A Refinement of Cayley's Formula for Trees

Ira M. Gessel *
Department of Mathematics
Brandeis University
Waltham, MA 02454-9110
gessel@brandeis.edu

Seunghyun Seo
Department of Mathematics
Seoul National University
Seoul 151-742, Korea
shyunseo@gmail.com

Submitted: Jun 30, 2005; Accepted: Jan 29, 2006; Published: Feb 8, 2006
Mathematics Subject Classification: 05A15

Dedicated to Richard Stanley on the occasion of his 60th birthday

Abstract

A *proper vertex* of a rooted tree with totally ordered vertices is a vertex that is the smallest of all its descendants. We count several kinds of labeled rooted trees and forests by the number of proper vertices. Our results are all expressed in terms of the polynomials

$$P_n(a,b,c) = c\prod_{i=1}^{n-1}(ia + (n-i)b + c),$$

which reduce to $(n+1)^{n-1}$ for $a = b = c = 1$.

Our study of proper vertices was motivated by Postnikov's hook length formula

$$(n+1)^{n-1} = \frac{n!}{2^n}\sum_T \prod_v \left(1 + \frac{1}{h(v)}\right),$$

where the sum is over all unlabeled binary trees T on n vertices, the product is over all vertices v of T, and $h(v)$ is the number of descendants of v (including v). Our results give analogues of Postnikov's formula for other types of trees, and we also find an interpretation of the polynomials $P_n(a,b,c)$ in terms of parking functions.

*Partially supported by NSF Grant DMS-0200596

1. Introduction

Cayley [2] showed that there are n^{n-2} (unrooted) trees on n vertices. Equivalently, there are $(n+1)^{n-1}$ forests of rooted labeled trees on n vertices. In this paper we study the homogeneous polynomials

$$P_n(a,b,c) = c\prod_{i=1}^{n-1}(ia+(n-i)b+c). \tag{1.1}$$

which reduce to $(n+1)^{n-1}$ for $a=b=c=1$. We refine Cayley's formula by showing that $P_n(a,b,c)$ counts rooted forests by the number of trees and the number of *proper vertices*, which are vertices that are less than all of their descendants other than themselves. Moreover, other evaluations of $P_n(a,b,c)$ have similar interpretations for other types of trees and forests: k-ary trees, forests of ordered trees, and forests of k-colored ordered trees (which reduce to ordered trees for $k=1$). We also give an interpretation of $P_n(a,b,c)$ in terms of parking functions.

This work grew out of a problem posed by Alexander Postnikov at the Richard Stanley 60th Birthday Conference. He gave the following identity, which he had derived indirectly [12, 13] and asked for a direct proof:

$$(n+1)^{n-1} = \frac{n!}{2^n}\sum_T\prod_v\left(1+\frac{1}{h(v)}\right), \tag{1.2}$$

where the sum is over all unlabeled (incomplete) binary trees T on n vertices, the product is over all vertices v of T, and $h(v)$ is the *hook length* of v, which is the number of descendants of v, including v.

Direct combinatorial proofs of Postnikov's identity were given by Seo [15], Du and Liu [5], and Chen and Yang [4]. Seo's approach involved counting binary trees by proper vertices in the following way: We rewrite (1.2) as

$$2^n(n+1)^{n-1} = n!\sum_T\prod_v\left(1+\frac{1}{h(v)}\right), \tag{1.3}$$

As we will see in Section 2 (see also [15]), the right side of (1.3) is equal to $\sum_B 2^{\operatorname{prop} B}$ where the sum is over all labeled binary trees B on $[n]=\{1,2,\ldots,n\}$ and $\operatorname{prop} B$ is the number of proper vertices of B. This observation led us to consider the polynomial $\sum_B t^{\operatorname{prop} B}$, which was shown in [15] to be equal to $P_n(2,t,t)$. We extended this result to count other kinds of forests and trees by proper vertices and found, surprisingly, that all of the results can be expressed in terms of the polynomials $P_n(a,b,c)$. Moreover, each type of forest or tree has a hook length formula analogous to Postnikov's. (Some of these hook length formulas have been found earlier by Du and Liu [5].)

A closely related statistic to the number of proper vertices is the number of *proper edges*, studied by Shor [17], Zeng [22], and Chen and Guo [3], where an edge $\{i,j\}$ is proper if i is the parent of j and i is less than all the descendants of j. Despite the similarity in definitions, the distribution of proper edges in trees is very different from that of proper vertices.

In Section 2, we prove some lemmas relating proper vertices of forests to hook length formulas. Sections 3 and 4 contain background information on exponential generating functions and the enumeration of trees and forests. Some properties of the polynomials $P_n(a, b, c)$, including a differential equation satisfied by its generating function, are proved in Section 5. The next three sections contain our main results on the enumeration by proper vertices of forests of rooted trees, k-ary trees, and forests of ordered trees, with corresponding hook length formulas. In Section 9 we discuss another interpretation of $P_n(a, b, c)$, due to Eğecioğlu and Remmel [6], in terms of ascents and descents in forests, and in Section 10 we give a parking function interpretation to a specialization of $P_n(a, b, c)$.

2. Hook length formulas

Throughout this paper, all trees are rooted, and all forests are forests of rooted trees. An *ordered tree* is a tree in which the children of each vertex are linearly ordered. An *ordered forest* is a forest of ordered trees in which the trees are also linearly ordered. We write $V(F)$ for the set of vertices of the forest F.

A vertex v in a tree is a *descendant* of a vertex u if u lies on the unique path from the root to v. Note that every vertex is a descendant of itself.

Let F be a forest with n vertices. Then a *labeling* of F is a bijection from $V(F)$ to $[n]$. A vertex of F is *proper* with respect to a labeling of F if its label is the smallest of the labels of all its descendants. The *hook length* $h(v)$ of a vertex v in a forest is the number of descendants of v (including v).

We will consider various kinds of trees and forests in this paper, and for each of them we will have a hook length formula something like (1.3). These hook length formulas follow from our results on counting forests by proper vertices and the following lemmas, which relates hook lengths to proper vertices.

Lemma 2.1. *Let F be a forest with n vertices, and let S be a subset of $V(F)$. Then the number of labelings of F in which every vertex in S is proper is*

$$\frac{n!}{\prod_{v\in S} h(v)}.$$

Proof. Let v be any vertex of F. We first define a bijection Θ_v from the set of labelings of F to itself with the following properties:

(i) For any vertex $u \neq v$ of F and any labeling L of F, u is proper with respect to $\Theta_v(L)$ if and only if u is proper with respect to L.

(ii) For any labeling L of F, the labelings $L, \Theta_v(L), \dots, \Theta_v^{h(v)-1}(L)$ are all different, and $\Theta_v^{h(v)}(L) = L$.

(iii) Vertex v is proper in exactly one of the labelings $L, \Theta_v(L), \dots, \Theta_v^{h(v)-1}(L)$.

