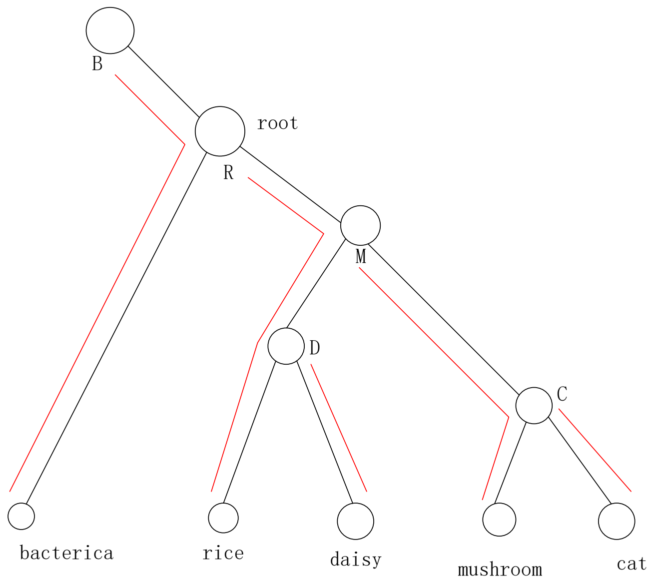
Tree Counting Note

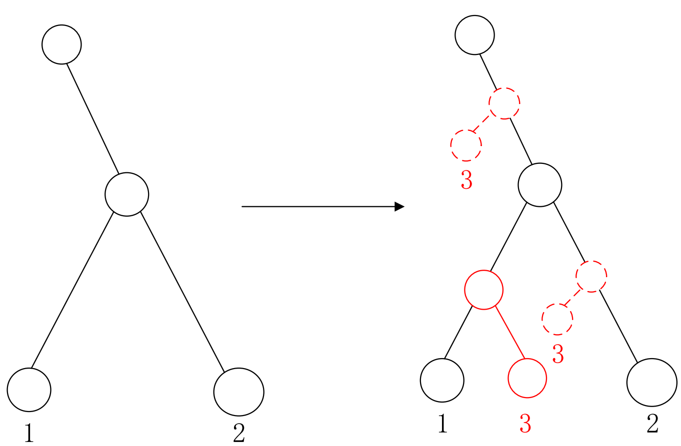
徐逸凡 516071910074

1 rooted binary trees with leaves labeled by $\{1, 2, \ldots, n\}$

考虑下面一张生物进化的树状图（一颗满二叉树full binary tree），叶子结点是现存的物种，一共有n个。如果添加一个结点，使根结点成为该结点的右儿子。可以观察出内点（interior nodes）数量也是n 个。这可以用归纳法证明，每当一个叶结点分裂出两个儿子，叶结点加1，内点加1，叶结点数量和内点数量相等。另一种方法是可以建立叶结点和内点的一一映射。将叶结点A沿着父节点向上，使得它第一次作为某个节点B的右儿子，将A和B对应起来。这个映射的逆映射也是存在的，任一内点，先找它的右儿子，然后一直往左找，直到找到叶结点。这说明该映射是一个双射，所以叶结点数量和内点数量相等。



我们将所有有n个编号的叶结点的**rooted binary tree**的集合记为T_n^l。现考虑$|T_n^l|$ 的大小。这里可以采用归纳法。



考虑从T_n^l到T_{n+1}^l，如果一棵树已经有n个叶结点，被标记为$1, 2, \ldots, n$，并且有$2n-1$条边（因为叶结点n个，内点n个，一共$2n$个结点）。现在要添加

