FPSAC 2009, Hagenberg, Austria DMTCS proc. **AK**, 2009, 793–804

A further correspondence between $(bc, \bar{b})$-parking functions and $(bc, \bar{b})$-forests

Heesung Shin[†] and Jiang Zeng[‡]

Université de Lyon; Université Lyon 1; Institut Camille Jordan, CNRS UMR 5208; 43 boulevard du 11 novembre 1l918, F-69622 Villeurbanne Cedex, France

Abstract. For a fixed sequence of n positive integers $(a, \bar{b}) := (a, b, b, \dots, b)$, an $(a, \bar{b})$-parking function of length n is a sequence $(p_1, p_2, \dots, p_n)$ of positive integers whose nondecreasing rearrangement $q_1 \le q_2 \le \cdots \le q_n$ satisfies $q_i \le a + (i-1)b$ for any $i = 1, \dots, n$. A $(a, \bar{b})$-forest on n-set is a rooted vertex-colored forests on n-set whose roots are colored with the colors $0, 1, \dots, a-1$ and the other vertices are colored with the colors $0, 1, \dots, b-1$. In this paper, we construct a bijection between $(bc, \bar{b})$-parking functions of length n and $(bc, \bar{b})$-forests on n-set with some interesting properties. As applications, we obtain a generalization of Gessel and Seo's result about $(c, \bar{1})$-parking functions [Ira M. Gessel and Seunghyun Seo, Electron. J. Combin. **11**(2)R27, 2004] and a refinement of Yan's identity [Catherine H. Yan, Adv. Appl. Math. **27**(2–3):641–670, 2001] between an inversion enumerator for $(bc, \bar{b})$-forests and a complement enumerator for $(bc, \bar{b})$-parking functions.

Résumé. Soit $(a, \bar{b}) := (a, b, b, \dots, b)$ une suite d'entiers positifs. Une $(a, \bar{b})$-fonction de parking est une suite $(p_1, p_2, \dots, p_n)$ d'entiers positives telle que son réarrangement non décroissant $q_1 \le q_2 \le \cdots \le q_n$ satisfait $q_i \le a + (i-1)b$ pour tout $i = 1, \dots, n$. Une $(a, \bar{b})$-forêt enracinée sur un n-ensemble est une forêt enracinée dont les racines sont colorées avec les couleurs $0, 1, \dots, a-1$ et les autres sommets sont colorés avec les couleurs $0, 1, \dots, b-1$. Dans cet article, on construit une bijection entre $(bc, \bar{b})$-fonctions de parking et $(bc, \bar{b})$-forêts avec des des propriétés intéressantes. Comme applications, on obtient une généralisation d'un résultat de Gessel-Seo sur $(c, \bar{1})$-fonctions de parking [Ira M. Gessel and Seunghyun Seo, Electron. J. Combin. **11**(2)R27, 2004] et une extension de l'identité de Yan [Catherine H. Yan, Adv. Appl. Math. **27**(2–3):641–670, 2001] entre l'énumérateur d'inversion de $(bc, \bar{b})$-forêts et l'énumérateur complémentaire de $(bc, \bar{b})$-fonctions de parking.

Keywords: Bijection, Forests, Parking functions

1 Introduction

It is well-known [Sta99] that parking functions and (rooted) forests on n-set are both counted by Cayley's formula $(n+1)^{n-1}$. Foata and Riordan [FR74] gave the first bijection between these two equinumerous sets. In the past years, many generalizations and refinements of this result were obtained (See [MR68,

[†] hshin@math.univ-lyon1.fr
[‡] zeng@math.univ-lyon1.fr

Kre80, Yan01, SP02, KY03, GS06]). In particular, Stanley and Pitman [SP02] introduced the notion of $(a, \bar{b})$-parking functions where a and b are two positive integers.

Recall that an $(a, \bar{b})$-*parking function (of length n)* (see [SP02]) is a sequence $(p_1, p_2, \dots, p_n)$ of positive integers whose nondecreasing rearrangement $q_1 \le q_2 \le \cdots \le q_n$ satisfies $q_i \le a + (i-1)b$ for $1 \le i \le n$. It is shown [SP02] that the number of $(a, \bar{b})$-parking functions is

$$a(a+bn)^{n-1}.$$

Looking for its forest counter parts, Yan [Yan01] defined a *(rooted)* $(a, \bar{b})$-*forest* (see section 2.2) to be a vertex-colored forest in which all roots are colored with the colors $0, 1, \dots, a-1$ and the other vertices are colored with the colors $0, 1, \dots, b-1$. She proved that the enumerator $\bar{P}_n^{(a,\bar{b})}(q)$ of complements of $(a, \bar{b})$-parking functions and the enumerator $I_n^{(a,\bar{b})}(q)$ of $(a, \bar{b})$-forests by the number of their inversions are identical, i.e.,

$$I_n^{(a,\bar{b})}(q) = \bar{P}_n^{(a,\bar{b})}(q). \tag{1}$$

It is an open problem to give a bijective proof of the identity (1). Generalizing a bijection of Foata and Riordan [FR74], Yan [Yan01] did give a bijection between $(a, \bar{b})$-forests and $(a, \bar{b})$-parking functions which is a bijective proof of (1) for $q = 1$, but this bijection does not keep track of the statistics involved in (1) even in ordinary $a = b = 1$ case. Note that Eu et al. [EFL05] were able to extend the bijection of Foata and Riordan to enumerate $(a, \bar{b})$-parking functions by their leading terms. Recently, Shin [Shi08] gave a bijective proof of (1) when $a = b = 1$.

