点的概念, 其中结点的度定义为与它相邻的点的个数, 同样用 $d(s)$ 标记。称度为 1 的结点为叶子点。

有根树主要为了和自由树进行区别, 在不影响理解的情形下, 有根树简称为“树”。修改定义 1 中的第二条, 可以得到其他特殊的树。如下:

定义 2. 一个有限顶点集 T 称为平面树 (plane tree) 或有序树 (ordered tree), 若 T 满足条件 a 及如下条件

- b'. 其余顶点 (除根外) 被有序地划分为 $m \geq 0$ 个不交的非空集合 T_1, \dots, T_m , 每个集合都构成一棵平面树。树 T_1, \dots, T_m 称为根的子树。

定义 3. 一个有限顶点集 T 称为 m 元树 (m -ary tree) ($m \geq 2$), 若 T 满足如下条件

- a'. T 或者是空树, 或者有一个特殊的顶点 v_0 , 称为 T 的根 (root);
- b'. 其余顶点 (除根外) 被有序地划分为 m 个不交的集合 (可能空) T_1, \dots, T_m , 每个集合都构成 m 元树。树 T_1, \dots, T_m 称为根的子树。

2 元树也称二叉树 (binary tree), 在计算机科学领域中有很重要的应用, 很多算法的分析都和二叉树有关。二叉树中, 顶点的两个子树分别被称为左子树和右子树。

若每个非叶子点的 m 个子树皆非空, 则称此 m 元树为完全的 (complete)。

由以上定义可知, 空树 (即不含结点的树) 属于二叉树, 但不属于有根树和平面树。

按树中的结点是否加以区别, 可以将树分为有标号的 (labeled) 和无标号的 (un-labeled) 两种。我们这里主要讨论有标号树的计数。对于平面树及 m 元树, 由于每个结点都是不同的, 有标号树与无标号树恰好相差 $n!$ 倍, 因此我们只需计数对应的无标号树即可。对于有标号树, 一个很重要的问题是研究它的标号之间的大小关系, 我们在本章的最后 3 节着重研究。

接下来的几节里, 我们将逐个研究这些树的计数问题。在此, 先给出将要用到的两个重要定理。

对于任意一个形式幂级数 $f(x) = a_0 + a_1x + a_2x^2 + \dots$, 称另一个形式幂级数 $g(x)$ 是 $f(x)$ 的复合逆 (compositional inverse), 若 $f(g(x)) = g(f(x)) = x$, 记作 $g(x) = f^{<-1>}(x)$ 。

定理 4. (拉格朗日反演公式) (*The Lagrange inversion formula*) [?, p.38]. 记 $F(x) = a_1x + a_2x^2 + \cdots$ 是常数项为 0 的形式幂级数, 且 $a_1 \neq 0$, 则 $F^{<-1>}(x)$ 存在且唯一, 且对于任意 $k, n \in \mathbb{Z}$, 有

$$n[x^n]F^{<-1>}(x)^k = k[x^{n-k}] \left(\frac{x}{F(x)} \right)^n = k[x^{-k}]F(x)^{-n}.$$

等价地, 若对于形式幂级数 $G(x)$, 其中 $G(0) \neq 0$, 而 $f(x) = xG(f(x))$, 则

$$n[x^n]f(x)^k = k[x^{n-k}]G(x)^n.$$

特别的, 当 $k = 1$ 时, 有

$$[x^n]f(x) = \frac{[x^{n-1}]G(x)^n}{n}.$$

定理 5. (指数公式) (*The Exponential Formula*) [?, p.5]. 给定函数 $f: \mathbb{P} \rightarrow K$, 定义函数 $h: \mathbb{P} \rightarrow K$ 如下:

$$\begin{aligned} h(\#S) &= \sum_{\{B_1, \dots, B_k\}} f(\#B_1)f(\#B_2) \cdots f(\#B_k), \#S > 0 \\ h(0) &= 1 \end{aligned}$$

则有

$$E_h(x) = \exp(E_f(x))$$

其中 $E_f(x) = \sum_{n \geq 0} f(n)x^n/n!$

接下来的几节里, 我们将利用这两个强大的工具, 对一些常见的树的计数问题进行研究。

1 有根树的计数

记 n 个结点的有根标号树的集合为 \mathcal{R}_n , 记其个数为 r_n . 记所有含 n 个结点的有标记自由树的集合为 \mathcal{F}_n , 个数为 f_n . 对于任意一个自由树, 标记其任意一个结点为根, 可以得到一棵有根树。故 $r_n = nf_n$. 我们有以下结论:

定理 6. (凯莱公式) (*Cayley's formula*) $f_n = n^{n-2}$.

