Séminaire Lotharingien de Combinatoire **78B** (2017)
Article #73, 12 pp.

Proceedings of the 29th Conference on Formal Power Series and Algebraic Combinatorics (London)

Schur positivity and labeled binary trees

Ira M. Gessel[1], Sean Griffin[*2] and Vasu Tewari[†2]

[1]*Department of Mathematics, Brandeis University, Waltham, MA 02453*
[2]*Department of Mathematics, University of Washington, Seattle, WA 98195*

Abstract. The first author introduced a multivariate generating function that tracks the distribution of ascents and descents on labeled plane binary trees and conjectured that it was Schur positive. In this article, we give a sketch for a proof of the stronger statement that the generating function restricted to trees with a fixed canopy is Schur positive. Central to our approach is a weighted extension of a bijection of Préville-Ratelle and Viennot relating pairs of paths and binary trees. We apply our results to construct a $\mathfrak{S}_n$-action on the regions of the Linial arrangement using a bijection of Bernardi. We then establish the γ-positivity for the distribution of right descents over local binary search trees.

Keywords: canopy, labeled trees, Schur positivity, gamma-positivity

1 Introduction

The study of permutation statistics is a classical theme in algebraic combinatorics. One statistic which has shown up in surprisingly many areas of mathematics is the descent statistic. In this article, we study a natural analogue of the descent statistic in the setting of labeled plane binary trees. Our results demonstrate that much more remains to be done in this regard, hints of which we provide in Section 5.

Given a positive integer n, let $\mathcal{T}_n^{\ell}$ ($\mathcal{T}_n$) denote the set of labeled (respectively unlabelled) plane binary trees on n nodes. The labels on the nodes are drawn from the set of positive integers $\mathbb{P}$, allowing repeated labels. All trees referred to in this extended abstract will be plane binary trees, where we distinguish left children from right children. The ascent and descent statistics on labeled trees come in two flavors each, depending on whether one compares the label of the parent node to the label of its right child or the label of its left child. The first author, in the 1990s, initiated their study, and established a functional equation for the generating function keeping track of these statistics over the set of standard labeled trees. A *standard labeled tree* $T \in \mathcal{T}_n^{\ell}$ is a labeled tree with distinct labels drawn from $[n] := \{1, \ldots, n\}$. Given a labeled tree T, let the *weight* of T be

$$\mathrm{wt}(T) = a_1^{\mathrm{rasc}(T)} a_2^{\mathrm{rdes}(T)} b_1^{\mathrm{lasc}(T)} b_2^{\mathrm{ldes}(T)}, \tag{1.1}$$

*stgriff@uw.edu. Partially supported by the Achievement Rewards for College Scientists Foundation and N.S.F. Grant 1101017.

†Supported by an A.M.S. Simons travel grant.

where lasc, ldes, rasc and rdes record the number of ascents and descents in the labeling to the left and right, such that rasc and ldes are given by weak inequalities, while rdes and lasc are given by strict inequalities. For example, a node labeled 2 with a right child labeled 4 is considered a right ascent in the labeling. We will say that the edge between these two nodes has weight a_1. This given, consider the following generating function

$$B := B(x) = \sum_{n\geqslant 1} \sum_{\text{standard } T\in\mathcal{T}_n^\ell} \mathrm{wt}(T)\frac{x^n}{n!}. \tag{1.2}$$

In unpublished work, the first author established that B satisfies an elegant functional equation that sheds light on the symmetries present in B and generalizes the well-known formula for the exponential generating function whose coefficients are the Eulerian polynomials. Subsequently, proofs of this functional equation were also given by Kalikow [8] and Drake [5]. Recent work involving the generating function B was inspired by the first author's observation that certain evaluations of B coincided with the number of regions in well-known deformations of Coxeter arrangements. This viewpoint has been pursued in [4, 6, 12], and a complete explanation has been offered by Bernardi [2].