第n+1 个叶结点到树上，有$2n$−1种添法（即将第n+1个叶结点挂到$2n$−1条边中的任意一条），形成一颗有$n+1$个叶结点的树。所以得到：

$$|T_{n+1}^l| = |T_n^l|(2n-1)$$

递推得到

$$|T_n^l| = (2n-3)!!$$

2 ranked rooted binary trees with n leaves labeled by $\{1, 2, \ldots, n\}$

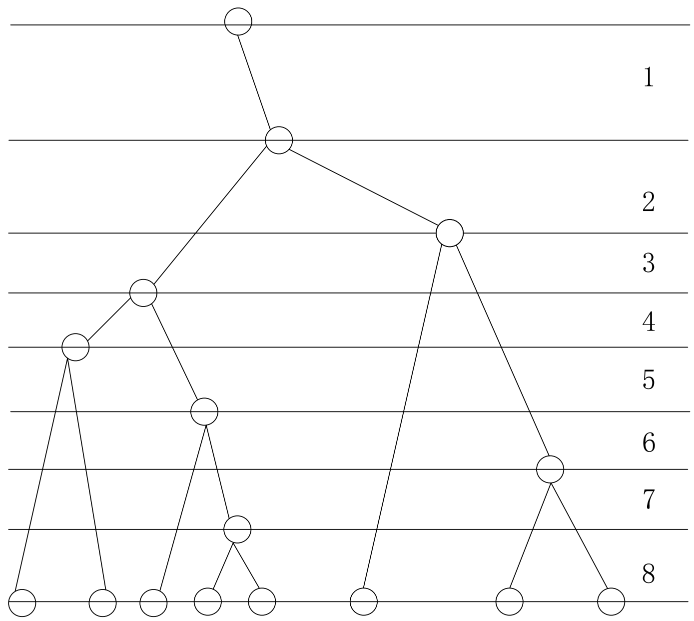
在一棵树中，内点具有偏序关系，现在我们要使这棵树**ranked**，就是将内点的偏序关系全序化。我们把所有有n个编号的叶结点的**ranked rooted binary tree**的集合记为R_n^l。类似地，我们考虑从R_n^l到R_{n+1}^l。现在一颗内点全序化的树有n个叶结点，现要添加第$n+1$个叶结点。经过每一个内点做一条水平线，这些水平线将边划分为$1+2+\cdots+n=\binom{n+1}{2}$条线段，在每条线段上都可以挂第$n+1$个叶结点。所以得到递推关系

$$R_{n+1}^l = \binom{n+1}{2} R_n^l$$

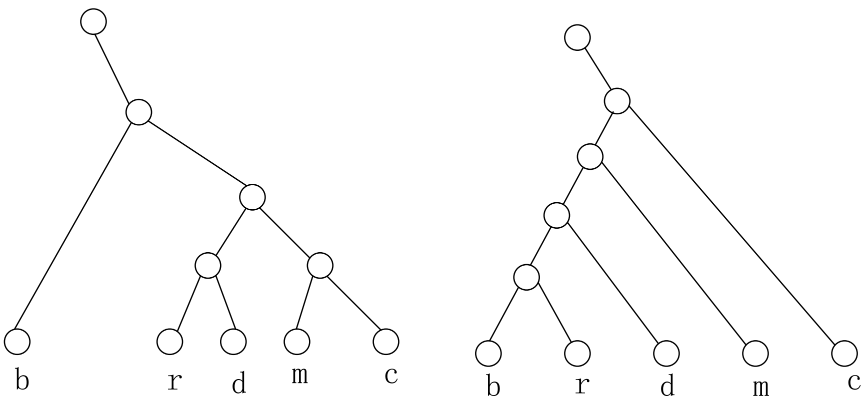
进而推出

$$R_{n+1}^l = \binom{n+1}{2}\binom{n}{2}\cdots\binom{2}{2} = \frac{n!(n+1)!}{2^n}$$

另一种看法也可以得到上式。从叶结点开始，每次选取2个结点合并成1个，直到最后只剩下1 个根节点，就构建出一棵叶结点编号的**ranked rooted binary tree**。



现在我们来看下面两棵生物进化树（**unranked**）出现的可能性。从直观上来看，(a)比(b)出现的可能性要大，因为(a)中的内点是偏序关系，还未完全确定任意两个内点之间的时间先后顺序（非父子关系的内点），而(b)中的内点已经是一个全序关系了。假设每一个可能出现的全序关系出现的可能性都是相等的（有些全序关系不可能出现，比如内点A表示猿分成人和猴两类，内点B 表示猴分成猩猩和狒狒两类，那么结点B在结点A之前的全序关系是不可能的。假设每种可能的全序关系出现的可能性相等）。计算将(a) 的内点全序化（**ranked**）的种数，除以$|R_n^l|$ 就得到(a) 出现的可能性。



(a) (b)

设λ_v是以内点v为根的子树的叶结点的个数减1，也就是以内点v为根的子树的内点个数。不考虑现有偏序关系，全序化有$(n-1)!$种。考虑现有的树上的偏序关系，设v是任意内点，事件E_v为v是全序化后以其为根的子树的内点中的最小元素，$P(E_v)=\frac{1}{\lambda_v}$。那么全序关系与现有的偏序关系相容的概率为

$$P\Big(\cap_{v\in\dot{V}(T)} E_v\Big)=\prod_{v\in\dot{V}(T)} P(E_v)=\prod_{v\in\dot{V}(T)}\frac{1}{\lambda_v}$$

$\dot{V}(T)$是T的内点集，不包括根节点的父节点。以上等式成立需要说明所有E_v是collectively independent（此计算可以类比递增序列出现的概率是每个元素都是从它以后所有元素中的最大值的概率的乘积）。所以与现有偏序关系相容的全序关系个数，即固定形态的树T 区分内点的rank 后的个数$|T|$为

$$|T|=\frac{(n-1)!}{\prod_{v\in\dot{V}(T)}\frac{1}{\lambda_v}}$$

T出现的概率为

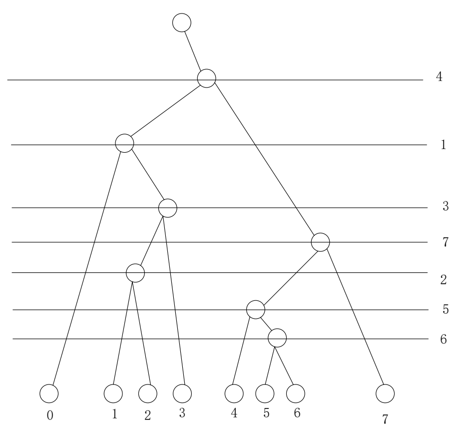
$$P(T)=\frac{|T|}{|B_n^l|}=\frac{2^{n-1}}{n!\prod_{v\in\dot{V}(T)}\frac{1}{\lambda_v}}$$

