树的枚举与树算法复杂性计算*

王 振 宇

(武汉工学工程研究所，武汉 430074)

关键词 树枚举、算法复杂性分析、简单生成树族

树结构上算法复杂性分析近几年来得到越来越多的注意. Flajolet[1] 研究了树上递归下降算法的复杂性分析，办法是为一个形式化的树算法描述语言中的各种构造建立一个复杂性度量表. 平行于树上的可加枚举问题[2]，本文作者在文献[3]中引入了所谓"树结构上可加复杂性算法"，得到过一些本质上只能用于递归算法分析的结果.

本文将展示一种新方法. 在此框架中，可加枚举问题及其一般求解格局将被直接用于树结构上算法的复杂性分析(递归的或非递归的). 这时，算法复杂性度量变成执行所考虑的算法过程中某些操作的执行次数. 进而在某些树族的树结构上算法执行中每个操作的执行次数又变成树上某个属性的枚举问题.

一、树上的广义可加枚举问题

令 F 记一树族，F_n 是它的由 n 个结点的、结构不同的树所构成的子族，且记 $y_n = |F_n|$. $\{y_n\}$ 的生成函数 $Y(x)$ 定义为

$$Y(x) = \sum_{n=0}^{\infty} y_n x^n. \tag{1}$$

以 E 记某树族 F 中树上的一个枚举问题. E 在树 T 上的枚举结果记为 $E(T)$，且

$$E_n = \sum_{T \in F_n} E(T), \quad E(x) = \sum_{n=0}^{\infty} E_n x^n. \tag{2}$$

F 可以是二叉树族 B，t 叉树族 C 和有序树族 O 之一.

按照文献[4]的定义，一个树族被称为简单生成树族，如果它的生成函数 Y 满足如下关系 ($\Phi(Y)$ 是 Y 的一个幂级数，具有非负实系数):

$$Y = \rho + x \cdot \Phi(Y), \quad \Phi(Y) = \sum_{i=0}^{\infty} \phi_i Y^i. \tag{3}$$

显然，"简单生成树族"覆盖二叉树，t 叉树和有序树族.

定义 令 E 是树族 F 上的一个枚举问题. 称 E 是一个广义可加枚举问题 (以下记为 GAEP)，如果存在一个实常数 c，使得对于每个树 $T \in F$，有

$$E(T) = c \cdot \sum_{i=1}^{m} E(T_i) + e(T), \tag{4}$$

其中 m 为 T 的子树的个数，$T_1, \cdots, T_m$ 是 T 的所有子树，e 是 F 上一个已求解的枚举问题，$e(T)$ 是 e 在 T 上的枚举结果.

在此定义中，e 表现为对可加性的某种修正，通常是一个较 E 简单的枚举问题. 明显，定义中的 GAEP 是作者在文献[2]引进的可加枚举问题的推广 ($c = 1$).

对于定义中的"另一个枚举问题" e，我们使用一组与 E 类似的记号: e_n, $e(x)$. 枚举问题

本文 1991 年 5 月 14 日收到. 1992 年 1 月 11 日收到修改稿.

* 国家自然科学基金资助项目.

的求解就是从 $e(x)$ 求出 $E(x)$ 或从 $\{e_n\}$ 求出 $\{E_n\}$。

定理 1 对于简单生成树族 F 上的广义可加枚举问题 E，有

$$E(x)=\frac{e(x)\cdot Y'}{Y'-cY'+c\Phi(Y)}, \tag{5}$$

其中 Y 和 $\Phi(Y)$ 由式(1)和式(3)所定义。对于二叉树族，式(3)中的 $\rho=1,\Phi(Y)=Y^2$，且 $Y(x)=B(x)=(1-\sqrt{1-4x})/2x$，定理 1 成为：

定理 2 如果 E 是二叉树族 B 上的一个 GAEP，则 E 的生成函数 $E(x)$ 由下式给出：

$$E(x)=\frac{e(x)}{1-2\cdot c\cdot xB(x)}, \tag{6}$$

特别地，对于 $c=1$，

$$E(x)=\frac{e(x)}{\sqrt{1-4x}}=\frac{xB'(x)}{B(x)-1}\cdot e(x)=\frac{B'(x)}{B(x)^2}\cdot e(x), \tag{7}$$