Our primary object of study is a multivariate generalization of B. To this end, let $\mathsf{x} = \{x_1, x_2, \ldots\}$ be a commuting set of indeterminates. To every $T \in \mathcal{T}_n^\ell$, we associate a monomial x_T as follows: For a vertex $v \in T$ labeled i, let $x_v := x_i$, and let $\mathsf{x}_T = \prod_{v\in T} x_v$. Consider the following power series in x with coefficients in the semiring $\mathbb{N}[a_1, a_2, b_1, b_2]$ of polynomials on $\{a_1, a_2, b_1, b_2\}$ with nonnegative integer coefficients.

$$S := S(\mathsf{x}) = \sum_{n\geqslant 1} \sum_{T\in\mathcal{T}_n^\ell} \mathrm{wt}(T)\mathsf{x}_T. \tag{1.3}$$

For $n \geqslant 1$, let $G_n := G_n(\mathsf{x})$ denote the degree n homogeneous summand of S only summing over labeled trees on n nodes. Clearly, $S = \sum_{n\geqslant 1} G_n$. The first author established a functional equation that revealed that S is a symmetric function in the x_i variables and then made the following conjecture based on empirical evidence.

Conjecture 1.1. *S is Schur positive, meaning that S may be expressed as a sum of Schur functions s_λ with coefficients in $\mathbb{N}[a_1, a_2, b_1, b_2]$.*

This conjecture served as the primary motivation for the line of research outlined in this extended abstract. In fact, we show a stronger fact; that S may be expressed as a sum of Schur functions with coefficients in $\mathbb{N}[a_1b_2, a_2b_1, a_1 + b_2, a_2 + b_1]$. This theorem follows from a recursive formula satisfied by S, which is (1.4) below. In this extended abstract, we introduce the algorithm `ExtendedPG`, which is a weighted extension of the Préville-Ratelle-Viennot bijection between certain pairs of paths and binary trees and describe how it can be used to decompose a labeled binary tree into subtrees in order to prove Theorem 1.2.

Theorem 1.2. *Let r_α denote the ribbon Schur function indexed by the composition α and let $\ell(\alpha)$ denote the length of α. Then we have that*

$$S = \sum_{n \geqslant 1} \sum_{\alpha \vDash n} (a_1 b_2 S + a_1 + b_2)^{n-\ell(\alpha)} (a_2 b_1 S + a_2 + b_1)^{\ell(\alpha)-1} r_\alpha. \tag{1.4}$$

The *canopy* of a binary tree $T \in \mathcal{T}_n$ is a certain word of length $n-1$ on the alphabet $\{U, D\}$. See the end of Section 2 for its precise definition. We can then consider the refinement of S only summing over labeled trees with a fixed canopy v,

$$G_{n,v} := G_{n,v}(\mathsf{x}) = \sum_{T \in \mathcal{T}^\ell_{n,v}} \mathrm{wt}(T) \mathsf{x}_T, \tag{1.5}$$

where $\mathcal{T}^\ell_{n,v} \subseteq \mathcal{T}^\ell_n$ is the set of labeled trees with canopy v. We will describe in Section 3 how `ExtendedPG` also provides a proof that each $G_{n,v}$ is Schur positive, which is a refinement of Theorem 1.2.

Theorem 1.3. *Fix $n \in \mathbb{P}$, and let v be a word of length $n-1$ in the alphabet $\{U, D\}$. Then $G_{n,v}$ is Schur positive.*

Our proof also yields an explicit Schur positive expansion of $G_{n,v}$ and consequently yields a Schur positive expansion of G_n. We state the expansion of G_n in Corollary 1.4 but omit its proof. We also omit the expansion of $G_{n,v}$.

First, let us call a binary tree *right-leaning* if every node which has a left child also has a right child. Let the set of right-leaning binary trees on n nodes be denoted by $\mathcal{RT}_n$. Furthermore, let $\dot{\mathcal{RT}}_n$ denote the set of right-leaning trees on n nodes with some subset of internal nodes marked. As a Corollary to Theorem 1.2, we have the following explicit expansion for G_n in terms of right-leaning trees.

Corollary 1.4. *Let $\mathrm{Bi}^u(\dot{T})$ ($\mathrm{Bi}^m(\dot{T})$) denote the number of unmarked (respectively marked) internal nodes with two children, and let $\mathrm{Uni}^u(\dot{T})$ ($\mathrm{Uni}^m(\dot{T})$) denote the number of unmarked (respectively marked) internal nodes with only one child (which has to be a right child given the definition of $\dot{\mathcal{RT}}_n$). Then G_n has the expansion given in* (1.6).