To define $\Theta_v(L)$, let D be the set of labels, with respect to L, of the descendants of v. Then we define $\Theta_v(L)(v)$ to be the smallest element of D greater than $L(v)$ if such an element exists; if $L(v)$ is the largest element of D then we define $\Theta_v(L)(v)$ to be the smallest element of D. If u is a descendant of v other than v, and $L(u)$ is the ith largest element of $D - \{L(v)\}$, then we define $\Theta_v(L)(u)$ to be the ith largest element of $D - \{\Theta_v(L)(v)\}$. Thus Θ_v preserves the set D of labels of descendants of v, cycles the label of v through the elements of D, and keeps the same relative order on the labels of the other descendants of v. It is not hard to verify that properties (i), (ii), and (iii) hold.

Now we prove Lemma 2.1 by induction on $|S|$. The lemma is trivial if S is empty. It suffices to show that if $v \in S$ then the number of labelings of F in which every vertex in S is proper is $1/h(v)$ times the number of labelings in which every vertex in $S - \{v\}$ is proper. By property (i) above, the bijection Θ_v permutes the labelings of F in which every vertex in $S - \{v\}$ is proper. By (ii), every orbit has size $h(v)$, and by (iii), each orbit contains exactly one labeling in which v is proper. □

The case $S = V(F)$ of Lemma 2.1 is the well-known formula for the number of increasing labelings of a tree. (See, for example, Knuth [11, Exercise 20, p. 70; solution, p. 592]. Although Knuth discusses only binary trees, his statement holds for all trees.)

Lemma 2.2. *Let F be a forest with n vertices. Then*

$$\sum_L (1+\alpha)^{\operatorname{prop} L} = n! \prod_{v \in V(F)} \left(1 + \frac{\alpha}{h(v)}\right), \tag{2.1}$$

where L runs over all labelings of F.

Proof. The left side of (2.1) is equal to $\sum_{L,S} \alpha^{|S|}$ where L runs over all labelings of F and S runs over all subsets of the set of vertices of F that are proper with respect to L. The right side of (2.1) is equal to

$$\sum_{S \subseteq V(F)} \frac{n!}{\prod_{v \in S} h(v)} \alpha^{|S|}.$$

The result then follows from Lemma 2.1. □

Let us say that two labelings of a forest F are *equivalent* if there is an automorphism of F that takes one labeling to the other. (An automorphism of an ordered forest must preserve the order structure, so ordered forests have no nontrivial automorphisms.) Let F be a forest on n vertices with automorphism group G. Then the $n!$ labelings of F fall into $n!/|G|$ equivalence classes. It is clear that equivalent labelings have the same number of proper vertices, so we obtain the following lemma by dividing (2.1) by $|G|$.

Lemma 2.3. *Let F be a forest with n vertices and let $\mathcal{L}$ be a set of representatives of equivalence classes of labelings of F. Then*

$$\sum_{L \in \mathcal{L}} (1+\alpha)^{\operatorname{prop} L} = |\mathcal{L}| \prod_{v \in V(F)} \left(1 + \frac{\alpha}{h(v)}\right). \tag{2.2}$$

Let us now define a *labeled forest* to be a forest whose vertex set is $[n]$ for some n. Thus replacing each vertex of a labeling of an arbitrary forest gives a labeled forest, and two labelings give the same labeled forest if and only if the labelings are equivalent.

Then (2.2), together with linearity, gives the following result.

Lemma 2.4. *Let $\mathcal{F}$ be a class of labeled forests with vertex set $[n]$ in which membership depends only on isomorphism class. Then*

$$\sum_{F\in\mathcal{F}}(1+\alpha)^{\operatorname{prop} F} = \sum_{F\in\mathcal{F}}\prod_{v\in[n]}\left(1+\frac{\alpha}{h(v)}\right).$$

For ordered forests, which have no nontrivial automorphisms, we can simplify this result, since each isomorphism class of ordered forests on n vertices corresponds to $n!$ labeled ordered forests. An *unlabeled forest* is formally defined as an isomorphism class of forests, but we think of an unlabeled forest as a forest in which the identity of the vertices is immaterial. Then the following lemma follows from either Lemma 2.3 or Lemma 2.4:

Lemma 2.5. *Let $\mathcal{A}$ be a set of unlabeled ordered forests with n vertices, and let $\mathcal{F}$ be the corresponding set of labeled ordered forests. Then*

$$\sum_{F\in\mathcal{F}}(1+\alpha)^{\operatorname{prop} F} = n!\sum_{F\in\mathcal{A}}\prod_{v\in V(F)}\left(1+\frac{\alpha}{h(v)}\right), \tag{2.3}$$

where the product is over all vertices v of F.

In later sections, it will be convenient to use the term "labeled forest" to refer more generally to any forest whose vertices are integers.

3. Exponential generating functions

We assume that the reader is familiar with the combinatorial interpretation of the operations of addition, multiplication, composition, and differentiation on exponential generating functions, as described, for example in [19, Chapter 5] or [1]. Rather than adapting the formal language of [1], we take an informal approach to describing the connection between exponential generating functions and the objects that they count. Thus in deriving functional or differential equations for generating functions we will describe the decompositions that lead to these equations, but omit the straightforward details of how the equations follow from the decompositions. All of our exponential generating functions will be in the variable x, so we will usually omit it, and write, for example A instead of $A(x)$. Since nearly all our generating functions are exponential, we will usually call them simply "generating functions"; when ordinary generating functions are used, we will note this explicitly.

There are two kinds of decompositions of labeled objects that are especially important: *unordered* and *ordered* decompositions. An *unordered* decomposition is a decomposition of a labeled object into an unordered collection of objects. An important example is the decomposition of a forest into its component trees. When there is an unordered decomposition of objects counted by a generating function f into objects counted by g, the generating functions are related by the "exponential formula" $f = e^g$. Decompositions may depend on a total ordering of the labels. For example, we can decompose a permutation (considered as a linear arrangement) by cutting it before each left-right minimum, yielding a set of permutations each starting with its smallest element. (Thus the permutation 4753126 is decomposed into $\{475, 3, 126\}$.) The permutation can be reconstructed from its constituents by arranging them in decreasing order of their first elements. The generating function identity in this example is

$$\sum_{n=0}^{\infty} n!\frac{x^n}{n!} = \exp\left(\sum_{n=1}^{\infty}(n-1)!\frac{x^n}{n!}\right). \tag{3.1}$$

An *ordered decomposition* is a decomposition into a sequence of objects. If objects counted by f have an ordered decomposition into objects counted by h then f and h are related by $f = 1/(1-h)$. The simplest example is the decomposition of a permutation, considered as a linear arrangement, into the sequence of its entries; here $g = x$ and $f = 1/(1-x)$. Another important example is the decomposition of an ordered forest into its constituent ordered trees.

4. Trees and forests

We introduce here the types of trees and forests that we will count by various parameters in later sections. Most of this material is well known, but it is convenient to clarify the definitions and state the known formulas that we will generalize later.