A different refinement of Cayley's formula was given by Gessel and Seo [GS06]. Using generating functions, they showed that the enumerator of forests with respect to proper vertices and the number of trees and the lucky enumerator of $(a, \bar{1})$-parking function are both equal to

$$au \prod_{i=1}^{n-1} (i + u(n-i+a)).$$

Bijective proof of above results for $a = 1$ have been given by Seo and Shin [SS07] and Shin [Shi08].

In this paper, we prove three main results. First, in Theorem 1, we establish a bijection between $(bc, \bar{b})$-parking functions and $(bc, \bar{b})$-forests, which is a generalization of the first author's recent bijection [Shi08]. Secondly, in Theorem 4, we generalize the aforementioned formula of Gessel and Seo to $(bc, \bar{b})$ case. Finally, in Theorem 5, we extend Gessel and Seo's hook-length formula [GS06, Corollay 6.3] for forests to $(a, \bar{b})$-forests.

The rest of this paper is organized as follows: In Section 2, we introduce definitions of various statistics on general parking functions and forests. The main theorems of this paper are presented in Section 3. The proofs of main theorems are given in Sections 4, 5, 6.

2 Definitions

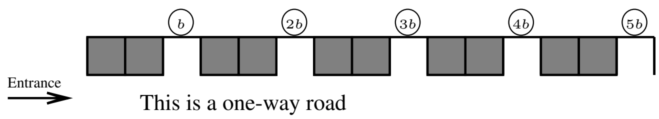
2.1 Statistics on $(bc, \bar{b})$-parking functions

From now on, we fix $a = bc$. We define a parking algorithm for $(bc, \bar{b})$-parking functions by generalizing algorithm in [GS06] for $(c, \bar{1})$-parking functions. Suppose that there are $1, 2, \dots, (n+c-1)b$ parking lots with only $n+c-1$ available parking spaces at $b, 2b, \dots, (n+c-1)b$, that means the positions are multiples of b.

Parking Space	1	2	③	4	5	⑥	7	8	⑨	10	11	⑫	13	14	⑮	16	17	⑱	
Cars' Number, c	□	□	3	□	□	1	□	□	5	□	□	$\emptyset$	□	□	4	□	□	2	
$\mathrm{jump}(P;c)$			0			1			7						0			2	$\mathrm{jump}(P)=10$
$\mathrm{block}(P;c)$			1			1			1						0			0	$\mathrm{block}(P)=3$
$\mathrm{jump}_{(6,\bar{3})}(P;c)$			3			4			10						0			2	$\mathrm{jump}_{(6,\bar{3})}(P)=19$
lucky car			✓												✓				$\mathrm{lucky}(P)=2$
critical car									✓						✓			✓	$\mathrm{crit}(P)=3$

$$\mathbf{JUMP}_{(6,\bar{3})}(P) = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

Fig. 1: A $(bc,\bar{b})$-parking function $P=(5,16,3,15,2)$ of length 5 and statistics of P for $b=3$, $c=2$ where circled numbers are available parking spaces



Given a $(bc,\bar{b})$-parking function $P=(p_1,p_2,\dots,p_n)$ of length n, suppose that Cars $1,2,\dots,n$ come to the parking lots in this order and car i prefers parking space p_i. We can park the n cars with $n+c-1$ parking spaces by the following *parking algorithm*: If p_i is occupied or non-available, then car i takes the next available space. If q_i be the actual parking space with i-th car for $i=1,\dots,n$, we define

$$\mathrm{park}(p_1,\dots,p_n)=(q_1,\dots,q_n).$$

In Figure 1, we give an example of a $(bc,\bar{b})$-parking function $(5,16,3,15,2)$ for $b=3$ and $c=2$. By the Parking Algorithm, we get a sequence with length 5,

$$\mathrm{park}(5,16,3,15,2)=(6,18,3,15,9).$$

The difference between the favorite parking space p_i and the actual parking space q_i is called the *jump* of car i, and denoted by $\mathrm{jump}(P;i)$, that is,

$$\mathrm{jump}(P;i)=q_i-p_i \quad \text{if} \quad \mathrm{park}(p_1,\dots,p_n)=(q_1,\dots,q_n).$$

Let $\mathrm{jump}(P)$ denote the sum of the jumps of P, that is,

$$\mathrm{jump}(P)=\sum_i \mathrm{jump}(P;i).$$

Clearly $\mathrm{jump}(P;i)\geq 0$. We say that car i is *lucky* if $\mathrm{jump}(P;i)=0$. Denote the number of lucky cars of P by $\mathrm{lucky}(P)$.

After parking all the n cars, there are $c-1$ non-occupied parking spaces which divide the parking lots into c *blocks* of parking lots. Let $\mathrm{block}(P;i)$ be the number of non-occupied parking spaces on the right of car i after running parking algorithm. Let $\mathrm{block}(P)$ denote the sum of blocks of all cars, i.e., $\mathrm{block}(P) = \sum_i \mathrm{block}(P;i)$. We define $(bc,\bar{b})$-jump of $(bc,\bar{b})$-parking function

$$\begin{aligned}\mathrm{jump}_{(bc,\bar{b})}(P;i) &= \mathrm{jump}(P;i) + b\cdot\mathrm{block}(P;i),\\ \mathrm{jump}_{(bc,\bar{b})}(P) &= \mathrm{jump}(P) + b\cdot\mathrm{block}(P) = bcn + \binom{n}{2}b - |P|,\end{aligned}$$

where $|P| = \sum p_i$. Note that $(bc,\bar{b})$-jump is identical to the complement of $|P|$ in [Yan01].