$$G_n = \sum_{\dot{T} \in \dot{\mathcal{RT}}_n} (a_1 b_2)^{\mathrm{Bi}^u(\dot{T})} (a_2 b_1)^{\mathrm{Bi}^m(\dot{T})} (a_1 + b_2)^{\mathrm{Uni}^u(\dot{T})} (a_2 + b_1)^{\mathrm{Uni}^m(\dot{T})} \sum_{\check{\mathbf{c}}(\dot{T}) \preccurlyeq \delta \preccurlyeq \hat{\mathbf{c}}(\dot{T})} r_\delta, \tag{1.6}$$

where $\check{\mathbf{c}}(\dot{T})$ and $\hat{\mathbf{c}}(\dot{T})$ are certain compositions of n associated to a marked tree whose definitions we omit from this extended abstract, and $\preccurlyeq$ stands for the usual refinement order on compositions. A fact worth noting about (1.6) is that the coefficient of r_α for every $\alpha \vDash n$ belongs to $\mathbb{N}[a_1 b_2, a_2 b_1, a_1 + b_2, a_2 + b_1]$ and furthermore, evaluates to the

Catalan number $\frac{1}{n+1}\binom{2n}{n}$ when we set $a_1 = a_2 = b_1 = b_2 = 1$. In particular, the coefficients of both $r_{(1,1,\dots,1)}$ and $r_{(n)}$ in (1.6) are the *homogenized Narayana polynomials*. Finally, the cardinalities of the sets $\dot{\mathcal{RT}}_n$ are the *large Schröder numbers*.

We conclude the introduction with an outline of the article. In Section 2, we introduce our main combinatorial objects and develop all the notation we need. In Section 3, we state the algorithm `ExtendedPG` and describe how it can be used to prove Theorems 1.2 and 1.3. In Section 4, we discuss some consequences of our results with a special emphasis on local binary search trees. Finally, we conclude with further avenues in Section 5.

2 Combinatorial preliminaries

In this section, we will introduce the main combinatorial objects of this article. Let $\mathbb{P}$ be the set of positive integers. Let $\mathbb{P}^+$ be the set of nonempty words on $\mathbb{P}$. Let $w = w_1w_2\dots w_n \in \mathbb{P}^+$ be a nonempty word. An *inversion* of w is a pair (i,j) with $1 \leqslant i < j \leqslant n$ such that $w_i > w_j$. The set of inversions of w is its *inversion set*. Let the *standardization* of w be the unique permutation $\mathrm{std}(w) \in \mathfrak{S}_n$ whose inversion set is the same as that of w. To the word w, we associate the monomial $\mathsf{x}_w = x_{w_1}x_{w_2}\dots x_{w_n}$.

For notions related to the algebra of symmetric functions, denoted by Sym, that are not made explicit here, we refer the reader to [11]. An important class of symmetric functions for us is that of *ribbon Schur functions*, which are indexed by compositions. To each composition $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_k)$ of n, we associate a subset of $[n-1]$ defined by $\mathrm{set}(\alpha) := \{\alpha_1, \alpha_1+\alpha_2, \dots, \alpha_1+\cdots+\alpha_{k-1}\} \subset [n-1]$. The ribbon Schur function associated to α is defined as

$$r_\alpha := r_\alpha(\mathsf{x}) = \sum x_{i_1}x_{i_2}\dots x_{i_n}, \tag{2.1}$$

where the sum is over all n-tuples $(i_1, i_2, \dots, i_n)$ of positive integers satisfying $i_j > i_{j+1}$ if and only if $j \in \mathrm{set}(\alpha)$ for $1 \leqslant j \leqslant n-1$.

Note that the nodes of a binary tree can be categorized into *leaf nodes*, which are nodes with no children, and *internal nodes*, which are nodes with at least one child. We can classify the internal nodes further. We refer to an internal node with two children as a *bivalent node*, and a node with only one child as a *univalent node*. Given a binary tree T, the sets of univalent nodes and bivalent nodes will be denoted by $\mathrm{Bi}(T)$ and $\mathrm{Uni}(T)$ respectively.

We will work with two different types of orderings on the nodes of a binary tree T derived from a traversal of its nodes. The *preorder traversal* is defined recursively, where we first record the root, then traverse the right subtree of T in preorder, and finally the left subtree of T in preorder. The *inorder traversal* is defined similarly, where we first traverse the left subtree of T recursively in inorder, then record the root of T, and finally traverse the right subtree of T in inorder. Given a node u in a labeled tree T, we refer to the label on u as u^ℓ. The preorder and inorder traversals naturally associate a reading

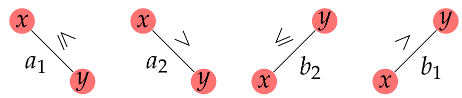

Figure 1: Determining the weight of a labeled edge.

word with a labeled tree, denoted pre(T) and in(T), respectively. Breaking pre(T) at the leaf nodes allows us to naturally associate a composition $\mathbf{c}(T)$ to T.