Let A be the generating function for labeled forests. Then since a tree consists of a root together with the forest of trees rooted at the children of the root, the generating function for trees is xA, and since a forest is a set of trees, the generating function for forests is e^{xA}. Thus A satisfies the functional equation

$$A = e^{xA}. \tag{4.1}$$

It is well known that the solution of (4.1) is

$$A = \sum_{n=0}^{\infty}(n+1)^{n-1}\frac{x^n}{n!},$$

and that more generally,

$$A^c = \sum_{n=0}^{\infty}c(n+c)^{n-1}\frac{x^n}{n!}. \tag{4.2}$$

Equivalent formulas can be found in [8, Sections 1.2.5 and 1.2.7], [19, Proposition 5.4.4] and [14, p. 97]. Since $A^c = e^{c\log A} = e^{cxA}$, the coefficient of c^i in A^c is the generating function for forests of i trees.

We say that a vertex of a tree has *degree* d if it has d children. A k-ary tree is an ordered tree in which every vertexe has degree k or 0. The ordinary generating function D for unlabeled k-ary trees in which the coefficient of x^n is the number of k-ary trees with n vertices of degree k (and thus $1 + (k-1)n$ vertices of degree 0) satisfies

$$D = 1 + xD^k, \tag{4.3}$$

and it is well known that

$$D^c = \sum_{n=0}^{\infty} \frac{c}{kn+c} \binom{kn+c}{n} x^n. \tag{4.4}$$

See, for example, [8, Exercise 2.7.1], [19, Proposition 6.2.2], and [9, pp. 200–201]. In computing hook lengths of k-ary trees, we will ignore vertices of degree 0. Thus the hook length of the root of the ternary tree in Figure 1 is 6.

For our purposes a labeled k-ary tree is a k-ary tree in which only the vertices of degree k are labeled. Figure 1 shows a labeled ternary tree. The exponential generating

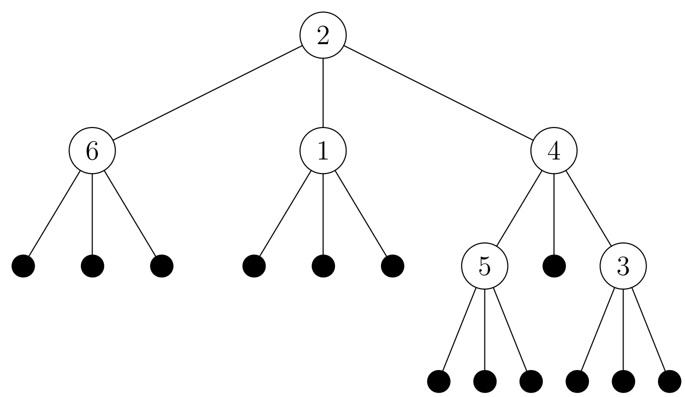

Figure 1: A labeled ternary tree

function for labeled k-ary trees is the same, as a power series, as the ordinary generating function for unlabeled k-ary trees, but as an exponential generating function we would rewrite (4.4) as

$$D^c = \sum_{n=0}^{\infty} c(kn+c-1)(kn+c-2)\cdots((k-1)n+c+1)\frac{x^n}{n!}$$

The generating function E for ordered trees (by the total number of vertices) satisfies $E = x/(1-E)$. As in the case of k-ary trees this formula holds for both the ordinary generating function, which counts unlabeled ordered trees, and for the exponential generating function, which counts labeled ordered trees. The generating function F for ordered forests (in either the unlabeled or labeled version) satisfies $F = 1/(1-E)$ and $E = xF$, and thus $F = 1/(1-xF)$.

The formula $F = 1/(1 - xF)$ is easily transformed into $F = 1 + xF^2$, so, as is well known, the generating function for ordered forests is the same as that for binary trees. More generally, the equations $F = 1/(1 - xF^k)$ and $F = 1 + xF^{k+1}$ are easily seen to be equivalent. It is clear that $F = 1 + xF^{k+1}$ is the functional equation for $(k + 1)$-ary trees. To find a combinatorial interpretation to

$$F = \frac{1}{1 - xF^k}, \tag{4.5}$$

we introduce *k-colored ordered forests*. A k-colored ordered forest is an ordered forest in which the edges are colored in colors 1, 2, ..., k with the property that among the children of any vertex, those of color 1 come first, then those of color 2, and so on. (But not every color must appear.) For example, Figure 2 shows a 2-colored ordered forest in

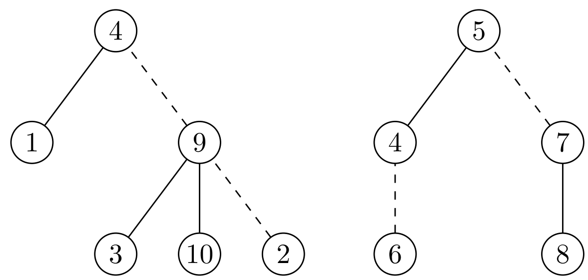

Figure 2: A 2-colored forest

which color 1 is represented by solid lines and color 2 by dotted lines.

It is easily seen that the generating for k-colored ordered forests satisfies (4.5). Moreover, the well-known bijection from ordered forests to binary trees (see, e.g., Stanton and White [20, p. 61]) generalizes easily to a bijection from k-colored ordered forests to $(k + 1)$-ary trees.

5. The generating function for $P_n(a, b, c)$

Our main objects of study are the polynomials $P_n(a, b, c)$ defined by

$$P_n(a, b, c) = c \prod_{i=1}^{n-1} (ia + (n - i)b + c). \tag{5.1}$$

Note that $P_n(a, b, c)$ is symmetric in a and b. It is also homogeneous of degree of n in a, b, and c, so one of the parameters is redundant.

We denote by $Q_n(a, b)$ the coefficient of c in $P_n(a, b, c)$, so

$$Q_n(a, b) = c^{-1} P_n(a, b, c)\Big|_{c=0} = \prod_{i=1}^{n-1} (ia + (n - i)b).$$

The polynomials $P_n(a,b,c)$ generalize the counting formulas of the previous section, since

$$c(n+c)^{n-1} = P_n(1,1,c)$$

and

$$\frac{c}{kn+c}\binom{kn+c}{n}n! = P_n(k-1,k,c). \tag{5.2}$$

Since the polynomial (5.2) has two free parameters, by rescaling we can recover $P_n(a,b,c)$ and we find that for $a \neq b$, we have

$$P_n(a,b,c)/n! = (b-a)^n \frac{\bar{c}}{\bar{b}n+\bar{c}}\binom{\bar{b}n+\bar{c}}{n}, \tag{5.3}$$

where $\bar{b} = b/(b-a)$ and $\bar{c} = c/(b-a)$.