Let $\mathrm{lucky}_{j,k}(P)$ denote the number of cars i such that $\mathrm{block}(P;i) = j$ and $\mathrm{jump}(P;i) = k$. We define the multi-statistic $\mathbf{JUMP}_{(bc,\bar{b})}$ by

$$\mathbf{JUMP}_{(bc,\bar{b})}(P) = \begin{pmatrix} \mathrm{lucky}_{0,0}(P) & \mathrm{lucky}_{0,1}(P) & \cdots & \mathrm{lucky}_{0,N}(P) \\ \mathrm{lucky}_{1,0}(P) & \mathrm{lucky}_{1,1}(P) & \cdots & \mathrm{lucky}_{1,N}(P) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{lucky}_{c-1,0}(P) & \mathrm{lucky}_{c-1,1}(P) & \cdots & \mathrm{lucky}_{c-1,N}(P) \end{pmatrix},$$

where $N = \binom{n+1}{2}b - n$.

A car c is called *critical* if there are only former cars parked on the right of the block containing c after parking. If car c is critical in a $(bc,\bar{b})$-parking function P, $\mathrm{crit}(P;c) = 1$. Otherwise, $\mathrm{crit}(P;c) = 0$. Denote the number of critical cars in a $(bc,\bar{b})$-parking function P by $\mathrm{crit}(P)$.

As an example, a $(bc,\bar{b})$-parking function is given in Figure 1 for $b = 3$ and $c = 2$ in order to illustrate different statistics.

2.2 Statistics on $(bc,\bar{b})$-Forests

A *(rooted) forest* is a simple graph on $[n] = \{1,\dots,n\}$ without cycles, whose every connected component has a distinguished vertex, called a *root*. A *(rooted)* $(a,\bar{b})$*-forest* on $[n]$ is a pair (F,κ) where F is a forest on $[n]$, κ is a mapping from the set of vertices in F to non-negative integers such that $\kappa(v) < a$ if v is a root and $\kappa(v) < b$, otherwise.

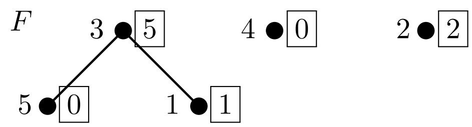
In a rooted forest F, a vertex j is called a *descendant* of a vertex i if the vertex i lies on the unique path from the root to the vertex j. In particular, every vertex is a descendant of itself. Denote the set of descendants of a vertex v by $D_F(v)$. The *hook-length* h_v of v is defined by the number of descendants of v in a forest. A vertex v is a parent of u if v and u are connected by one edge and u is a descendant of v.

As defined by Mallows and Riordan [MR68], an *inversion* in a rooted forest is an ordered pair (i,j) such that $i > j$ and j is a descendant of i. Let $\mathrm{Inv}(F;v)$ denote the set of ordered pairs (v,x) such that $v > x$ and $x \in D_F(v)$. Denote the number of all inversions in a rooted forest F by $\mathrm{inv}(F)$. We need to generalize the notion of inversions to $(bc,\bar{b})$-forests as follows: Let $\bar{\kappa}(v)$ denote the remainder of $\kappa(v)$ modulo b, i.e.,

$$\kappa(v) \equiv \bar{\kappa}(v) \mod b \quad \text{with } 0 \le \bar{\kappa}(v) \le b-1.$$

Define the inversion $\mathrm{inv}(F;v)$ of a $(bc,\bar{b})$-forest F by

$$\mathrm{inv}(F;v) = |\mathrm{Inv}(F;v)| + \bar{\kappa}(v)\cdot|D_F(v)|.$$



Vertex v	1	2	3	4	5	
$\kappa(v)$	1	2	5	0	0	
$\lvert\mathrm{Inv}(F;v)\rvert$	0	0	1	0	0	
$\bar{\kappa}(v)$	1	2	2	0	0	
$\lvert D_F(v)\rvert$	1	1	3	1	1	
$\mathrm{inv}(F;v)$	1	2	7	0	0	$\mathrm{inv}(F)=10$
$\mathrm{tcol}(F;v)$	1	0	1	0	1	$\mathrm{tcol}(F)=3$
$\mathrm{inv}_{(6,\bar{3})}(F;v)$	4	2	10	0	3	$\mathrm{inv}_{(6,\bar{3})}(F)=19$
proper vertex				✓	✓	$\mathrm{prop}(F)=2$
root		✓	✓	✓		$\mathrm{tree}(F)=3$

$$\mathbf{INV}_{(6,\bar{3})}(F)=\begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{pmatrix}$$

Fig. 2: A $(bc,\bar{b})$-forest F on $[5]$ and statistics of F for $b=3$, $c=2$ where $\kappa(v)$ is boxed

Let $\mathrm{inv}(F)$ denote the sum of $\mathrm{inv}(F;v)$ over all vertices v of F, i.e.,

$$\mathrm{inv}(F)=\sum_v \mathrm{inv}(F;v).$$

Given a $(bc,\bar{b})$-forest F, a vertex v is called a *proper vertex* if the vertex v is the smallest among all its descendants and its color is a multiple of b, that is, $\mathrm{inv}(F;v)=0$. Let $\mathrm{prop}(F)$ denote the number of all proper vertices in a rooted forest F. By definition, every leaf v with $\bar{\kappa}(v)=0$ is a proper vertex.