To a plane binary tree $T \in \mathcal{T}_n$ we can associate a word of length $n-1$ on the alphabet $\{U, D\}$ via the following process. Traverse the tree with inorder traversal. For each node traversed, record a D if it is missing a left child, and then record a U if it is missing a right child. We obtain a word of length $n+1$ on the alphabet $\{U, D\}$ whose first letter is D and last letter is U, and hence we may omit these letters. The resulting word of length $n-1$ is called the *canopy* of the tree, and will be denoted by Can(T).

3 A weighted extension of the Push-Gliding algorithm

In this section, we provide a sketch of a proof of Theorem 1.2 using a weighted variant of the Push-Gliding algorithm of Préville-Ratelle and Viennot [10]. The main result in [10] is a generalization of the m-Tamari lattices to posets of arbitrary paths. Préville-Ratelle and Viennot were motivated by a question arising from what has come to be known as 'rational Catalan combinatorics'.

We will define a *path* to be a sequence of nodes situated in the plane joined by up steps $U = (1, 1)$ and down steps $D = (1, -1)$. Let (u, v) be a pair of paths in the plane starting at $(0, 0)$. We shall call such a pair *glued* if u and v end at the same point and u stays weakly above v. An equivalent variation of the Push-Gliding algorithm in [10] takes as input a glued pair of paths (u, v) and outputs a unique binary tree $T(u, v)$. We extend the algorithm to labeled paths and trees, keeping track of edge weights.

First let us establish some terminology used in our algorithms. Given a plane binary tree T and a node y in T, let T_y be the subtree of T rooted at y. The *right branch* of T is the sequence of nodes in T starting at the root and going right. A *tail* τ attached to the tree T is a path whose first node is on the right branch of T and whose first step is a U. Let τ_i denote the i^{th} node of τ. A *labeled tail* of T is a tail attached to the tree T whose nodes are labeled with positive integers. In our extended algorithm, at every step we keep track of a labeled binary tree T and a labeled tail τ attached to T. The weight associated to an edge, whether it is part of a tree or part of the tail, is indicated in Figure 1 based on the orientation of the edge, and the weight of a labeled tree or tail is the product of its edge weights.

We first introduce the operations `WGlide` and `WPush`, illustrated in Figures 2 and 3, and then state our main algorithm `ExtendedPG`. Each Glide and Push operations may change the weight $\mathrm{wt}(T,\tau) := \mathrm{wt}(T)\mathrm{wt}(\tau)$, but these changes will only involve the nodes x, y and z indicated in Figures 2 and 3. It can be checked that before and after these operations, the weights of the two bold edges only depend on the standardization of the word read from their labels, $\mathrm{std}(x^\ell y^\ell z^\ell)$. Furthermore, all other edge weights are preserved. The following claim allow us to track $\mathrm{wt}(T,\tau)$.

Claim 3.1. *In Cases* $(\mathcal{P}1)$ *and* $(\mathcal{G}1)$ *of* **??** *1***??** *2,* $\mathrm{wt}(T,\tau) = \mathrm{wt}(T',\tau')$. *In Cases* $(\mathcal{P}2)$ *and* $(\mathcal{G}2)$*, the combined weight of the edges xz and yz in* T' *is* a_1b_2 *if* $x^\ell \leqslant z^\ell$ *or* a_2b_1 *if* $x^\ell > z^\ell$.

Algorithm 1: `WGlide`

Input: (T,τ,I): $T \in \mathcal{T}^\ell$, τ a tail attached to T, I a set of nodes in T.
`// Assume` $\tau_1 \neq \mathrm{root}(T)$.
Output: (T',τ',I'): $T' \in \mathcal{T}^\ell$, τ' a tail attached to T', I' a set of nodes in T'.
begin
  Let $y := \tau_1$, and let $z := \tau_2$;
  Let x be the parent of y in T;
  Let τ' be obtained from τ by replacing the starting node y with x;
  if $\mathrm{std}(x^\ell y^\ell z^\ell) \in \{123, 213, 231\}$ **then return** (T,τ',I) `// Case` $(\mathcal{G}1)$;
  else return $(T,\tau',I \cup T_y)$ `// Case` $(\mathcal{G}2)$;
end