It follows from (5.3), (4.3), and (4.4) that

$$\sum_{n=0}^{\infty} P_n(a,b,c)\frac{x^n}{n!} = G^{c/(b-a)},$$

where G satisfies

$$G = 1 + (b-a)xG^{b/(b-a)}. \tag{5.4}$$

It does not seem to be easy to obtain a combinatorial interpretation for $P_n(a,b,c)$ directly from (5.4). Instead, we derive a differential equation whose combinatorial meaning is clearer. Let $H = G^{1/(b-a)}$, so that

$$\sum_{n=0}^{\infty} P_n(a,b,c)\frac{x^n}{n!} = H^c. \tag{5.5}$$

We claim that H satisfies the differential equation

$$H' = H^{a+1} + bxH^aH'. \tag{5.6}$$

To see this, (5.4) gives

$$H^{b-a} = 1 + (b-a)xH^b. \tag{5.7}$$

Differentiating with respect to x and dividing both sides by $(b-a)H^{b-a-1}$ gives (5.6). (Although this derivation is not valid for $a = b$, a limit argument shows that (5.6) does hold in this case.) Conversely, (5.6), together with the initial value $H(0) = 1$, implies (5.7), and thus (5.5).

We note that (5.7) can be written in the more symmetric form

$$H^b(1 + axH^a) = H^a(1 + bxH^b); \tag{5.8}$$

since all coefficients are positive we might hope for a combinatorial interpretation, but we have not found one.

We now give a completely self-contained proof that (5.6) implies (5.5). For our applications, it is convenient to switch a and b from (5.6) and to add a redundant variable.

Theorem 5.1. *Let $g = g(x)$ be the solution of the differential equation*

$$g' = ug^{b+1} + axg^b g' \tag{5.9}$$

with $g(0) = 1$, where the prime denotes the derivative with respect to x. Then

$$g^c = \sum_{n=0}^{\infty} P_n(a, bu, cu)\frac{x^n}{n!} \tag{5.10}$$

and

$$\log g = \sum_{n=1}^{\infty} uQ_n(a, bu)\frac{x^n}{n!}. \tag{5.11}$$

Proof. Let $\bar{P}_n(c)$ be the coefficient of $x^n/n!$ in g^c, where the dependence of $\bar{P}_n(c)$ on u, a, and b is implicit. From (5.9) we obtain

$$(g^c)' = cug^{b+c} + acxg^{b+c-1}g' = cug^{b+c} + \frac{ac}{b+c}x(g^{b+c})' \tag{5.12}$$

Equating coefficients of $x^n/n!$ in (5.12) gives

$$\begin{aligned}\bar{P}_{n+1}(c) &= cu\bar{P}_n(b+c) + \frac{nac}{b+c}\bar{P}_n(b+c)\\ &= \frac{c(na+bu+cu)}{b+c}\bar{P}_n(b+c).\end{aligned}$$

Since $\bar{P}_0(c) = 1$, we get

$$\bar{P}_n(c) = cu(a+(n-1)bu+cu)(2a+(n-2)bu+cu)\cdots((n-1)a+bu+cu).$$

The formula for $\log g$ follows by taking the coefficient of c in g^c. □

6. Counting forests by proper vertices

As noted earlier, $P_n(1,1,1) = (n+1)^{n-1}$ is the number of forests on $[n]$. Thus we might hope that $P_n(a,b,c)$ counts these forests according to some parameters. Theorem 6.1 shows that this is the case.

For a forest F, let $\operatorname{prop} F$ be the number of proper vertices and let $\operatorname{tree} F$ be the number of trees in F. We give two proofs of the following result, one using the differential equation for the generating function for $P_n(a,b,c)$ and one which is more combinatorial, though not completely bijective. A bijective proof of Theorem 6.1 has recently been given by Seo and Shin [16].

Theorem 6.1. *We have*

$$\sum_F a^{n-\operatorname{prop} F} b^{\operatorname{prop} F - \operatorname{tree} F} c^{\operatorname{tree} F} = P_n(a,b,c), \tag{6.1}$$

where the sum is over all forests F on $[n]$.

First proof. Let I be the generating function for forests by the number of proper vertices; i.e.,

$$I = \sum_{n=0}^{\infty} \sum_{F} u^{\operatorname{prop} F} \frac{x^n}{n!}$$

where F runs over all forests on $[n]$.

Let J be the corresponding generating function for trees by proper vertices, so that $I = e^J$ by the exponential formula. Next we will show that $J' = uI + xI'$. The coefficent of $x^n/n!$ in J' counts trees on $\{0, 1, \dots, n\}$ by proper vertices. Let T be such a tree. Then T consists of a root together with the forest F obtained by deleting the root. If the root of T is 0 then T has one more proper vertex than F. Thus the contribution to J' from trees in which the root is 0 is uI. If the root is not 0, then T has an improper root with a label in $[n]$, and 0 appears in F. Thus the contribution to J' from trees in which the root is not 0 is xI'. So $J' = uI + xI'$ as claimed.

Then

$$I' = (e^J)' = e^J J' = I(uI + xI') = uI^2 + xII',$$

so by Theorem 5.1,

$$I = \sum_{n=0}^{\infty} P_n(1, u, u) \frac{x^n}{n!},$$

as shown by Seo [15, Corollary 8].

The number of proper vertices of a forest is the same as the number of proper vertices in all its constituent trees, so $I^c = \exp(cJ)$ counts forests by proper vertices in which every tree is weighted by c, and thus

$$\sum_{F} u^{\operatorname{prop} F} c^{\operatorname{tree} F} = P_n(1, u, cu) \tag{6.2}$$

Using the fact that $P_n(a, b, c)$ is a homogeneous polynomial of degree n in a, b, and c, we can rescale (6.2) to get (6.1). ☐

We can interpret (6.1) as counting forests on $[n]$ in which each vertex is assigned a weight. We will call a vertex *weakly proper* if it is proper, but not the smallest vertex in its tree and *minimal* if it is the smallest vertex in its tree. Then (6.1) counts forests in which every improper vertex is weighted a, every weakly proper vertex is weighted b, and every minimal vertex is weighted c.

Second proof of Theorem 6.1. Let $p_n(a, b, c)$ count forests F on $[n]$ in which every improper vertex is weighted a, every weakly proper vertex is weighted b, and every minimal vertex in a component of F is weighted c. Then $c^{-1} p_n(a, b, c)$ counts forests with the same weight on each vertex except that the smallest vertex of F is weighted by 1. Suppose that a forest F on $[n]$ is weighted in this way. If 1 is a root of F, delete the vertex 1 of F. Then we obtain a forest on $\{2, 3, \dots, n\}$ in which the minimal vertices are weighted either b or c. (See (a) of Figure 3.) Thus these forests are counted by $p_{n-1}(a, b, b+c)$. If 1 is not a root of F, interchange the label 1 and its parent i and delete the vertex i. Then we have a forest