Denote the root of the tree including a vertex v in an $(bc,\bar{b})$-forest F by $R(v)$. A *tree-color* $\mathrm{tcol}(F;v)$ of a vertex v in a $(bc,\bar{b})$-forest F is defined by $\mathrm{tcol}(F;v)=\lfloor\frac{\kappa(R(v))}{b}\rfloor$. Let $\mathrm{tcol}(F)$ denotes the sum of root colors of all vertices, i.e., $\mathrm{tcol}(F)=\sum_v \mathrm{tcol}(F;v)$. We define the $(bc,\bar{b})$-inversion of $(bc,\bar{b})$-forest F by

$$\begin{aligned}\mathrm{inv}_{(bc,\bar{b})}(F;v) &= \mathrm{inv}(F;v)+b\cdot\mathrm{tcol}(F;v),\\ \mathrm{inv}_{(bc,\bar{b})}(F) &= \mathrm{inv}(F)+b\cdot\mathrm{tcol}(F).\end{aligned}$$

Note that $(bc,\bar{b})$-inversion is identical to the (bc,b)-inversion in [Yan01].

Let $\mathrm{prop}_{j,k}(F)$ denote the number of vertices such that $\mathrm{tcol}(F;v)=j$ and $\mathrm{inv}(F;v)=k$. We define the multi-statistic $\mathbf{INV}_{(bc,\bar{b})}$ by

$$\mathbf{INV}_{(bc,\bar{b})}(F)=\begin{pmatrix} \mathrm{prop}_{0,0}(F) & \mathrm{prop}_{0,1}(F) & \cdots & \mathrm{prop}_{0,N}(F) \\ \mathrm{prop}_{1,0}(F) & \mathrm{prop}_{1,1}(F) & \cdots & \mathrm{prop}_{1,N}(F) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{prop}_{c-1,0}(F) & \mathrm{prop}_{c-1,1}(F) & \cdots & \mathrm{prop}_{c-1,N}(F) \end{pmatrix},$$

where $N = \binom{n+1}{2}b - n$.

If a vertex v is a root of a forest F, we define $\mathrm{tree}(f;v) = 1$. Otherwise, $\mathrm{tree}(f;v) = 0$. Denote the number of trees (or roots) in a $(bc, \bar{b})$-forest F by $\mathrm{tree}(F)$.

In Figure 2, an example of a $(bc, \bar{b})$-forest F on n-set is given for $b = 3$ and $c = 2$ in order to illustrate different statistics.

3 Main Results

Let $PF_{(bc,\bar{b})}$ be the set of $(bc, \bar{b})$-parking functions of length n and $F_{(bc,\bar{b})}$ be the set of $(bc, \bar{b})$-forests on $[n]$. First of all, we recall the mapping $\varphi : F_{(1,\bar{1})} \to PF_{(1,\bar{1})}$ between forests and ordinary parking functions in [Shi08]. Given a forest $F \in F_{(1,\bar{1})}$ and a vertex $v \in [n]$, let h_v be the number of descendants of v in F and $D_F(v) = \{d_1, d_2, \ldots, d_{h_v}\}$ is the set of descendants of v in F. We define a cyclic permutation θ_v on $D_F(v)$ by

$$\theta_v = (d_1 d_2 \cdots d_{k-1} v)$$

where $d_1 > d_2 > \ldots > d_{k-1}$ are all the descendants of $v \in V(F)$ greater than v and $\theta_v(d_i) = d_{i+1}$ for $1 \le i \le k-1$ and $\theta_v(v) = d_1$. Let $\theta_F = \theta_1\theta_2 \cdots \theta_n$. We attach to each vertex v in F a triple of labels

$$(\theta_F(v), \mathrm{inv}(F : v), \mathrm{post}(\theta_F(F) : \theta_F(v)))$$

where $\theta_F(F)$ is a forest by relabeling v by $\theta_F(v)$ and $\mathrm{post}(F : v)$ is a postorder index of v in F. We define the mapping $f : [n] \to [n]$ by

$$v \mapsto \mathrm{post}(\theta_F(F) : \theta_F(v)) - \mathrm{inv}(F : v)$$

for every vertex v. The bijection $\varphi : F_{(1,\bar{1})} \to PF_{(1,\bar{1})}$ is defined by

$$\varphi(F) = (f(\theta_F^{-1}(1)), f(\theta_F^{-1}(2)), \ldots, f(\theta_F^{-1}(n))). \tag{2}$$

Now we generalize the mapping φ to a bijection between $(bc, \bar{b})$-forests and $(bc, \bar{b})$-parking functions. We define the mapping $\varphi : F_{(bc,\bar{b})} \to PF_{(bc,\bar{b})}$ as follows: Given a $F \in F_{(bc,\bar{b})}$, the connected components of a forest F can be classified according to tree-colors. Let F_k be the sub-forests of F satisfying

$$\mathrm{tcol}(F : v) = k$$

for all $v \in F_k$. We define a cyclic permutation θ_v on $D_F(v)$ as above. When we define a postorder index $\mathrm{post}(F : v)$ of v in F, forests $F_{c-1}, F_{c-2}, \ldots, F_0$ are traversed in this order. We attach to each vertex v in F a quadruple of labels

$$(\theta_F(v), \mathrm{inv}(F : v), \mathrm{post}(\theta_F(F) : \theta_F(v)), \mathrm{tcol}(F : v))$$

where $\theta_F(F)$ is a forest by relabeling v by $\theta_F(v)$. After that, we define the mapping $f : [n] \to [n]$ by

$$v \mapsto (\mathrm{post}(\theta_F(F) : \theta_F(v)) + c - 1 - \mathrm{tcol}(F : v))b - \mathrm{inv}(F : v)$$

on every vertex v. The mapping $\varphi : F_{(bc,\bar{b})} \to PF_{(bc,\bar{b})}$ is also defined by (2). For example, the forest F in Figure 2 goes to the parking function P in Figure 1 by the mapping φ.