定理实际上告诉我们, 在自由树与 $n-2$ 长的序列之间存在一个一一映射。但我们不准备给出双射, 而是采用生成函数的方法证明。

证明 我们只需要证明 $r_n = n^{n-1}$. 我们定义一个图 T 为有根森林, 若它的每个连通分支都是有根树。记 n 个结点的有根森林的集合为 \mathcal{RF}_n , 个数为 t_n . 记 $R(x) = \sum_{n=0}^{\infty} r_n x^n/n!$, $RF(x) = \sum_{n=0}^{\infty} t_n x^n/n!$. 由指数公式 $RF(x) = e^{R(x)}$. 另一方面, 易知 $r_n = n \times t_{n-1}$, 故 $R(x) = xRF(x) = xe^{R(x)}$.

由拉格朗日反演公式,

$$\begin{aligned}
 r_n &= n! \times [x^n]R(x) \\
 &= n! \times \frac{1}{n} [x^{n-1}](e^x)^n \\
 &= n! \times \frac{1}{n} \frac{n^{n-1}}{(n-1)!} \\
 &= n^{n-1}.
 \end{aligned}$$

□

2 二叉树的计数

为了计数无标号二叉树的个数, 我们考虑其普通型生成函数。类似于上一节, 定义 n 个结点的二叉树的集合为 \mathcal{B}_n , 个数为 b_n , 其普通型生成函数为 $B(x)$. 则首先我们有如下的结论:

定理 7. (1.) b_n 满足如下递推关系: 当 $n = 0$ 时, $b_0 = 1$; 当 $n \geq 1$ 时,

$$b_n = \sum_{i=0}^{n-1} b_i b_{n-1-i}$$

(2.) $B(x)$ 满足函数方程 $B(x) = 1 + xB^2(x)$.

证明

(1.) 由定义, 显然空树属于二叉树, 因此 $b_0 = 1$. 对于任意 $n \geq 1$ 个结点的二叉树, 我们从中选择一个结点作为根, 再将其余的 $n - 1$ 个点任意的分到两个子树中。对于两个子树, 对应的方案数分别为 b_i 与 b_{n-1-i} , 故

$$b_n = \sum_{i=0}^{n-1} b_i b_{n-1-i}.$$

(2.)

$$\begin{aligned}
 B(x) &= 1 + \sum_{n=1}^{\infty} b_n x^n \\
 &= 1 + \sum_{n=1}^{\infty} \sum_{i=0}^{n-1} b_i b_{n-1-i} x^n \\
 &= 1 + x \sum_{n=0}^{\infty} \sum_{i=0}^n b_i b_{n-i} x^n \\
 &= 1 + x \sum_{i=0}^{\infty} b_i x^i \sum_{j=0}^{\infty} b_j x^j \\
 &= 1 + x B^2(x).
 \end{aligned}$$

□

利用拉格朗日反演公式，可得如下二叉树的计数公式：

具有 n 个顶点的二叉树的个数为第 n 个卡特兰数，即 $C_n = \frac{1}{n+1} \binom{2n}{n}$.

证明 令 $F(x) = B(x) - 1$ ，则对于 $n > 0$ ， $[x^n]B(x) = [x^n]F(x)$ ；另一方面，由定理 7， $F(x) = x(F(x) + 1)^2$. 利用拉格朗日反演公式，得

$$\begin{aligned} b_n &= [x^n]B(x) = [x^n]F(x) \\ &= \frac{1}{n} [x^{n-1}]((x+1)^2)^n \\ &= \frac{1}{n} \binom{2n}{n-1} \\ &= \frac{1}{n+1} \binom{2n}{n}. \end{aligned}$$