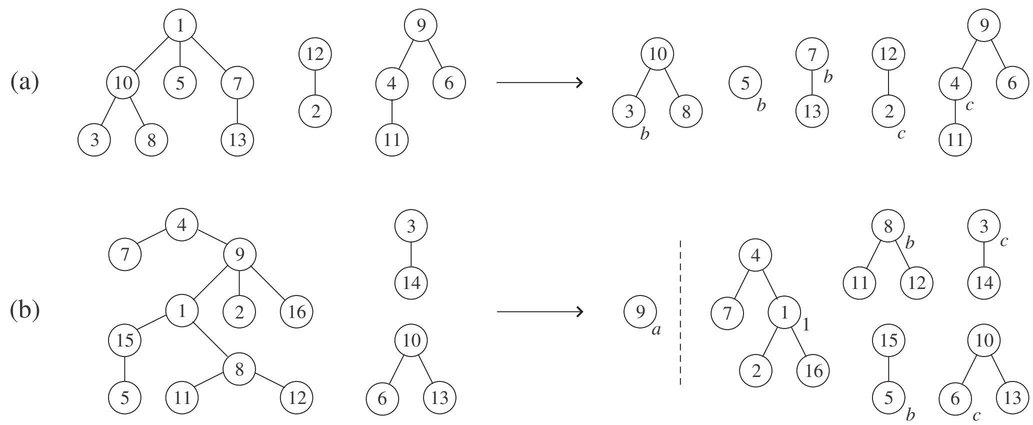

Figure 3: A weight preserving decomposition of forests

on $[n] \setminus \{i\}$ such that the smallest vertex 1 is weighted 1 and the other minimal vertices are weighted either b or c. Moreover, there are $n-1$ possibilities for the parent i of 1 and it is weighted a. Thus these forests are counted by $(n-1)\,a \cdot (b+c)^{-1} p_{n-1}(a,b,b+c)$. (See (b) of Figure 3.) Combining two cases yields

$$c^{-1} p_n(a,b,c) = p_{n-1}(a,b,b+c) + (n-1)\,a\,(b+c)^{-1} p_{n-1}(a,b,b+c),$$

with the initial condition $c^{-1} p_1(a,b,c) = 1$. It follows easily that

$$p_n(a,b,c) = P_n(a,b,c).$$

□

The symmetry $P_n(a,b,c) = P_n(b,a,c)$ implies that among forests of j trees, the number with i improper vertices is equal to the number with i weakly proper vertices. (It is clear that the case $j=1$ of this symmetry implies the general case.) We do not have a bijective proof of this symmetry.

It is instructive to look at the case $i=0$ of this symmetry, which can be explained by known bijections. By the homogeneity of the polynomials $P_n(a,b,c)$, the symmetry in this case is equivalent to the identity $P_n(0,1,c) = P_n(1,0,c) = c(c+1)\cdots(c+n-1)$, in which the coefficients are unsigned Stirling numbers of the first kind. First, $P_n(0,1,c)$ counts forests on $[n]$ with n proper vertices. These are increasing forests, and there is a well known bijection [18, p. 25] that takes an increasing forest of j trees to a permutation with j left-right minima. Next, $P_n(1,0,c)$ counts forests in which the number of proper vertices equals the number of trees. Since every tree has at least one leaf, and every leaf is proper, in these forests every tree has a single leaf, which is the least vertex of the tree. If we read the trees from leaf to root, we get a set of j permutations, each of which starts with its least element, and as described in Section 3 this set of permutations can be arranged to form a permutation with j left-right minima.

A different symmetry of forests that has a very similar special case is described in [7]. (See also [10].)

Since the proper vertices of a forest are the same as the proper vertices of its component trees, (5.11) gives a corresponding result for trees;

Corollary 6.2. *We have*

$$\sum_T a^{n-\operatorname{prop} T} b^{\operatorname{prop} T-1} = Q_n(a,b),$$

where T runs over all trees on $[n]$. ☐

Applying Lemma 2.4 gives a hook-length formula for forests:

Corollary 6.3.

$$\sum_F c^{\operatorname{tree} F} \prod_{v\in[n]} \left(1+\frac{\alpha}{h(v)}\right) = P_n(1, 1+\alpha, c(1+\alpha)),$$

where F runs over all forests on $[n]$. ☐

In particular, taking $\alpha = 1$ gives

$$\frac{1}{n!}\sum_F c^{\operatorname{tree} F} \prod_{v\in[n]} \left(1+\frac{1}{h(v)}\right) = \frac{1}{n!}\,P_n(1,2,2c) = \frac{c}{n+c}\binom{2n+2c}{n},$$

which for $c = 1$ is the Catalan number C_{n+1}.

7. k-ary trees

In this section we give an analogue of Theorem 6.1 for k-ary trees. We first count k-ary trees by proper vertices, obtaining a result of Seo [15, Corollary 7].

Theorem 7.1. *We have*

$$\sum_T u^{\operatorname{prop} T} = P_n(k, (k-1)u, u), \tag{7.1}$$

where the sum is over all k-ary trees T on $[n]$.

Proof. Let J be the generating function for $\sum_T u^{\operatorname{prop} T}$ as in the statement of the theorem. By reasoning similar to that used in the first proof of Theorem 6.1, we will obtain the differential equation

$$J' = uJ^k + kxJ^{k-1}J'. \tag{7.2}$$

We consider k-ary trees on the vertex set $\{0, 1, \ldots, n\}$ and look at the location of vertex 0. The term uJ^k in (7.2) corresponds to the case in which 0 is the root of the tree. If 0 is not the root, the tree can be decomposed into a root R (the factor x), the subtree containing 0 rooted at a child of R (the factor J') and a $(k-1)$-tuple of the other subtrees rooted at the children of R (the factor J^{k-1}). The factor k is the number of positions in which the subtree containing 0 can lie in relation to the other subtrees.

So by Theorem 5.1,

$$J = \sum_{n=0}^{\infty} P_n(k, (k-1)u, u)\frac{x^n}{n!}. \tag{7.3}$$

□

Applying Lemma 2.2, gives the following hook-length formula for k-ary trees, which is equivalent to Corollary 3.11 of Du and Liu [5]. The case $k = 2$, $\alpha = 1$ is equivalent to Postnikov's formula (1.3).

Corollary 7.2. *We have*

$$n! \sum_T \prod_{v \in V(T)} \left(1 + \frac{\alpha}{h(v)}\right) = P_n(k, (k-1)(1+\alpha), 1+\alpha)$$

where T runs over all unlabeled k-ary trees on n vertices.

We can generalize Theorem 7.1 to include a parameter that corresponds to the number of trees in a forest, and thereby find a result analogous to Theorem 6.1. In order to do this, we must find an unordered decomposition of k-ary trees that is compatible with proper vertices; i.e., no improper vertex of a k-ary tree becomes proper when restricted to its component. To define this decomposition, we first describe a path in each k-ary tree that we call its *decomposing path*. The decomposing path starts at the root and ends at a leaf. From any labeled (i.e., non-leaf) vertex v of the k-ary tree, the path goes to either the first or second child of v according to the following rule: the path goes from v to its first child unless v is improper and the smallest descendant of v is also a descendant of the first child of v; in this case the path goes from v to the second child of v. When the path reaches a leaf, it stops.

For example, Figure 4 shows a binary tree with its decomposing path. The unordered decomposition of the k-ary tree is obtained by removing each edge on its decomposing path (except the last edge, which ends at a leaf) and replacing each lost child with a leaf. We denote by $\operatorname{comp} T$ the number components in the decomposition of a k-ary tree T, which is the same as the number of labeled vertices in the decomposing path. The unordered decomposition of the tree of Figure 4 is shown in Figure 5.