Theorem 1 (Main Theorem) *The mapping φ is a bijection between $(bc,\bar{b})$-forests and $(bc,\bar{b})$-parking functions satisfying*

$$(\mathbf{INV}_{(bc,\bar{b})},\mathrm{tree})(F)=(\mathbf{JUMP}_{(bc,\bar{b})},\mathrm{crit})\varphi(F),$$

for all $(bc,\bar{b})$-forests F.

By definitions, the statistics $\mathrm{inv}_{(bc,\bar{b})}$, inv, tcol, and prop can be written as follows:

$$\begin{aligned}
\mathrm{inv}_{(bc,\bar{b})}(F) &= \mathrm{inv}(F)+b\cdot\mathrm{tcol}(F),\\
\mathrm{inv}(F) &= (1,1,1,\ldots,1)\,\mathbf{INV}_{(bc,\bar{b})}(F)(0,1,2,\ldots,N)^T,\\
\mathrm{tcol}(F) &= (0,1,2,\ldots,(c-1))\,\mathbf{INV}_{(bc,\bar{b})}(F)(1,1,1,\ldots,1)^T,\\
\mathrm{prop}(F) &= (1,1,1,\ldots,1)\,\mathbf{INV}_{(bc,\bar{b})}(F)(1,0,0,\ldots,0)^T.
\end{aligned}$$

Similarly, the statistics $\mathrm{jump}_{(bc,\bar{b})}$, jump, block, and lucky can also be written as follows:

$$\begin{aligned}
\mathrm{jump}_{(bc,\bar{b})}(P) &= \mathrm{jump}(P)+b\cdot\mathrm{block}(P),\\
\mathrm{jump}(P) &= (1,1,1,\ldots,1)\,\mathbf{JUMP}_{(bc,\bar{b})}(P)(0,1,2,\ldots,N)^T,\\
\mathrm{block}(P) &= (0,1,2,\ldots,(c-1))\,\mathbf{JUMP}_{(bc,\bar{b})}(P)(1,1,1,\ldots,1)^T,\\
\mathrm{lucky}(P) &= (1,1,1,\ldots,1)\,\mathbf{JUMP}_{(bc,\bar{b})}(P)(1,0,0,\ldots,0)^T.
\end{aligned}$$

As a consequence, we derive the following corollary from Theorem 1.

Corollary 2 *The bijection $\varphi: F_{(bc,\bar{b})}\to PF_{(bc,\bar{b})}$ has the following property:*

$$(\mathrm{inv}_{(bc,\bar{b})},\mathrm{inv},\mathrm{tcol},\mathrm{prop},\mathrm{tree})(F)=(\mathrm{jump}_{(bc,\bar{b})},\mathrm{jump},\mathrm{block},\mathrm{lucky},\mathrm{crit})\varphi(F),$$

for $F\in F_{(bc,\bar{b})}$.

Introduce the following enumerators of $(bc,\bar{b})$-forest and $(bc,\bar{b})$-parking functions:

$$\begin{aligned}
I_n^{(bc,\bar{b})}(q,u,t) &= \sum_{F\in F_{(bc,\bar{b})}} q^{\mathrm{inv}_{(bc,\bar{b})}(F)}u^{\mathrm{prop}(F)}t^{\mathrm{tree}(F)},\\
\bar{P}_n^{(bc,\bar{b})}(q,u,t) &= \sum_{P\in PF_{(bc,\bar{b})}} q^{\mathrm{jump}_{(bc,\bar{b})}(P)}u^{\mathrm{lucky}(P)}t^{\mathrm{crit}(P)}.
\end{aligned}$$

Then we can derive a partial refinement of (1) from Corollary 2.

Corollary 3 *We have*

$$I_n^{(bc,\bar{b})}(q,u,t)=\bar{P}_n^{(bc,\bar{b})}(q,u,t).$$

Define the homogeneous polynomial

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

Theorem 4 *We have*

$$\sum_{P \in PF_{(bc,\bar{b})}} u^{\mathrm{lucky}(P)} t^{\mathrm{crit}(P)} = \sum_{F \in F_{(bc,\bar{b})}} u^{\mathrm{prop}(F)} t^{\mathrm{tree}(F)} = P_n(b, b-1+u, ct(b-1+u)). \quad (3)$$

Remark. For $b = c = 1$ and $b = t = 1$, we recover, respectively, two results of Gessel and Seo [GS06, Theorem 6.1 and Corollay 10.2].

Theorem 5 *We have the hook-length formula of $(a, \bar{b})$-forests*

$$\sum_{F \in F_{(a,\bar{b})}} c^{\mathrm{tree}(F)} \prod_v \left(1 + \frac{\alpha}{h_v}\right) = P_n(b, b(1+\alpha), ac(1+\alpha)), \quad (4)$$

where the sum is over all $(a, \bar{b})$-forests on n-set.