Algorithm 2: `WPush`

Input: (T, τ, I): $T \in \mathcal{T}^{\ell}$, τ a labeled tail attached to T, I a set of nodes in T.
Output: (T', τ', I'): $T' \in \mathcal{T}^{\ell}$, τ' a labeled tail attached to T', I' a set of nodes in T'.
begin
  Let $y := \tau_1$, and let $z := \tau_2$;
  Let $(z, \tau_3, \tau_4, \ldots, \tau_k)$ is the longest segment of τ starting at z going down, let $\rho = (\tau_3, \tau_4, \ldots, \tau_k)$, considered as a (possibly empty) tree with only right children;
  Let T'' be the tree with root z, left subtree T_y and right subtree ρ;
  Let T' be the tree obtained from T by replacing T_y with T'';
  Let τ' be the final segment of τ starting at τ_k;
  if *y has a parent x in T* **then**
    if $\mathrm{std}(x^{\ell} y^{\ell} z^{\ell}) \in \{132, 312, 321\}$ **then return** (T', τ', I) `// Case` $(\mathcal{P}1)$;
    else return $(T', \tau', I \cup T_y)$ `// Case` $(\mathcal{P}2)$;
  else
    return (T', τ', I)
  end
end

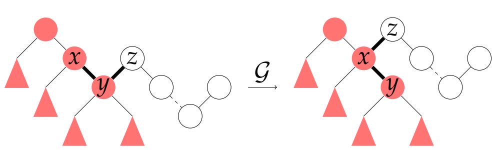

Figure 2: Glide Operation

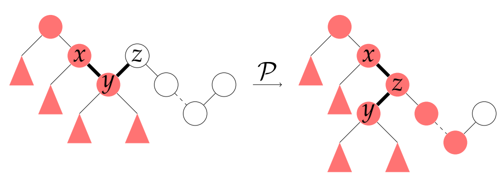

Figure 3: Push Operation

Préville-Ratelle and Viennot have shown the original Push-Gliding algorithm involving unlabeled paths and trees is a bijection such that $\mathrm{Can}(T(u, v)) = v$. An example of **??** 3 is demonstrated in Figure 4. Uninfected tree nodes are shaded dark red, infected tree nodes are shaded light green, and path nodes are circled. Given an input (u, v) for **??** 3, let W be the word read off from the labeling of v in order. Note that $W = \mathrm{in}(T(u, v))$. If I is the set of infected nodes which **??** 3 returns, let $w = w_1 w_2 \ldots w_{\ell}$ be the subword of W of the labels of the nodes not in I.

Claim 3.2. *The set I is the disjoint union of the maximal infected subtrees of $T(u, v)$. The inorder reading words of these trees form nonadjacent intervals in W.*

Assuming the above claim, then there exist subtrees T_1, T_2,..., T_{m-1} of the final output tree T such that for each i, T_i is either empty or a maximally infected subtree of T, and such that $W = w_1 \,\mathrm{in}(T_1) w_2 \cdots w_{\ell-1} \,\mathrm{in}(T_{\ell-1}) w_{\ell}$.

Algorithm 3: `ExtendedPG`

Input: (u, v): A glued pair of paths with a labeling of v.
Output: $(T(u, v), I)$: $T(u, v) \in \mathcal{T}^{\ell}$, a set I of "infected" nodes of $T(u, v)$.
begin
  if *v consists of only down steps* **then**
    return $(v, \varnothing)$ `// Consider` v `a tree with only right children;`
  end
  Translate u into a word u' on the alphabet $\{\mathcal{P}, \mathcal{G}\}$ by replacing each U with a $\mathcal{P}$ and each D with a $\mathcal{G}$;
  Let T be the longest initial segment of v consisting of D steps;
  Let τ be the final segment of v starting at the rightmost child of T;
  $I \longleftarrow \varnothing$;
  while *τ is not a single node* **do**
    if $u'_1 == \mathcal{P}$ **then**
      $(T, \tau, I) \longleftarrow$ `WPush`(T, τ, I);
      Remove $\mathcal{P}$ from the front of u';
    else
      $(T, \tau, I) \longleftarrow$ `WGlide`(T, τ, I);
      Remove $\mathcal{G}$ from the front of u';
    end
  end
  return (T, I)
end