Theorem 7.3. *For $k > 1$ we have*

$$\sum_T v^{n-\operatorname{prop} T} u^{\operatorname{prop} T - \operatorname{comp} T} w^{\operatorname{comp} T} = P_n(kv, (k-1)u, w). \tag{7.4}$$

where the sum is over all k-ary trees T on $[n]$.

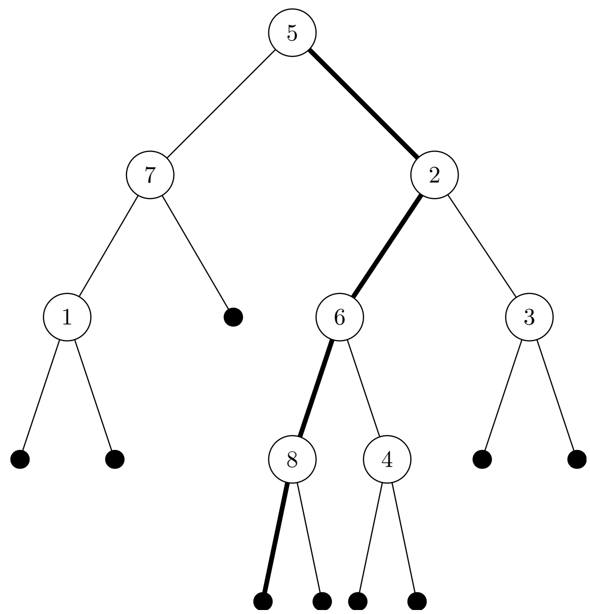

Figure 4: A binary tree and its decomposing path

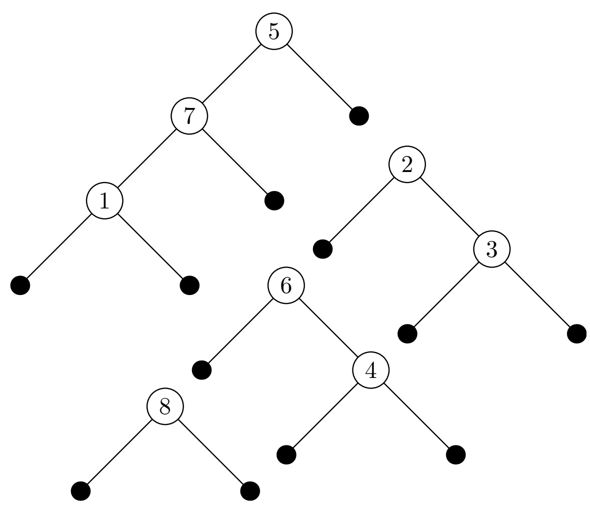

Figure 5: Decomposition of the tree of Figure 4

Proof. We first prove

$$\sum_T u^{\operatorname{prop} T} c^{\operatorname{comp} T} = P_n(k, (k-1)u, cu). \tag{7.5}$$

By (5.10) and Theorem 7.1, to prove (7.5) it is sufficient to show that the generating function for the left side is J^c, where J is as in the proof of Theorem 7.1. This follows from a verification that the unordered decomposition for a k-ary tree has several properties. Let the decomposition of a k-ary tree T be $\{T_1, \dots, T_m\}$. We must show that $\operatorname{comp} T_i = 1$ for each i, that T can be recovered from $\{T_1, \dots, T_m\}$, and that $\operatorname{prop} T = \sum_i \operatorname{prop} T_i$. Moreover we must also show that any such set of trees $\{T_1, \dots, T_m\}$ with disjoint labels comes from the decomposition of some k-ary tree.

The components T_i are characterized by the property that either the first child of the root of T_i is a leaf, or the root of T_i is improper, its smallest descendant is a descendant of its first child, and its second child is a leaf. This implies that $\operatorname{comp} T_i = 1$. We can reconstruct the order of the components since the first component T_1 will contain the smallest vertex of T, the next component T_2 will contain the smallest vertex of T not in T_1, and so on. To reconstruct T, let r_i denote the root of tree T_i. Then in T, r_{i+1} must have been either the first or second child of r_i: r_{i+1} was the first child of r_i in T if and only if the first child of T_i is a leaf. The decomposition preserves proper and improper vertices since the smallest descendant of any improper vertex v on the decomposing path remains a descendant of v in its component (and descendants of vertices not on the decomposing path are not changed). The last assertion is clear.

The homogeneity of $P_n(a, b, c)$, together with (7.5), implies that

$$\sum_T v^{n - \operatorname{prop} T} u^{\operatorname{prop} T} c^{\operatorname{comp} T} = P_n(kv, (k-1)u, cu).$$

Replacing c with w/u gives (7.4). $\square$

In should be pointed out that Theorem 7.3 is not valid for unary trees. However, there is a different (and simpler) decomposition for unary trees that gives an analogue of Theorem 7.3 in this case.

In contrast to Corollary 6.3, we cannot use Theorem 7.3 to add a parameter for components to Corollary 7.2, since the number of components of a k-ary tree is not an isomorphism invariant.

8. Ordered forests

Next we count k-colored ordered forests by proper vertices. The case $k = 1$ of the next theorem was given by Seo [15, equation (14)].

Theorem 8.1. *We have*

$$\sum_F u^{\operatorname{prop} F} = P_n(k, (k+1)u, u), \tag{8.1}$$

where the sum is over all k-colored ordered forests F on $[n]$, and for $n \geq 1$,

$$\sum_T u^{\operatorname{prop} T} = -P_n(k, (k+1)u, -u), \tag{8.2}$$

where the sum is over all k-colored ordered trees T on $[n]$.

Proof. Let K be the generating function for k-colored ordered forests, where a forest F is weighted $u^{\operatorname{prop} F}$. Let L be the analogous generating function for k-colored ordered trees, so $K = 1/(1-L)$. We claim that

$$L' = uK^k + kxK^{k-1}K'. \tag{8.3}$$

By similar reasoning to that used before, we see that in (8.3) the term uK^k counts k-colored ordered trees on $\{0, 1, \ldots, n\}$ in which 0 is the root, and for each i from 1 to k, $xK^{k-1}K'$ is the generating function for forests in which 0 is in a forest joined to the root by an edge of color i. It follows from (8.3) and $K = 1/(1-L)$ that

$$K' = uK^{k+2} + kxK^{k+1}K'.$$

So by Theorem 5.1 we have

$$K = \sum_{n=0}^{\infty} P_n(k, (k+1)u, u)\frac{x^n}{n!}, \tag{8.4}$$

proving the first formula.

For the second formula, we note that

$$L = 1 - K^{-1},$$

and a formula for the coefficients of K^{-1} follows from (8.4) and (5.10).

□

Applying Lemma 2.5 to Theorem 8.3 gives hook length formulas for k-colored ordered forests and trees. A result equivalent to the case $k = 1$ of (8.5) was given by Du and Liu [5, Theorem 4.14].