Remark. For $a = b = 1$ this is Gessel and Seo's hook-length formula [GS06, Corollay 6.3].

4 Proof of Theorem 1

The inverse map of the extended mapping φ can be defined like the method in the paper [Shi08]: Given a $(bc, \bar{b})$-parking function P, all cars are parked by the parking algorithm. At that time, we record the $\mathrm{jump}(P; c)$ for every car in next row. After finishing, we draw an edge between the car c and the closest car on its right which is larger than c in its same block. We get the forest-structure on the cars as vertices. That is a forest D. By defining

$$|\mathrm{inv}(F; v)| \equiv \mathrm{jump}(P; c) \mod |D_F(v)|,$$

we can recover two forests I and F. By $\bar{\kappa}(v) := \lfloor \frac{\mathrm{jump}(P;c)}{b} \rfloor$, we can recover the color of v in F where $\theta_F(v) = c$.

We can prove that φ is weight preserving by the following lemma.

Lemma 6 *There is a bijection $\varphi : F_{(bc,\bar{b})} \to PF_{(bc,\bar{b})}$ between $(bc, \bar{b})$-forests and $(bc, \bar{b})$-parking functions such that*

$$(\mathrm{inv}, \mathrm{tcol}, \mathrm{tree})(F; v) = (\mathrm{jump}, \mathrm{block}, \mathrm{crit})(\varphi(F); \theta_F(v)),$$

for all $(bc, \bar{b})$-forests F and all vertices $v \in F$.

Proof: If we use the function $d \mapsto (g+c-1-k)b$ instead of $d \mapsto (g+c-1-k)b-i$, all cars are lucky since all images of f are different. So using the original function $d \mapsto ((g+c-1-k)b-i)$, the value of $\mathrm{jump}(P : c)$ increases by $\mathrm{inv}(T : v)$ where $\theta_F(v) = c$. Thus $\mathrm{inv}(F : v) = \mathrm{jump}(\varphi(F) : \theta_F(v))$.

Suppose that $\mathrm{tcol}(F; v) = k$, which means that a vertex v is in F_k. So a label of $\theta_F(v)$ is also in D_k. Then car $\theta_F(v)$ is parked actually in a k-th block. Then $\mathrm{block}(\varphi(F); \theta_F(v)) = k$.

If a vertex v is a root of a tree in F, a parent of $\theta_F(v)$ is the root of D. So there is no car larger than the car $\theta_F(v)$ on its right in same block. Hence the car $\theta_F(v)$ is critical. □

5 Proof of Theorem 4

The first equality follows from Corollary 3 for $q = 1$, i.e.,

$$\sum_{F \in F_{(bc,\bar{b})}} u^{\mathrm{prop}(F)} t^{\mathrm{tree}(F)} = \sum_{P \in PF_{(bc,\bar{b})}} u^{\mathrm{lucky}(P)} t^{\mathrm{crit}(P)}.$$

To prove the second equality in Theorem 4, we need to appear for two Prüfer-like algorithms: the colored Prüfer code [CKSS04] and reverse Prüfer algorithm in [SS07]. Given a $(bc,\bar{b})$-forest F, deleting the largest leaves successively $v_n, \ldots, v_1$ where σ_i is the parent of v_i or $\sigma_i = -\operatorname{tcol}(F : v_i)$ if v_i is a root and the color $c_i = \bar{\kappa}(v_i)$. Then the *colored Prüfer code* of F is defined by

$$\sigma = \begin{pmatrix} \sigma_n & \sigma_{n-1} & \cdots & \sigma_1 \\ c_n & c_{n-1} & \cdots & c_1 \end{pmatrix} \in \begin{pmatrix} \{-(c-1), \ldots, n\} \\ \{0, \ldots, b-1\} \end{pmatrix}^{n-1} \times \begin{pmatrix} \{-(c-1), \ldots, 0\} \\ \{0, \ldots, b-1\} \end{pmatrix}.$$

In order to count the number of proper vertices, we define the *reverse colored Prüfer algorithm* as follows: Starting from a colored Prüfer code $\sigma = \begin{pmatrix} \sigma_n & \sigma_{n-1} & \cdots & \sigma_1 \\ c_n & c_{n-1} & \cdots & c_1 \end{pmatrix}$. Let F_1 be the forest with unlabeled single vertex v_1 by $\operatorname{tcol}(F : v_1) = -\sigma_1$. For each $i = 2, \ldots, n$, we assume that F_{i-1} is the forest obtained from the subcode $\begin{pmatrix} \sigma_{i-1} & \sigma_{i-2} & \cdots & \sigma_1 \\ & c_{i-2} & \cdots & c_1 \end{pmatrix}$. Let ℓ be the minimal element in $[n]$ which does not appear in F_{i-1}. To construct F_i from F_{i-1} and (σ_i, c_{i-1}), we should consider the following two cases.

1. Suppose that σ_i appears in F_{i-1}. Then the unlabeled vertex v in F_{i-1} is labeled by ℓ with color c_{i-1} in T_i. Since the new label ℓ is minimal among the unused labels in T_{i-1}, the vertex v with the color c_{i-1} is a proper vertex in T if and only if $c_{i-1} = 0$.