□

完全类似的思路，可以将以上的结论推广到任意 m 元树。

定理 8. 记具有 n 个顶点的 m 元树的个数为 $C(m, n)$ ，记其普通型生成函数为 $C_m(x)$. 则有

$$(1.) C_m(x) = 1 + xC_m(x)^m.$$

$$(2.) C(m, n) = \frac{1}{n} \binom{mn}{n-1}.$$

证明 与上面定理和推论完全类似，略去。 □

注：

(1.) $C(m, n)$ 有时被称作推广的卡特兰数。

(2.) 可以证明：具有 n 个中间点（即非叶子点）的完全二叉树的个数是卡特兰数；类似的，具有 n 个中间点的完全 m 元树的个数是推广的卡特兰数。事实上，我们可以利用如下简单的对应，将完全二叉树对应于普通二叉树，使得得到的二叉树的结点数等于原来二叉树的中间点的个数：

对于任意一个完全二叉树，将所有叶子点去掉，得到一个二叉树；反过来，对于任意一个二叉树，在每个结点的空子树上添加只有一个结点的子树，得到一个完全二叉树。

m 元树的对应完全类似。

3 平面树的计数

定理 9. 记 n 个点的平面树的全体为 \mathcal{P}_n ，个数为 p_n ，其普通型生成函数为 $P(x)$. 则有：

(1.) $P(x)$ 满足函数方程

$$P(x) = \frac{x}{1 - P(x)}.$$

(2.) n 个点的平面树的个数为

$$p_n = C_{n-1} = \frac{1}{n} \binom{2n-2}{n-1}.$$

证明 (1.) 由平面树的定义知, $p_0 = 0$. 对于任意一个平面树, 去掉其根, 得到一个平面森林。若平面森林的连通分支为 i , 则对应的森林的生成函数为 $P(x)^i$, 故有

$$P(x) = \sum_{i=0}^{\infty} x P(x)^i = \frac{x}{1 - P(x)}.$$

(2.) 由拉格朗日反演公式, 有

$$\begin{aligned} p_n = [x^n]P(x) &= \frac{1}{n} [x^{n-1}] \left(\frac{1}{1-x} \right)^n \\ &= \frac{1}{n} \binom{-n}{n-1} (-1)^{n-1} \\ &= \frac{1}{n} \binom{2n-2}{n-1}. \end{aligned}$$

细心的同学可能已经注意到 p_n 恰好也被卡特兰数所计数, 事实上, 卡特兰数是组合数学中一个很重要的数列, 最早由比利时数学家欧仁·查理·卡特兰 (1814-1894) 研究。除我们这里讲的两种有序树以外, 很多组合结构都被其计数, 在参考文献 [?] 中, 作者给出了卡特兰数的 66 种组合证明。事实上, 二叉树与平面树之间存在着一个自然、著名的映射, 我们这里略作描述。

由平面树到二叉树的映射, 被称作平面树的“分层”, 定义如下: 对以任意具有 $n+1$ 个结点的平面树, 去掉其根, 对于每个非根结点 s , 将原树中相邻右边的兄弟作为新树中结点 s 的右孩子, 而将 s 的最左边的孩子作为新树中结点 s 的左孩子。例如, 我们有如下对应:

4 0-1-2 递增树、交错排列与 Euler 数

称一棵递增标号树为 0-1-2 递增树 (0-1-2 increasing tree), 若此树的孩子个数不超过 2. 对于此类特殊的递增标号树, 我们有如下结论:

定理 10. $[n]$ 上的 0-1-2 递增树与 $[n]$ 上的交错排列同分布。

证明 (代数证明) 根据二者的性质, 我们分别计算它们的指数型生成函数 $Alt(x) = \sum_{n=0}^{\infty} |Alt_n| \frac{x^n}{n!}$, $IT(x) = \sum_{n=0}^{\infty} |IT_n| \frac{x^n}{n!}$.