而当 $c=1/2$ 时，

$$E(x)=\frac{e(x)}{1-xB(x)}=B(x)\cdot e(x). \tag{8}$$

在大多数应用中，$e(x)$ 是形为 $e(x)=a_1f_1(x)+\cdots+a_mf_m(x)$ 的线性组合，这时相应的 $E(x)$ 必然是形为 $E(x)=a_1G(x)\cdot f_1(x)+\cdots+a_mG(x)\cdot f_m(x)$ 的线性组合。如果对给定的 $f_i(x)$ 我们能通过查表得到相应的 $G(x)\cdot f_i(x)$，那将是很诱人的。在大多数应用中，$f_i(x)$ 是下列表达式：

$$1,x,B(x),xB(x),\frac{xB'(x)}{B(x)^2}=\frac{x}{\sqrt{1-4x}},$$

可对 $c=1$（可加枚举问题）和 $c=1/2$ 的情况依式(7)或(8)分别构造 e-E 映射表 1 和表 2。

二、应用例子

Brinck 和 Foo 在文献[5]中分析了二叉树上的穿线搜索算法。令 E_n 为在 B_n 中所有二叉树上所考虑算法的执行中用于某一操作的全部时间。分析指出，要用到的所有复杂性度量递推关系可分为三类：

第一类：
$$E_n=2\cdot\sum_{i=0}^{n-1}b_iE_{n-i-1}+e_n, \tag{9}$$

第二类：
$$E_n=\sum_{i=0}^{n-1}b_iE_{n-i-1}+e_n, \tag{10}$$

第三类：
$$E_n=\sum_{i=0}^{n-1}b_iE_{n-i-1}+E_{n-1}+e_n. \tag{11}$$

$$E(x)=2xB(x)E(x)+e(x); \tag{9'}$$

$$E(x)=xB(x)E(x)+e(x); \tag{10'}$$

$$E(x)=xB(x)E(x)+xE(x)+e(x). \tag{11'}$$

如果 E 是一个 GAEP，且 $c=1$，即 E 是一个可加枚举问题，则它属于式(9)或(9')类。在这种情况下，e-E 映射表 1 对于复杂性度量或分析的计算，即从 $e(x)$ 计算 $E(x)$，或从 $\{e_n\}$ 计算 $\{E_n\}$，是适用的。

对于第二类递推关系(10)或(10')，我们可以验证，如果 E 是一个 GAEP，且 $c=1/2$，则式(10)成立，且在复杂性度量或分析的计算中，e-E 映射表 2 是适用的。

表1 第一类 ($c=1$ 时) GAEP 的 e-E 映射

e_n	$e(x)$	$E(x)$	E_n
I $\begin{cases}1, & n=i\\0, & n\neq i\end{cases}$	x^i	$\frac{x^i}{\sqrt{1-4x}}$	$\binom{2n-2i}{n-i}$
II $b_n, n\geqslant 0$	$B(x)$	$\frac{1}{2x}\left(\frac{1}{\sqrt{1-4x}}-1\right)$	$\frac{2n+1}{n+1}\binom{2n}{n}$
III $\begin{cases}0, & n=0\\b_{n-1}, & n\geqslant 1\end{cases}$	$xB(x)$	$\frac{1}{2}\left(\frac{1}{\sqrt{1-4x}}-1\right)$	$\frac{2n-1}{n}\binom{2n-2}{n-1}$
IV $b_{n+1}, n\geqslant 0$	$\frac{B(x)-1}{x}$	$\frac{1-2x-\sqrt{1-4x}}{2x\sqrt{1-4x}}$	$\frac{1}{2}\binom{2n+4}{n+2}-\binom{2n+2}{n+1}$
V $b_{n+2}, n\geqslant 0$	$\frac{B(x)-1-x}{x}$	$\frac{1-2x-2x^2-\sqrt{1-4x}}{2x\sqrt{1-4x}}$	$\frac{1}{2}\binom{2n+6}{n+3}-\binom{2n+4}{n+2}-\binom{2n+2}{n+1}$
VI $n\cdot b_{n-1}, n\geqslant 0$	$\frac{xB'(x)}{B(x)^2}$	$\frac{x}{1-4x}$	4^{n-1}

对于第三类递推关系(11)或(11′),容易验证关系

$$E(x)=\frac{e(x)}{1-x-xB(x)} \quad 和 \quad \frac{1}{1-x-xB(x)}=B(x)^2.$$

定理 3 如果二叉树上的枚举问题 E 的 E_n 满足式(11),则其生成函数如下式:

$$E(x)=B(x)^2\cdot e(x). \tag{12}$$

我们可以为式(11)建立一张类似的 e-E 映射表(表 3).

Brinck 和 Foo 在他们的文章中定义了所谓"左壳结点"和"右壳结点". 一棵二叉树上的左壳结点是在该树上只通过自根结点开始的左枝周游到达的结点. 联结两个左壳结点的边称为左壳边. 类似地,可定义右壳结点和右壳边.