2. Suppose that σ_i does not appear in T_{i-1}. Then the unlabeled vertex v in F_{i-1} is labeled by σ_i in F_i.

   (a) If $\sigma_i \leq 0$, then the vertex v is a proper vertex in F, as in case (1) and the unlabeled vertex in F_i becomes a root in F.

   (b) If $\sigma_i = l$, then the vertex v is a proper vertex in F, as in case (1).

   (c) If $\sigma_i \neq l$, then the vertex v will have a descendant labeled by ℓ. Thus, the vertex v is not proper vertex in F.

So there are exactly $i - 1 + c$ choices of σ_i and one choice of c_{i-1} in case (1), case (2a), and case (2b), such that the newly labeled vertex v is a proper vertex in F. Because the number of i's such that $\sigma_i \leq 0$ in a colored Prüfer-code equals the number of the roots in F, $\operatorname{tree}(F)$ is enumerated by nonpositive number in the colored Prüfer-code of a forest F. Thus we have the following formula:

$$\begin{aligned}\sum_{F\in F_{(bc,\bar{b})}} u^{\mathrm{prop}(F)} t^{\mathrm{tree}(F)} &= ct && \text{by } \sigma_1 \in \{0,-1,\dots,-(c-1)\}\\ &\times \prod_{i=2}^{n} (b(n-i+1)+(i-1+ct)(b-1+u)) && \text{by } (\sigma_i, c_{i-1})\\ &\times (b-1+u) && \text{by } c_{n-1}\\ &= P_n(b, b-1+u, ct(b-1+u)).\end{aligned}$$

This completes the bijective proof of equation (3).

6 Proof of Theorem 5

By Theorem 4, the right side of (4) is

$$\sum_{F\in F_{(b,\bar{b})}} (1+b\alpha)^{\mathrm{prop}(F)} (\frac{ac}{b})^{\mathrm{tree}(F)}.$$

Replacing α by α/b in (4), it suffices to prove the identity:

$$\sum_{F\in F_{(a,\bar{b})}} c^{\mathrm{tree}(F)} \prod_{v} \left(1+\frac{\alpha}{bh_v}\right) = \sum_{F\in F_{(b,\bar{b})}} (1+\alpha)^{\mathrm{prop}(F)} (\frac{ac}{b})^{\mathrm{tree}(F)}. \tag{5}$$

We follow Gessel and Seo's proof [GS06] in the case of $a = b = 1$. For each (unlabeled) forest $\tilde{F}$ on n sets, a *labeling* of $\tilde{F}$ is a bijection from $V(\tilde{F})$ to $[n]$ and $(a,\bar{b})$-*coloring* κ is a mapping from $V(\tilde{F})$ to nonnegative numbers such that $\kappa(v) < a$ if v is a root and $\kappa(v) < b$ otherwise. Define the set of $(a,\bar{b})$-forests

$$L_{(a,\bar{b})}(\tilde{F}) = \left\{(L,\kappa) : L \text{ is a labeling and } \kappa \text{ is a } (a,\bar{b})\text{-coloring of } \tilde{F}\right\}.$$

Lemma 7 *Let $\tilde{F}$ be a (unlabeled) forest with n vertices. If S is a subset of $V(\tilde{F})$, then the number of labelings $L \in L_{(b,\bar{b})}(\tilde{F})$ such that every vertex in S is a proper vertex is*

$$\frac{n!b^n}{\prod_{v\in S}(bh_v)}. \tag{6}$$

Proof: Clearly the cardinality of $L_{(b,\bar{b})}(\tilde{F})$ is $n!b^n$. Among the elements of $L_{(b,\bar{b})}(\tilde{F})$, the probability that some vertex $v \in S$ is a proper vertex equals $\frac{1}{bh_v}$. In other words, the number of labelings $L \in L_{(b,\bar{b})}(\tilde{F})$ such that every vertex in S is a proper vertex is $\frac{1}{bh_v}$ times the number of labelings in which every vertex in $S \setminus \{v\}$ is a proper vertex. By induction on $|S|$, we are done. □

Let us consider the formula

$$\sum_{L\in L_{(b,\bar{b})}(\tilde{F})}(1+\alpha)^{\mathrm{prop}(L)}(\frac{ac}{b})^{\mathrm{tree}(L)}=\sum_{L\in L_{(b,\bar{b})}(\tilde{F})}\sum_{S}\alpha^{|S|}(\frac{ac}{b})^{\mathrm{tree}(L)},$$

where S runs over the subsets of the set of proper vertices of L. Reversing the order of two summations, it follows by Lemma 7 that

$$\begin{aligned}\sum_{S\subset V(\tilde{F})}(\frac{ac}{b})^{\mathrm{tree}(\tilde{F})}\sum_{L}\alpha^{|S|} &= \sum_{S\subset V(\tilde{F})}(\frac{ac}{b})^{\mathrm{tree}(\tilde{F})}\frac{n!b^n}{\prod_{v\in S}(bh_v)}\alpha^{|S|}\\ &= n!b^n(\frac{ac}{b})^{\mathrm{tree}(\tilde{F})}\prod_{v\in V(\tilde{F})}(1+\frac{\alpha}{bh_v}),\end{aligned}$$

where $L\in L_{(b,\bar{b})}(\tilde{F})$ such that every vertex in S is a proper vertex. Therefore,

$$\sum_{L\in L_{(b,\bar{b})}(\tilde{F})}(1+\alpha)^{\mathrm{prop}(L)}(\frac{ac}{b})^{\mathrm{tree}(L)}=n!b^n(\frac{ac}{b})^{\mathrm{tree}(\tilde{F})}\prod_{v\in V(\tilde{F})}(1+\frac{\alpha}{bh_v}) \tag{7}$$