首先考虑 $Alt(x)$, 若记 $a_n = |Alt_n|$, $a(x) = Alt(x)$, 则可以证明 $a_n (n \neq 1)$ 具有如下递推关系:

$$\begin{aligned} a_{2n+1} &= \sum_{i=1}^n \binom{2n}{2i-1} a_{2i-1} a_{2n-2i+1}; \\ a_{2n+1} &= \sum_{i=1}^n \binom{2n}{2i} a_{2i} a_{2n-2i}; \\ a_{2n} &= \sum_{i=1}^{n-1} \binom{2n-1}{2i} a_{2i} a_{2n-2i-1} \end{aligned}$$

事实上,注意到当 j 为奇数时,集合 $|\{\sigma \in Alt_n, \sigma(j) = 1\}|$ 的个数为 $\binom{n-1}{j-1}a_{j-1}a_{n-j}$; 而当 j 为偶数时, 个数为 0. 因此对 a_n 分类计数, 即可得后面的两个式子; 类似地分析集合 $|\{\sigma \in Alt_n, \sigma(j) = n\}|$, 可得第一个式子。将前两个式子相加, 将后一个式子乘以 2, 则得到一个统一的递推关系如下:

$$2a_{n+1} = \sum_{i=0}^n \binom{n}{k} a_k a_{n-k} (n \neq 1).$$

因此,

$$\begin{aligned} 2a'(x) &= 2 + \sum_{n=1}^{\infty} 2a_{n+1} \frac{x^n}{n!} \\ &= 2 + \sum_{n=1}^{\infty} \sum_{i=0}^n \binom{n}{k} a_k a_{n-k} \frac{x^n}{n!} \\ &= 1 + \sum_{n=0}^{\infty} \sum_{i=0}^n \binom{n}{k} a_k a_{n-k} \frac{x^n}{n!} \\ &= 1 + a^2(x) \end{aligned}$$

即 $a(x)$ 满足微分方程 $2a' = 1 + a^2$ 和初值条件 $a(0) = 1$, 由基本的微分方程的知识, 解此微分方程, 可得 $a(x) = \tan(x) + \sec(x)$.

将 $2a' = 1 + a^2$ 两边对 x 求导, 得

$$2a'' = 2a \cdot a'. \quad (1)$$

(1)两边展开成形式幂级数, 再取两边 $\frac{x^n}{n!}$ 的系数, 得 a_n 的另一个递推关系。

$$a_{n+2} = \sum_{k=0}^n \binom{n}{k} a_{k+1} a_{n-k} \quad (2)$$

接下来考虑 $0-1-2$ 递增树, 我们记 b_n 为 $[n]$ 上的 $0-1-2$ 递增树的个数, 我们只需证明 b_n 也具有相同的递推关系(2) 和初值条件。初值条件是平凡的, 下面考虑其递推关系。

对于任意一棵 $[n+2]$ 上的 $0-1-2$ 树 T , 考虑根的孩子。由于根至多只有两个孩子, 去掉 T 中与根相邻接的两条边, 树 T 被分解成了两棵树 (可能有一棵为空树)。我们按照包含结点 $n+2$ 的树的大小进行分类, 此树可能包含结点由 1 到 $n+1$. 若包含的结点个数为 $k+1$ 个, 选择 $n+2$ 以外的 k 个标号的方案数为 $\binom{n}{k}$, 将其构成一棵 $0-1-2$ 递增树的方案数为 b_{k+1} , 而另外一棵树的方案树为 b_{n-k} . 故有 $a_{n+2} = \sum_{k=0}^n \binom{n}{k} a_{k+1} a_{n-k}$.

至此, 我们证明了二者的同分布性, 同时, 给出了其生成函数, □

5 Parking Function

5.1 基本概念

Parking Function 的定义首先是由 *Konheim* 和 *Weiss* 在他们研究计算机科学中的一个分配问题中提出的。

几乎所有的信息归档、分类和存储系统中都会出现如下的结构: 每个记录或者信息单元都具有一个自然命名和记录辨识命名. 所有可能的命名, 记为 $\{a_1, a_2, \dots, a_m\}$, 比我们常见的数目大很多. r 记为这个大数, $\{a_{i_1}, a_{i_2}, \dots, a_{i_r}\}$ 记为将要存储的某个文件的数据. 存储的过程包括一个重要步骤: 分配每个 a_{i_k} 一个唯一的记录其位置的数 $A_{i_k} \in \{0, 1, \dots, n-1\}$, 其中 n 是内存的大小. 当然, r 远小于 n . 于是一个问题便出现了: 如何设置一个过程来分配位置数的记录使得在只知道这个记录的命名的前提下, 存储和恢复这个记录的时间最短.