In the example in Figure 4, $w = 213455$, T_1 consists of a single node labeled 1, T_4 is the green subtree consisting of three nodes labeled 342 in inorder, and $T_2 = T_3 = T_5 = \varnothing$. For i such that $T_i = \varnothing$, then w_i and w_{i+1} are adjacent nodes in v. If these nodes form an up step, let us denote this by $w_i w_{i+1} = U$ and similarly for a down step. Define

$$\mathrm{wt}_i = \begin{cases} a_1[w_i w_{i+1} = D] + b_2[w_i w_{i+1} = U] & \text{if } T_i = \varnothing \text{ and } w_i \leqslant w_{i+1}, \\ a_2[w_i w_{i+1} = D] + b_1[w_i w_{i+1} = U] & \text{if } T_i = \varnothing \text{ and } w_i > w_{i+1}, \\ a_1 b_2 & \text{if } T_i \neq \varnothing \text{ and } w_i \leqslant w_{i+1}, \\ a_2 b_1 & \text{if } T_i \neq \varnothing \text{ and } w_i > w_{i+1}. \end{cases}$$

where if P is a statement, $[P]$ evaluates to 1 if true and 0 if false. Therefore, the sequence of weights wt_i in our example is $(a_2b_1, a_1, b_2, a_1b_2, a_1)$. It can be seen that $x_w = x_2x_1x_3x_4x_5^2$ is a term in $r_{(1,5)}$, and the decomposition corresponds to a term in the expansion of $(a_2b_1S)(a_1)(b_2)(a_1b_2S)(a_1)r_{(1,5)}$, which is part of the sum in (1.4).

Given a labeled binary tree T, it corresponds to a unique glued pair of paths (u, v)

where v is labeled with the word $\mathrm{in}(T)$. Performing **??** 3 on this pair yields a decomposition of T, and the right-hand side of (1.4) can be interpreted in terms of these decompositions, as demonstrated in our example. In fact, given a decomposition of a tree $T \in \mathcal{T}_n^\ell$ into the subtrees T_i, the word w, and the weights wt_i, the canopy $\mathrm{Can}(T)$ can be recovered from this data. If to each subtree $T_i = \varnothing$ we associate the U or D step given by wt_i and to each $T_i \neq \varnothing$ we associate $D\mathrm{Can}(T_i)U$, then $\mathrm{Can}(T)$ is the sequence of steps given by concatenating these sequences in order. From these observations, it is possible to prove Theorem 1.3.

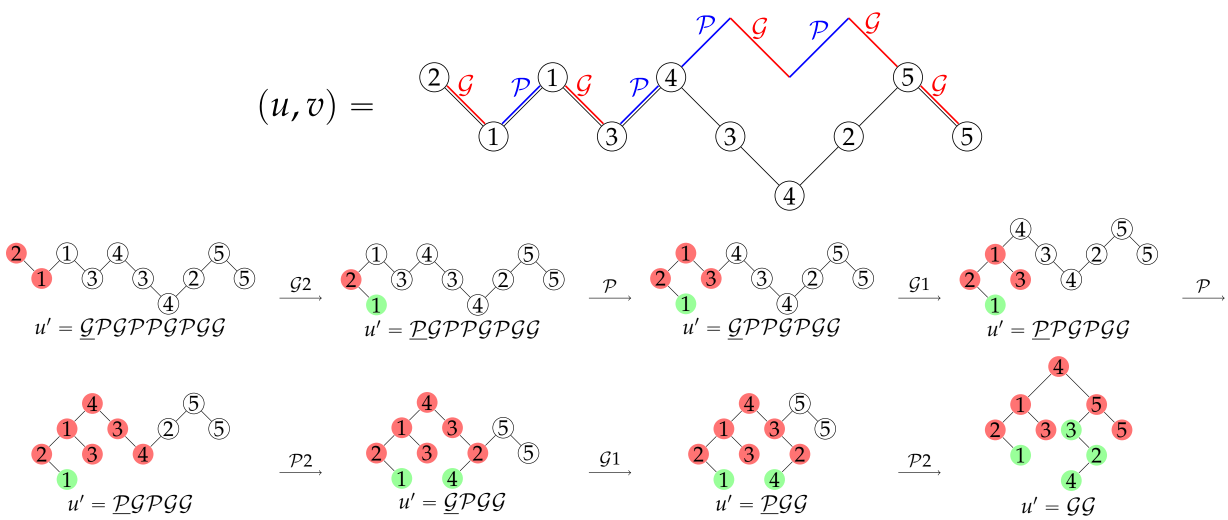

Figure 4: `ExtendedPG` performed on a glued pair (u, v).