Corollary 8.2. *We have*

$$n! \sum_F \prod_{v \in V(F)} \left(1 + \frac{\alpha}{h(v)}\right) = P_n(k, (k+1)(1+\alpha), 1+\alpha), \tag{8.5}$$

where the sum is over all unlabeled k-colored ordered forests F on n vertices, and

$$n! \sum_T \prod_{v \in V(T)} \left(1 + \frac{\alpha}{h(v)}\right) = -P_n(k, (k+1)(1+\alpha), -(1+\alpha)),$$

where the sum is over all unlabeled k-colored ordered trees F on n vertices.

From the ordered decomposition of k-colored ordered forests into k-colored ordered trees, we get an unordered decomposition as described in Section 3. One way to do this is to look at the sequence of roots, and cut the forest after each left-right minimum in this sequence. (Instead of using the root of each tree for comparison, we could use the smallest or largest element of each tree.) Thus we decompose each k-colored ordered forest into components which are k-colored ordered forests in which the first root is the smallest. For a k-colored ordered forest F, let $\operatorname{comp} F$ be the number of components in this sense; that is, the number of left-right minima in the sequence of roots. Then since

$$K^c = \sum_{n=0}^{\infty} P_n(k,(k+1)u,cu)\frac{x^n}{n!},$$

we have

$$P_n(k,(k+1)u,cu) = \sum_F u^{\operatorname{prop} F} c^{\operatorname{comp} F}.$$

By the same reasoning as in Theorem 7.3, this leads to the analogous formula for k-colored ordered trees:

Theorem 8.3. *We have*

$$\sum_F u^{n-\operatorname{prop} F} v^{\operatorname{prop} F-\operatorname{comp} F} w^{\operatorname{comp} F} = P_n(ku,(k+1)v,w),$$

where the sum is over all k-colored ordered forests on $[n]$.

As noted in Section 4, there is a simple bijection from k-colored ordered forests to $(k+1)$-ary trees, but this bijection does not preserve proper vertices. However, we do have a relation between proper vertices of these forests and trees:

Corollary 8.4. *We have*

$$\sum_T v^{n-\operatorname{prop} T} u^{\operatorname{prop} T-\operatorname{comp} T} w^{\operatorname{comp} T} = \sum_F u^{n-\operatorname{prop} F} v^{\operatorname{prop} F-\operatorname{comp} F} w^{\operatorname{comp} F}$$

where T runs over $(k+1)$-ary trees on $[n]$ and F runs over k-colored ordered forests on $[n]$.

Proof. By Theorem 8.3, the sum on F is equal to $P_n(ku,(k+1)v,w)$. In (7.4) we replace k by $k+1$, switch u and v, and use $P_n(a,b,c) = P_n(b,a,c)$ to show that the sum on T is also equal to $P_n(ku,(k+1)v,w)$. $\square$

9. Descents in trees

In this section, we describe another forest interpretation, due to Eğecioğlu and Remmel [6], for $P_n(a,b,c)$. If vertex i is the parent of vertex j in a tree, we call the edge $\{i,j\}$ an *ascent* if $i < j$ and a *descent* if $i > j$, and for a forest F we write $\operatorname{asc} F$ for the number of ascents of F and $\operatorname{des} F$ for the number of descents of F. (A different notion of descents in trees was studied in [7].) We use the differential equation approach to count forests by ascents and descents; Eğecioğlu and Remmel [6] gave a bijective proof of a more general result.

Theorem 9.1. *We have*

$$\sum_F a^{\operatorname{des} F} b^{\operatorname{asc} F} c^{\operatorname{tree} F} = P_n(a, b, c), \tag{9.1}$$

where the sum is over all forests F on $[n]$.

Proof. Since the ascents and descents of a forest are the same as those of its constituent trees, it is sufficient to prove the case $c = 1$ of (9.1). Let M be the generating function for the polynomials $\sum_F a^{\operatorname{des} F} b^{\operatorname{asc} F}$. We shall show that M satisfies the differential equation

$$M' = M^{b+1} + axM^b M',$$

which by Theorem 5.1 implies the result.

Let $N = \log M$, so that M counts trees by descents and ascents. Now let F be a forest on the vertex set $\{0, 1, \ldots, n\}$. We decompose F into forests F_1 and F_2 that together contain all the vertices of F other than 0. We let F_1 be the forest of trees rooted at the children of 0, and we let F_2 be the forest of all vertices that are not descendants of 0. In order to reconstruct F from F_1 and F_2 we consider two cases. First, suppose that 0 is a root of F. Then F is determined by F_1 and F_2 and the ascents and descents of F are the same as those of F_1 and F_2, except that F has one additional ascent for each edge joining 0 to one of its children. The contribution to M' from forests in which 0 is a root is thus $e^{bN}M = M^b M = M^{b+1}$.

If 0 is not a root of F, then to reconstruct F from F_1 and F_2 it is sufficient to know the parent of 0 in F; thus F is determined by F_1, F_2, and the choice of a vertex in F_2. All of the ascents and descents of F_1 and F_2 will be ascents and descents of F, and as in the first case, F will have an extra ascent for every child of 0. In addition, F will have an extra descent for the edge joining 0 to its parent. The generating function for forests with a marked vertex is xM', so the contribution to M' from forests in which 0 is not a root is $aM^b \cdot xM'$. ∎

10. Parking functions

A parking function on an n-set K is a function $f : K \to [n+1]$ satisfying a certain property, described below. We think of the elements of K as cars and the elements of $[n+1]$ as parking spaces and we think of f as a function that assigns to each car its favorite parking space. The cars arrive in some order at their favorite parking spaces. If space $f(z)$ is not already occupied when car z arrives there, then car z parks in space $f(z)$. More generally, if there is some available space s, with $f(z) \le s < n+1$, then car z parks in the first (i.e., smallest) such space. Otherwise, car z does not park at all. If all cars are able to park, then we call f a *parking function*. Although cars are not allowed to park in space $n+1$, we will see that the existence of this space is useful. It is well known (and easy to see) that whether all cars are able to park or not does not depend on their arrival order. However, we will assume that the cars are totally ordered and that they arrive in increasing order.

It is useful to represent the parking function $f : K \to [n+1]$ by the ordered partition $(B_1, \ldots, B_{n+1})$ of K in which $B_i = f^{-1}(i)$; i.e., B_i is the set of cars that prefer to park in space i. A necessary and sufficient condition for f to be a parking function is that for each i, $|B_1| + \cdots + |B_i| \geq i$.

It is well known that the number of parking functions on $[n]$ is $(n+1)^{n-1}$ [19, p. 140]. So the exponential generating function for parking functions is

$$A = \sum_{n=0}^{\infty} (n+1)^{n-1} \frac{x^n}{n!}.$$

We consider a slight generalization of parking functions. Instead of taking the set of parking spaces for n cars to be $[n+1]$, we take it to be $[n+c]$, for some positive integer c (but we keep the number of cars at n). Then we call f a *c-parking function* if all cars are able to park using the same parking procedure as before, except that cars may now park in spaces 1 to $n+c-1$. An alternative characterization of c-parking functions is that with $B_i = f^{-1}(i)$, we have $|B_1| + \cdots + |B_i| \geq i - c + 1$. It is clear that if all cars park, they will leave exactly c empty spaces, one of which will be the last space, $n+c$. Thus if we represent a c-parking function by an ordered partition, then cutting it after every empty parking space decomposes it into a concatenation of c ordinary parking functions. Therefore the generating function for c-parking functions is

$$A^c = \sum_{n=0}^{\infty} c(n+c)^{n-1} \frac{x^n}{n!}.$$

Further properties of c-parking functions can be found in Yan [21].