这个问题有一个更为简要和生动的描述. 有一个单行的有 n 个停车位的停车场. 停车位从第一个位置开始依次按 $1, 2, \dots, n$ 来命名. 在实际生活中, 每辆车都是有最想要的停车位, 我们将他们停车的偏好按照位置编号构成一个序列. 如 $n = 3$ 时, $(1, 1, 2)$ 便是一个偏好序列, 表明第一、二辆车想停一号位, 第三辆车想停二号位. 当然, 这条单行道非常狭窄, 驶入的车无法掉头回走, 每辆车驶入后都首先寻找它的偏好停车位, 如果这个位置是空的, 停在那里; 否则, 它将继续往前开直到发现第一个空置的停车位并泊车. 我们称一个序列是一个 *Parking Function* 如果它能让所有的车完成泊车. 例如 $(1, 1, 2)$ 便是一个 *Parking Function*, 然而 $(2, 3, 2)$ 不是. 对于 $(1, 1, 2)$, 第一辆车按偏好驶入一号位, 第二辆车经过一号位后发现被占据只能前进一步驶入二号位, 第三辆车经过二号位后只能驶入三号位, 完成泊车. 然而对 $(2, 3, 2)$, 在第一辆车和第二辆车依次占据了二号位和三号位之后, 第三辆车经过其偏好位置二号位, 却因为二号位被占据只能前往三号位, 但是三号位已被第二辆车占据, 导致第三辆车无法泊车.

5.2 简单结论

在看了上面的例子后, 很自然产生一个问题, 怎么判断一个序列是否是 *Parking Function* 呢? 对排列和逆序数感兴趣的读者不难想出下面的判别条件.

定理 11. 设 $\alpha = (a_1, a_2, \dots, a_n) \in \mathbb{P}^n$, $b_1 \leq b_2 \leq \dots \leq b_n$ 是 α 的按照递增 (允许相等) 顺序的重新排列. 则 α 是 *Parking Function* 当且仅当 $b_i \leq i$.

证明 假设某个 $b_i > i$, 那么至少有 $n-i+1$ 辆车偏好 $n-i$ 个位置 $i+1, i+2, \dots, n$ 于是便不能完成停车. 于是条件是必要的. 充分性可以通过对 n 进行归纳来证明. $n = 1$ 时显然成立. 可设 $a_1 = k$. 设对 $1 \leq i \leq n-1$,

$$a'_i = \begin{cases} a_{i+1} & \text{若 } a_{i+1} \leq k \\ a_{i+1} - 1 & \text{若 } a_{i+1} > k. \end{cases}$$

于是序列 $(a'_1, a'_2, \dots, a'_{n-1})$ 满足归纳中的作为 *Parking Function* 的条件. 这就意味着第二辆车及其后所有的车可在第一辆车占据了 k 位之后仍然完成泊车. 命题得证.

回到前面的例子. 对于 $(1, 1, 2)$, 它本身便是一个非递减的排列, 而且每个位置 i 上的 a_i 满足 $a_i \leq i$, 于是便是一个 *Parking Function*. 然而对 $(2, 3, 2)$, 它的非递减的排列是 $(2, 2, 3)$, 第一个位置 $a_1 = 2 > 1$, 不符合定理 1 的条件, 便不是 *Parking Function*. 给出了 *Parking Function* 的判别条件之后, 很自然就会

产生一个疑问: n 长的序列中到底有多少个 *Parking Function* 呢? 下面的定理回答了这个问题.

定理 12. (*Konheim and Weiss, 1966*) 令 $f(n)$ 是长为 n 的 *Parking Function*. 则有

$$f(n) = (n+1)^{n-1}.$$

证明 (*Pollak, c.1974*). 增加一个额外的停车位 $n+1$, 然后把 1 与 $n+1$ 相连并按原来顺序, 将停车位重组成一个圆. 允许 $n+1$ 作为一个偏好位置. 这样一来所有的车便都能找到位置泊车. α 是一个 *Parking Function* 当且仅当空车位是 $n+1$. 若 $\alpha = (a_1, a_2, \dots, a_n)$ 使第 i 辆车停在位置 p_i , 那么 $(a_1 + j, a_2 + j, \dots, a_n + j) \pmod{n+1}$ 将会导致第 i 辆车停在 $p_i + j$. 于是仅有一个向量

$$(a_1 + i, a_2 + i, \dots, a_n + i) \pmod{n+1} \quad i = 0, 1, \dots, n$$

于是有

$$f(n) = \frac{(n+1)^n}{n+1} = (n+1)^{n-1}.$$

上述的证明简洁优美, 其结果也许会引起读者的兴趣: $[n]$ 上的 *Parking Function* 的数目为什么恰恰跟 $[n]$ 上的有根森林数目相同? 下面我们将要说明它们有着相同的递推关系.