4 Linial arrangements and local binary search trees

We will now consider the case where $a_2 = b_1 = 0$. In the setting outlined in the introduction, this corresponds to considering labeled trees such that they have only weak right ascents and weak left descents. We will refer to such labeled trees as *local binary search trees* (henceforth *LBS trees*) after Stanley. The problem of enumerating standard LBS trees was first considered by Postnikov [9], wherein he showed that these were equinumerous with intransitive trees (or alternating trees). Postnikov's interest was in enumerating the number of regions in a well-known deformation of the Coxeter arrangement of type A, the *Linial arrangement*, which, for a given $n \in \mathbb{P}$, is defined to be the arrangement of hyperplanes $x_i - x_j = 1$ for $1 \leqslant i < j \leqslant n$ in $\mathbb{R}^n$. We will denote this arrangement by $\mathcal{L}_n$. We will construct another subset of the set of labeled trees that is equinumerous with LBS trees, and use it to define an action of the symmetric group on the regions of the Linial arrangement. It is worth emphasizing that the same subset of trees has been considered by Bernardi [2] to solve the problem of finding a bijection between Linial regions and LBS trees. Furthermore, recall that the analogous question of constructing an $\mathfrak{S}_n$-action on the regions of the Shi arrangement is well-studied and gives rise to the classical parking function representation. Our $\mathfrak{S}_n$-action shares a key feature with the parking function representation, that of h-positivity. We describe the details next.

Corollary 1.4 in the case under consideration yields

$$G_n = \sum_{T\in\mathcal{RT}_n} (a_1b_2)^{|\mathrm{Bi}(T)|}(a_1+b_2)^{|\mathrm{Uni}(T)|}h_{\mathbf{c}(T)} = \sum_{T\in\mathcal{T}_n} a_1^{\mathrm{r}(T)}b_2^{\ell(T)}h_{\mathbf{c}(T)}. \tag{4.1}$$

where the second equality follows from the Foata-Strehl action on trees. Let $\mathrm{r}(T)$ and $\ell(T)$ denote the number of right edges and left edges in T, respectively. Since $\mathrm{r}(T)+\ell(T)=n-1$ for any $T\in\mathcal{T}_n$, we can set $b_2=1$. We are now ready to describe our $\mathfrak{S}_n$-module whose graded Frobenius characteristic is given by G_n. A *B-tree* is defined to be a standard labeled binary tree satisfying the condition that every internal node has a label that is less than the label of its right child, provided it exists. Otherwise, it is less than the label of its left child.

Let $\mathcal{T}_n^B$ denote the set of B-trees on n nodes. Our arguments reveal that the number of standard LBS trees on n nodes is equal to the cardinality of $\mathcal{T}_n^B$. Now note that if $T\in\mathcal{T}_n^B$ and $\mathrm{pre}(T)$ is broken into subwords precisely at the terminal nodes, then each subword is increasing. This allows us to define an obvious $\mathfrak{S}_n$-action on $\mathcal{T}_n^B$. Given $\sigma\in\mathfrak{S}_n$, we relabel the node labeled i with $\sigma(i)$ and then sort the labels on every subword that contributes to $\mathrm{pre}(T)$ from before. This ensures that the resulting tree is still a B-tree.

Let $\mathbb{C}\mathcal{T}_n^B$ denote the vector space generated by formal linear combinations of trees in $\mathcal{T}_n^B$. Then clearly $\mathbb{C}\mathcal{T}_n^B$ is a $\mathfrak{S}_n$-module graded by the number of right edges.

Theorem 4.1. *The graded Frobenius characteristic of the $\mathfrak{S}_n$-module $\mathbb{C}\mathcal{T}_n^B$ is given by*

$$\sum_{T\in\mathcal{T}_n} a_1^{\mathrm{r}(T)}h_{\mathbf{c}(T)}.$$

Using Bernardi's bijection between $\mathcal{T}_n^B$ and regions of $\mathcal{L}_n$, our action can be lifted to an action of $\mathfrak{S}_n$ on the regions of $\mathcal{L}_n$.

The previous theorem gives a new formula for the number of regions in $\mathcal{L}_n$ as $|\mathcal{T}_n^B| = \sum_{T\in\mathcal{T}_n}\binom{n}{\mathbf{c}(T)}$, where for any composition $\alpha=(\alpha_1,\ldots,\alpha_k)\vDash n$ we let $\binom{n}{\alpha} := \frac{n!}{\alpha_1!\cdots\alpha_k!}$.