If f is a parking function $K \to [n+1]$ then we call a car $z \in K$ *lucky* if it succeeds in parking in its preferred space. Let $\operatorname{lucky} f$ denote the number of lucky cars associated to f.

Theorem 10.1. *We have*

$$\sum_f u^{\operatorname{lucky} f} = P_n(1, u, u),$$

where the sum is over all parking functions $f : [n] \to [n+1]$.

Proof. Let $q_n(u) = \sum_f u^{\operatorname{lucky} f}$, where the sum is over all parking functions f on $[n]$ and let $Q = \sum_{n=0}^{\infty} q_n(u) x^n / n!$. We will show that Q satisfies the differential equation

$$Q' = uQ^2 + xQ'Q, \tag{10.1}$$

which by Theorem 5.1 implies the result. To do this we evaluate Q' by counting parking functions $f : [n+1] \to [n+2]$. The restriction f^- of such a parking function to $[n]$ is a 2-parking function, which leaves two empty spaces, $e+1$ and $n+2$, for some e with $0 \le e \le n$. If $n+1$ is lucky in f, then $f(n+1)$ must be $e+1$, and the contribution to Q' from this case is uQ^2. If $n+1$ is not lucky in f then $f(n+1)$ is 1, 2, …, or e. In this case f^- decomposes into a parking function on e cars, with one of the first e spaces "marked," followed by a parking function on $n-e$ cars. The generating function for parking functions with a marked space is xQ' so the contribution to Q' from this case is $xQ'Q$. Adding together these two contributions gives (10.1). □

Since the number of lucky cars in a c-parking function is the total number of lucky cars among its constituents, the generating function for c-parking functions by lucky cars is Q^c, and we have the following result:

Corollary 10.2. *For any positive integer c, we have*

$$\sum_f u^{\mathrm{lucky}\, f} = P_n(1, u, cu),$$

where the sum is over all c-parking functions $f : [n] \to [n+c]$.

There is an ordered decomposition for parking functions that gives us a result analogous to equation (8.2). We modify the ordered partition representation of a parking function $f : K \to [n+1]$, where $|K| = n$, by removing the empty block B_{n+1}. Let us call the result the *reduced ordered partition* corresponding to f. Then f is called a *prime parking function* if its reduced ordered partition cannot be decomposed as a nontrivial concatenation of reduced ordered partitions for parking functions; equivalently, if $(B_1, B_2, \ldots, B_n)$ is the reduced ordered partition of f, then $|B_1| + |B_2| + \cdots + |B_i| > i$ for $1 \le i < n$. Then every parking function can be decomposed uniquely into prime parking functions.

Corollary 10.3. *For $n \ge 1$,*

$$\sum_f u^{\mathrm{lucky}\, f} = -P_n(1, u, -u),$$

where the sum is over all prime parking functions f on $[n]$.

Proof. Let R be the generating for prime parking functions f weighted by $u^{\mathrm{lucky}\, f}$. Then since the number of lucky cars in a parking function is compatible with the decomposition into prime parking functions, we have $Q = 1/(1-R)$, with Q as in the proof of Theorem 10.1. Thus $R = 1 - Q^{-1}$, and the result follows from Theorem 10.1 and (5.10). □

Acknowledgment. The authors wish to thank an anonymous referee for several helpful suggestions.

References

[1] F. Bergeron, G. Labelle, and P. Leroux, *Combinatorial Species and Tree-Like Structures*, Encyclopedia of Mathematics and its Applications, 67, Cambridge University Press, Cambridge, 1997. Translated from the 1994 French original by Margaret Readdy.

[2] A. Cayley, A theorem on trees, *Quart. J. Math.* **23** (1889), 376–378.

[3] W. Y. C. Chen and V. J. W. Guo, Bijections behind the Ramanujan polynomials, *Adv. in Appl. Math.* **27** (2001), 336–356.

[4] W. Y. C. Chen and L. L. M. Yang, On the hook length formula for binary trees, arXiv:math.CO/0504260.

[5] R. R. X. Du and F. Liu, (k, m)-Catalan numbers and hook length polynomials for trees, arXiv:math.CO/0501147.

[6] Ö. Eğecioğlu and J. B. Remmel, Bijections for Cayley trees, spanning trees, and their q-analogues, *J. Combin. Theory Ser. A* **42** (1986), 15–30.

[7] I. M. Gessel, Counting forests by descents and leaves, *Electron. J. Combin.* **3** (1996), no. 2, #R8.

[8] I. P. Goulden and D. M. Jackson, *Combinatorial Enumeration*, Wiley, New York, 1983.

[9] R. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*, 2nd. edition, Addison-Wesley, Reading, MA, 1994.

[10] L. H. Kalikow, Symmetries in trees and parking functions, *Discrete Math.* **256** (2002), 719–741.

[11] D. E. Knuth, *The Art of Computer Programming, Volume 3: Searching and Sorting*, Addison-Wesley, Reading, MA, 1973.

[12] A. Postnikov, Permutohedra, associahedra, and beyond, *Retrospective in Combinatorics: Honoring Richard Stanley's 60th Birthday*, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2004. Slides of the talk are available at `http://www-math.mit.edu/~apost/talks/perm-slides.pdf`.

[13] A. Postnikov, Permutohedra, associahedra, and beyond, arXiv:math.CO/0507163.

[14] J. Riordan, *Combinatorial Identities*, Wiley, New York, 1968.

[15] S. Seo, A combinatorial proof of Postnikov's identity and a generalized enumeration of labeled trees, *Electron. J. Combin.* **11 (2)** (2004–2005), #N3.

[16] S. Seo and H. Shin, A generalized enumeration of labeled trees and reverse Prüfer algorithm, arXiv:math.CO/0601009.

[17] P. W. Shor, A new proof of Cayley's formula for counting labeled trees, *J. Combin. Theory Ser. A* **71** (1995), 154–158.

[18] R. P. Stanley, *Enumerative Combinatorics, Vol. 1*, Wadsworth & Brooks/Cole, Monterey, CA, 1986.

[19] R. P. Stanley, *Enumerative Combinatorics, Vol. 2*, Cambridge University Press, Cambridge, 1999.

[20] D. Stanton and D. White, *Constructive Combinatorics*, Springer-Verlag, New York, 1986.

[21] C. Yan, Generalized parking functions, tree inversions, and multicolored graphs. *Adv. in Appl. Math.* **27** (2001), 641–670.

[22] J. Zeng, A Ramanujan sequence that refines the Cayley formula for trees, *Ramanujan J.* **3** (1999), 45–54.