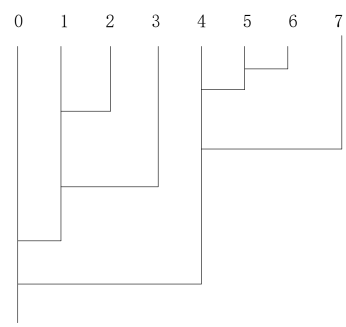
另一种建树的模式是叶结点所及分裂，形成T的形态后，再给每个叶结点加label，即概率乘上$n!$。可以证明，两种方式算出来是一样的。

另外，关于最大性独立，对于一维序列和树来说是成立的。但是，在$\mathbf{N}^2, \mathbf{N}^3$不一定成立。在$\mathbf{N}^2$上定义关系$(x_1, y_1) \leq (x_2, y_2)$当且仅当$x_1 \leq x_2, y_1 \leq y_2$。这是个偏序关系。在二维平面的第一象限，在每个整点A上放置这样一个有序对，记为$f(A)$。一个整点A取到最大值是说，对任意满足$A \leq B$的B都有$f(A) \leq f(B)$。设事件E_A 为A 取到最大值，事件E_B为B取到最大值。当$A \leq B$或$B \leq A$时，E_A, E_B是独立的。当A, B无法比较大小时，E_A和E_B是不独立的。（样本空间为一个偏序集，两个点取到最大值的概率是独立的当且仅当这两个点具有偏序关系）。

3 rooted ranked oriented binary tree

定义R_n^o为有n个叶结点的**rooted ranked oriented binary tree**。根据之前所讲的叶结点与内点的一一对应（叶结点往上，找第一次作为右儿子的父结点），我们作出下面两幅图。





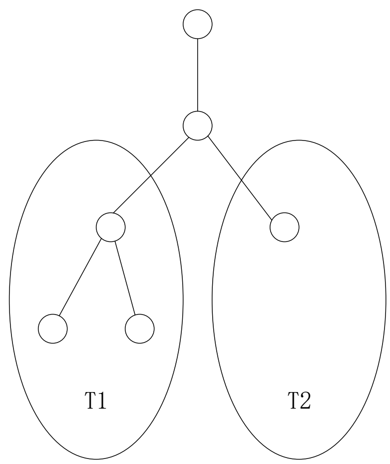
左右两幅图是一一对应的，所以

$$|R_n^o| = (n-1)!$$

我们还可以通过$|R_n^o|$得到$|R_n^l|$，对$|R_n^o|$除以2^{n-1}消掉$n-1$个内点左右儿子的

次序，然后乘以$n!$，即为n个叶结点标号，得到

$$|R_n^l| = \frac{(n-1)!n!}{2^{n-1}}$$



如图，T_1, T_2是根的左子树和右子树，刚开始时T_1, T_2均只有1个结点。然后叶结点随机分裂。$P_n(k)$是指出现T_1有k个顶点，T_2有$n-k$个顶点的概率，其中$1 \le k \le n-1$。现对n用归纳法证明$P_n(k) = \frac{1}{n}$。

1. $n = 3$时，$P_3(1) = \frac{1}{2}, P_3(2) = \frac{1}{2}$。

2. 若$1 \le k \le n-1$时，$P_n(k) = \frac{1}{n}$成立。现要证明$\forall 1 \le kn$，$P_{n+1}(k) = \frac{1}{n}$。
   当$1 \le k \le n-1$时：

   $$P_{n+1}(k) = P_n(k-1)\frac{k-1}{n} + P_n(k)\frac{n-k}{n} = \frac{1}{n-1}\frac{k-1}{n} + \frac{1}{n-1}\frac{n-k}{n} = \frac{1}{n}$$

   当$k = n$时：

   $$P_{n+1}(n) = P_n(n-1)\frac{n-1}{n} = \frac{1}{n-1}\frac{n-1}{n} = \frac{1}{n}$$

   所以$n+1$时结论也成立。

由归纳法可知，$\forall 1 \le k \le n-1, P_n(k) = \frac{1}{n}$。

$\sigma \in S_n$，求$P(|1, \sigma(1), \ldots, \sigma^2(1)| = k)$。

$$P(|1, \sigma(1), \ldots, \sigma^2(1)| = k) = \frac{\binom{n-1}{k-1}\frac{k!}{k}(n-k)!}{n!} = \frac{1}{n}$$