记 $P(i)$ 为 $[n]$ 上的 *Parking Function* 的数目. 特别地, 我们即将带有“组合意味”地解释下面的递推关系

$$P(n+1) = \sum_{i=0}^n \binom{n}{i} (i+1)P(i)P(n-i),$$

分别适用于 $[n+1]$ 上的有根森林和 $[n+1]$ 上的 *Parking Function*. 虽然可以用纯粹代数方法来证明递推关系, 但是这并不能让我们获得关键的组合观点.

首先, 令 F 是 $[n+1]$ 上的有根森林, F_1 是 F 中含顶点 1 的连通分支. 设 F_1 含有 $i+1$ 个顶点. 于是我们有 $\binom{n}{i}$ 种方法选择除去 1 以外的 i 个顶点, 同时有 $P(i)$ 种方法确定含 $i+1$ 个顶点的树. 一旦完成上述过程, $i+1$ 个顶点的任何一个都可以作为 F_1 的根. 最后, 我们有 $P(n-i)$ 种方法确定剩下 $n-i$ 个顶点组成的有根森林. 因此, 根据乘法原理, 在 $[n+1]$ 个顶点上的确有 $\binom{n}{i}(i+1)P(i)P(n-i)$ 个有根森林, 其中顶点 1 是在顶点数目为 $i+1$ 的连通分支中. 对所有的 i 进行加和, 我们便得到了上述递推关系.

接下来设 f 是 $[n+1]$ 上的 *Parking Function*. 设最后一辆车驶入时, 停车位 $i+1$ 是空的, 其中 $i \in [0, n]$. 这就意味着存在集合 S 由 i 辆车组成, 这些车停在前 i 个停车位上, 而且它们的偏好序列构成了一个 $[i]$ 上的 *Parking Function*. 如果我们对后面的 $n-i$ 辆车的车位号减去 $i+1$, 则它们自动构成一个 $[n-i]$ 上的 *Parking Function* (注意到这 $n-i$ 辆车不可能有在 $[i+1]$ 中的数作为偏好车位, 否则车位 $i+1$ 在最后就不会空下来). 最后, 因为 f 构成一个 *Parking Function*, 有 $f(n+1) \leq i+1$. 也就是说, 对 $f(n+1)$ 有 $i+1$ 种选择. 考虑到 S 有 $\binom{n}{i}$ 种可能性, 根据乘法原理和对 i 求和就证明了递推关系. 具有相同的递归关系, 说明 *Parking Function* 和有根森林这两个组合结构有着内在的联系.

Parking Function 一经提出便激起了大量数学家的研究兴趣. *Foata* 和 *Riordan* 以及 *Francon* 构造了从 *Parking Function* 到 $[n]$ 上的无圈函数的集合的映射; *Kreweras* 研究了 *Parking Function* 和标号树的生成函数共同满足的递推关系; *Stanley* 和 *Pitman* 解释了 *Parking Function* 和超平面排列、*intervalorders*、非交分拆、*polytope* 之间的联系. 最近 *Parking Function* 又成了概率学家和统计学家的新宠.

树的计数是计数组合学中非常经典、非常精彩的一节, 由于树紧凑的递归结构, 生成函数这一强大的工具得以使用, 这里我们仅仅选了一些极其经典的结论, 以求一管以窥全豹.

关于树的一些组合双射以及树与其它组合结构之间的映射使得树上的研究方法可以应用到格路、字、排列等等; 同样, 它们的研究技巧同样可以用来处理一些树的问题. 例如: 我们前面讲到的递增树与排列的对应, 以及后面将会涉及到的平面树与迪克路的对应等等. 这些研究手段一定程度上体现了组合数学的一个特点, 即研究一些看似无关的物体之间的内在联系, 从不同的角度认识组合物体.