Let us say that two labelings with colorings of a forest $\tilde{F}$ are *equivalent* if there is an automorphism of $\tilde{F}$ that takes one labeling with coloring to the other. Let $\tilde{F}$ be a forest on n set with automorphism group G. Then the $n!b^{n-\mathrm{tree}(\tilde{F})}a^{\mathrm{tree}(\tilde{F})}$ labelings with colorings of F fall into $n!b^{n-\mathrm{tree}(\tilde{F})}a^{\mathrm{tree}(\tilde{F})}/\,|G|$ equivalence classes. Define

$$\tilde{L}_{(a,\bar{b})}(\tilde{F})=\left\{L\in F_{(a,\bar{b})}:\text{The underlying graph of } L \text{ is } \tilde{F}\right\}.$$

Clearly $\left|\tilde{L}_{(a,\bar{b})}(\tilde{F})\right| = n!b^{n-\mathrm{tree}(\tilde{F})}a^{\mathrm{tree}(\tilde{F})}/\,|G|$ and equivalent labelings with coloring have the same number of proper vertices of trees, dividing (7) by $|G|$, so we obtain the following.

$$\sum_{L\in\tilde{L}_{(b,\bar{b})}(\tilde{F})}(1+\alpha)^{\mathrm{prop}(L)}(\frac{ac}{b})^{\mathrm{tree}(L)}=\left|\tilde{L}_{(a,\bar{b})}(\tilde{F})\right|c^{\mathrm{tree}(\tilde{F})}\prod_{v\in V(\tilde{F})}(1+\frac{\alpha}{bh_v}).$$

Summing over all (unlabeled) forests $\tilde{F}$ yields

$$\sum_{\tilde{F}}\sum_{L\in\tilde{L}_{(b,\bar{b})}(\tilde{F})}(1+\alpha)^{\mathrm{prop}(L)}(\frac{ac}{b})^{\mathrm{tree}(L)}=\sum_{\tilde{F}}\left|\tilde{L}_{(a,\bar{b})}(\tilde{F})\right|c^{\mathrm{tree}(\tilde{F})}\prod_{v\in V(\tilde{F})}(1+\frac{\alpha}{bh_v}).$$

As $F_{(a,\bar{b})}=\bigcup_{\tilde{F}}\tilde{L}_{(a,\bar{b})}(\tilde{F})$, we obtain (5).

7 Concluding Remarks

In this paper, we give a bijective proof of (1) in the $(bc,\bar{b})$ case. The problem of giving a bijective proof of (1) in the general $(a,\bar{b})$ case is still open. It seems that the construct of such a bijection in the $(1,\bar{b})$ case is crucial.

Acknowledgement

This work is supported by la Région Rhône-Alpes through the program "MIRA Recherche 2008", project 08 034147 01.

References

[CKSS04] Manwon Cho, Dongsu Kim, Seunghyun Seo, and Heesung Shin, *Colored Prüfer codes for k-edge colored trees*, Electron. J. Combin. **11** (2004), no. 1, Note 10, 7 pp. (electronic).

[EFL05] Sen-Peng Eu, Tung-Shan Fu, and Chun-Ju Lai, *On the enumeration of parking functions by leading terms*, Adv. in Appl. Math. **35** (2005), no. 4, 392–406.

[FR74] Dominique Foata and John Riordan, *Mappings of acyclic and parking functions*, Aequationes Math. **10** (1974), 10–22.

[GS06] Ira M. Gessel and Seunghyun Seo, *A refinement of Cayley's formula for trees*, Electron. J. Combin. **11** (2004/06), no. 2, Research Paper 27, 23 pp. (electronic).

[Kre80] G. Kreweras, *Une famille de polynômes ayant plusieurs propriétés énumeratives*, Period. Math. Hungar. **11** (1980), no. 4, 309–320.

[KY03] Joseph P. S. Kung and Catherine Yan, *Gončarov polynomials and parking functions*, J. Combin. Theory Ser. A **102** (2003), no. 1, 16–37.

[MR68] C. L. Mallows and John Riordan, *The inversion enumerator for labeled trees*, Bull. Amer. Math. Soc. **74** (1968), 92–94.

[Shi08] Heesung Shin, *A new bijection between forests and parking functions*, arXiv:0810.0427.

[SP02] Richard P. Stanley and Jim Pitman, *A polytope related to empirical distributions, plane trees, parking functions, and the associahedron*, Discrete Comput. Geom. **27** (2002), no. 4, 603–634.

[SS07] Seunghyun Seo and Heesung Shin, *A generalized enumeration of labeled trees and reverse Prüfer algorithm*, J. Combin. Theory Ser. A **114** (2007), no. 7, 1357–1361.

[Sta99] Richard P. Stanley, *Enumerative combinatorics. Vol. 2*, Cambridge Studies in Advanced Mathematics, vol. 62, Cambridge University Press, Cambridge, 1999, With a foreword by Gian-Carlo Rota and appendix 1 by Sergey Fomin.

[Yan01] Catherine H. Yan, *Generalized parking functions, tree inversions, and multicolored graphs*, Adv. in Appl. Math. **27** (2001), no. 2-3, 641–670, Special issue in honor of Dominique Foata's 65th birthday (Philadelphia, PA, 2000).