Brinck-Foo 命题 1 B_n 中所有二叉树上的左壳结点总数由下式给出:

$$q_n=b_{n+1}-b_n. \tag{13}$$

事实上, Brinck-Foo 得到了如下的递推关系:

$$q_0=0,\quad q_n=\sum_{i=1}^{n-1}q_ib_{n-i-1}+b_n,\quad n\geqslant 1. \tag{14}$$

这是一个第二类枚举问题,式(14)右端的最末项是表 2 中 e_n 列的情况 (III) 和 (I) 的线性组合: (III)-(I). 通过查表 2 中 E_n 列的相应值,可立即得到式(13).

Brinck-Foo 命题 2 在 B_n 中所有二叉树上,位于自根结点到按前序周游的最末结点的通路上的结点总数由下式给出:

$$p_n=b_{n+2}-2\cdot b_{n+1}. \tag{15}$$

这时,递推关系成为

$$p_0=0,\quad p_n=\sum_{i=0}^{n-1}b_ip_{n-i-1}+p_{n-1}+b_n,\quad n\geqslant 1. \tag{16}$$

所以,这是一个第三类枚举问题,式(16)右端的最末项是表 3 中 e_n 列的情况 (III) 和 (I)的线性组合: (III)-(I).通过查表 3 中 E_n 列的相应值,就可立即得到$p_n=b_{n+2}-b_{n+1}-b_{n+1}=b_{n+2}-2b_{n+1}$, 这就证明了式(15).

Brinck-Foo 命题 3 分析下列二叉树中序查找中计算后继结点的算法

表 2　第二类($c = 1/2$ 时) GAEP 的 e-E 映射

e_n	$e(x)$	$E(x)$	E_n
I $\begin{cases}1, & n=0\\ 0, & n>0\end{cases}$	1	$B(x)$	$b_n,\ n\geqslant 0$
II $\begin{cases}1, & n=1\\ 0, & n\neq 1\end{cases}$	x	$xB(x)$	$\begin{cases}0, & n=0\\ b_{n-1}, & n>0\end{cases}$
III $b_n,\ n\geqslant 0$	$B(x)$	$B(x)^2$	$b_{n+1},\ n\geqslant 0$
IV $\begin{cases}0, & n=0\\ b_{n-1}, & n>0\end{cases}$	$xB(x)$	$xB(x)^2 = B(x) - 1$	$\begin{cases}0, & n=0\\ b_n, & n>0\end{cases}$
V $b_{n+1},\ n\geqslant 0$	$\dfrac{B(x)-1}{x}$	$B(x)^3$	$b_{n+2} - b_{n+1}, n\geqslant 0$

表 3　第三类 GAEP e-E 映射

e_n	$e(x)$	$E(x)$	E_n
I $\begin{cases}1, & n=0\\ 0, & n>0\end{cases}$	1	$B(x)^2$	$b_{n+1},\ n\geqslant 0$
II $\begin{cases}1, & n=1\\ 0, & n\neq 1\end{cases}$	x	$xB(x)^2$	$\begin{cases}0, & n=0\\ b_n, & n>0\end{cases}$
III $b_n,\ n\geqslant 0$	$B(x)$	$B(x)^3$	$b_{n+2} - b_{n+1}, n\geqslant 0$

```
procedure insucc(p)
    q ← p.rhs
    if p.rtag = branch then while q.ltag = branch do q ← q.lhs
    return
```

Brinck 和 Foo 指出：分析此算法的关键是以 B_n 中的所有二叉树的所有结点为输入集合时进行测试 q.ltag = branch 的总次数。这种次数满足的递推关系是

$$k_0 = 0,\ k_n = 2 \cdot \sum_{i=0}^{n-1} k_i b_{n-i-1} + b_{n+1} - 2b_n. \tag{17}$$

显然，这是一个第一类(可加)枚举问题，式(17)右端的最末两部分是表 1 中 e_n 列的情况(IV) ($i = 0$)，(II) 和 (I) 的线性组合：(IV) $-$ 2 $\cdot$ (II) $+$ (I)，通过查表 1 中 E_n 列的相应值，就可立即得到我们所要的结果：

$$k_n = \frac{1}{2}\binom{2n+4}{n+2} - \binom{2n+2}{n+1} - \binom{2n+2}{n+1} + \binom{2n}{n} = \frac{n-1}{n+2} n b_n.$$

如同文献[1]所做，上述定理同 Darboux 渐进定理相结合也可用于分析树算法复杂性的渐进特性。

参　考　文　献

[1] Flajolet, Ph., Steyeart, J. M., *Math. System Theory*, 19(1987), 301—331.
[2] 王振宇，数学物理学报，4(1984)，31—38.
[3] 王振宇，武汉大学学报(计算机软件工程专刊)，1988，57—63.
[4] Meir, A., Moon, J. W., *Can. J. of Math.*, 30(1978), 997—1015.
[5] Brinck, K., Foo, N. Y., *The Computer J.*, 24(1981), 148—155.