We now turn our discussion to another notion of importance both in algebraic combinatorics and discrete geometry, that of γ-positivity. Postnikov [9] proved a functional equation for the generating function for the distribution of right vertices over the set of intransitive trees. That is, he considered the polynomial $f_n(x)=\sum_{k\geqslant 1} f_{nk}x^k$ where f_{nk} equals the number of intransitive trees on $[n+1]$ with k right vertices. We refer the reader to [9] for the terminology used herein. By comparing the functional equation in [9, Theorem 3] with the functional equation for the generating function B in the case under consideration (where $a_2=b_1=0$) we obtain the following result.

Theorem 4.2. *For $n \geqslant 1$, the number of intransitive trees on $[n+1]$ with k right vertices equals the number of standard local binary search trees on $[n]$ with $k-1$ right descents. In particular, the polynomials $f_n(x)$ considered by Postnikov are γ-positive. Therefore, the sequence of coefficients of $f_n(x)$ is unimodal.*

The second half of the above theorem follows from Equation (4.1). In fact, we conjecture something stronger; that for all $n \geqslant 1$, all of the roots of the polynomial $f_n(x)$ are negative real numbers. This in turn would imply that the coefficients of $f_n(x)$ form a *log-concave sequence*.

5 Further avenues

We now turn our attention to the case where exactly one of the variables has been set to 0. In this case we recover some deformation of the classical parking function representation. This is already implicit in our functional equation, but there is one curious aspect worth emphasizing: Our representation is not h-positive. In particular, it cannot be recovered as a graded Frobenius characteristic of the symmetric group action on parking functions or on labeled Dyck paths. It would be interesting to construct an $\mathfrak{S}_n$ representation with the right Frobenius characteristic.

Not included in this extended abstract are two more proofs of Theorem 1.2. The first is a generating function proof using Gessel's functional equation combined with a consequence of the Carlitz-Scoville-Vaughan Theorem [3]. The second involves a lift of Gessel's functional equation for the generating function S in the Hopf algebra of noncommutative symmetric functions defined in the seminal paper [7] and utilizes classical techniques to cancel out negative terms via flip equivalence on trees. Indeed, we demonstrate that all of our proofs have natural interpretations in the noncommutative setting.

There is much more to explore in terms of representation theory. In future work, we plan to explore the representation theory behind this Schur positivity phenomena, both from the $\mathfrak{S}_n$ perspective and the 0-Hecke perspective. Finally, the connections to diagonal harmonics mentioned above and work of Bergeron-Préville-Ratelle [1] remain to be explored.

Acknowledgements

We are especially grateful to Sara Billey for helpful discussions and remarks. Additionally, the second and third authors would like to thank Patricia Hersh and Jia Huang for helpful correspondence.

References

[1] F. Bergeron and L. F. Préville-Ratelle. “Higher trivariate diagonal harmonics via generalized Tamari posets”. *J. Combin.* **3** (2012), pp. 317–341. DOI.

[2] O. Bernardi. “Deformations of braid arrangements and Trees”. 2016. arXiv:1604.06554.

[3] L. Carlitz, R. Scoville, and T. Vaughan. “Enumeration of pairs of sequences by rises, falls and levels”. *Manuscripta Math.* **19** (1976), pp. 211–243. DOI.

[4] S. Corteel, D. Forge, and V. Ventos. “Bijections between affine hyperplane arrangements and valued graphs”. *European J. Combin.* **50** (2015), pp. 30–37. DOI.

[5] B. Drake. “An inversion theorem for labeled trees and some limits of areas under lattice paths”. PhD thesis. Brandeis University, 2008.

[6] D. Forge. “Linial arrangements and local binary search trees”. 2014. arXiv:1411.7834.

[7] I. Gelfand, D. Krob, A. Lascoux, B. Leclerc, V. Retakh, and J.-Y. Thibon. “Noncommutative symmetrical functions”. *Adv. Math.* **112** (1995), pp. 218–348. DOI.

[8] L. Kalikow. “Symmetries in trees and parking functions”. *Discrete Math.* **256** (2002), pp. 719–741. DOI.

[9] A. Postnikov. “Intransitive trees”. *J. Combin. Theory Ser. A* **79** (1997), pp. 360–366. DOI.

[10] L. F. Préville-Ratelle and X. Viennot. “An extension of Tamari lattices”. 2014. arXiv:1406.3787.

[11] R. P. Stanley. *Enumerative Combinatorics*. Vol. 2. Cambridge Univ. Press, 1999.

[12] V. Tewari. “Gessel polynomials, rooks, and extended Linial arrangements”. 2016. arXiv:1604.06894.