LABELED PLANE BINARY TREES AND SCHUR-POSITIVITY

IRA M. NESSEL, SEAN GRIFFIN, AND VASU TEWARI

Abstract. In 1995, Gessel introduced a multivariate generating function that tracks the distribution of ascents and descents in labeled binary trees. In addition to proving that it is symmetric, he conjectured that it is Schur-positive. We prove this conjecture by developing a weighted extension of a bijection of Prévile-Ratelle and Viennot relating pairs of paths and binary trees. In fact, this extension allows us to establish a stronger version of Gessel’s conjecture showing that the generating function restricted to labeled binary trees with a fixed canopy is still Schur-positive. We place our results in the context of a particular symmetric boolean decomposition of the lattice of noncrossing partitions, using a bijection of Edelman and the “modified Foata-Strehl equivalence” of Brändén. We also discuss applications in the setting of hyperplane arrangements. We show that a certain specialization of our multivariate generating function equals the Frobenius characteristic of the natural $S_n$-action on regions of the semiorder arrangement, which we then expand in terms of Foulkes characters. We also construct an $S_n$-action on regions of the Linial arrangement using a set of trees studied by Bernardi. We further prove $\gamma$-nonnegativity for the distribution of right descents over local binary search trees.

Contents

1. Introduction 2
2. Combinatorial preliminaries 6
  2.1. Compositions 6
  2.2. Symmetric functions 6
  2.3. Noncommutative symmetric functions 7
  2.4. Words 8
  2.5. Unlabeled trees and associated notions 9
  2.6. Labeled trees 10
  2.7. Marked trees 11
3. The Prévile-Ratelle–Viennot bijection 12
  3.1. Binary paths and PG sequences 13
  3.2. The PG Algorithm 14
4. Weighted extension of the Préville-Ratelle–Viennot bijection 17
  4.1. The extended PG algorithm 18
  4.2. The case of fixed canopy 25
  4.3. Noncommutative version 29
  4.4. A symmetric boolean decomposition of the lattice of noncrossing partitions 30

2010 Mathematics Subject Classification. Primary 05A19, 05E05, 05E18; Secondary 05A05, 05E10, 06A07.

Key words and phrases. labeled trees, Schur-positivity, hyperplane arrangements, Tamari lattices, ascents-descents.

The first author was supported by a grant from the Simons Foundation (#427060, Ira Gessel). The second author was supported by the ARCS Foundation Fellowship, as well as NSF Grant DMS-1101017. The third author was supported by an AMS-Simons Travel Grant.
1. Introduction

The study of permutation statistics is a classical theme in algebraic combinatorics with its genesis in work by MacMahon [40]. An important statistic introduced by MacMahon is the descent statistic on permutations. The generating function for the distribution of this statistic gives rise to the well-known Eulerian polynomials, which show up in many areas in mathematics and the reader is referred to [47] for a detailed survey. Since the work of MacMahon, the descent statistic on permutations has been studied in depth, and yet it continues to inspire new research [53, 54]. In this article we study ascent-descent statistics on labeled plane binary trees. For brevity’s sake, by a tree, we will always mean a plane binary tree. Whether the tree is labeled or not will be clear from context. We remark that the notion of descents has been studied with regards to other combinatorial objects before, such as in the case of standard Young tableaux (SYTs). However, viewing SYTs as P-partitions reveals that their descents are in fact descents of permutations in disguise. In contrast, the ascent-descent statistics that we study here are indeed different, as they depend on the embedding of the labeled trees in the plane and take into account the orientation of the edges.

More specifically, the ascent and descent statistics on labeled trees come in two flavors depending on whether one compares the label of the parent node to the label of its right child or its left child. We recover the case of ascents and descents of permutations by considering labeled trees in which no node has a left child, or alternatively, by considering labeled trees in which no node has a right child. Thus, the study of these statistics on labeled binary trees is a natural generalization of the study of ascents and descents on permutations. Given a positive integer $n$, let $T_{n,\ell}$ (respectively $T_{n}$) denote the set of labeled (respectively unlabeled) plane binary trees on $n$ nodes. The labels on the nodes are drawn from the set of positive integers $\mathbb{P}$, allowing repeats. A standard labeling of a tree $T \in T_{n,\ell}$ is a labeling of its nodes with distinct labels drawn from $[n] := \{1, \ldots, n\}$, and we will call a tree with a standard labeling a standard labeled tree. For any labeled tree $T \in T_{n,\ell}$, let $\text{lasc}(T)$, $\text{ldes}(T)$, $\text{rasc}(T)$ and $\text{rdes}(T)$ denote the number of ascents and descents in the labeling to the left and right, such that rasc and ldes are determined by weak inequalities, whereas rdes and lasc are strict.

Gessel, in the 1990s, initiated the study of these statistics and considered the following generating function tracking their distribution over the set of standard labeled trees.

\[
B := B(x; a_1, a_2, b_1, b_2) = \sum_{n \geq 1} \sum_{\substack{T \in T_{n,\ell} \text{ standard} \atop T \in T_{n}}} a_1^{\text{rasc}(T)} a_2^{\text{rdes}(T)} b_1^{\text{lasc}(T)} b_2^{\text{ldes}(T)} \frac{x^n}{n!}.
\]
In unpublished work, Gessel showed that $B$ satisfies the functional equation

$$
(1 + a_1 B)(1 + b_2 B) = e^{[(a_1 b_2 - b_1 a_2)B + a_1 - a_2 - b_1 + b_2]x}.
$$

Equation (1.1) was later proved by Kalikow [39] and Drake [20].

Subsequently, different proofs of Equation (1.1) were given by Kalikow [39] and Drake [20]. From the definition of $B$, we observe that $B(x; a_1, a_2, b_1, b_2) = B(x; b_1, b_2, a_1, a_2)$ and $B(x; a_1, a_2, b_1, b_2) = B(x; a_2, a_1, b_1, b_2)$. The former is explained by reflecting a standard labeled tree across a vertical line passing through its root, whereas the latter follows from changing the label of a node from $i$ to $n - i + 1$ in a standard labeled tree in $T_n^+$. Equation (1.1) brings to light another pair of symmetries, that $B(x; a_1, a_2, b_1, b_2) = B(x; b_2, a_2, b_1, a_1)$ and $B(x; a_1, a_2, b_1, b_2) = B(x; a_1, b_1, a_2, b_2)$. These equalities are not obvious from the definition and a simple bijective proof for them remains elusive, although a complicated bijection can be derived from the work of Kalikow [39].

The recent impetus to the study of $B$ has been fueled by connections with enumerative aspects of the theory of hyperplane arrangements. Let $B_n := B_n(a_1, a_2, b_1, b_2)$ denote the coefficient of $x^n/n!$ in $B$ for $n \geq 1$. The expansion for $B_n$ when $1 \leq n \leq 5$ is given in Appendix [3]. Gessel observed that certain evaluations of $B_n$ coincide with the number of regions in various well-known deformations of Coxeter arrangements [29]. This viewpoint has been pursued in [18, 24, 65], and a complete explanation has been offered by Bernardi [10]. Given a subset $A$ of $\{-1, 0, 1\}$, we can consider the arrangement in $\mathbb{R}^n$ consisting of all hyperplanes $x_i - x_j = a$ where $i < j$ and $a \in A$. For $A = \{0\}$, $A = \{-1, 0, 1\}$, $A = \{-1, 1\}$, $A = \{0, 1\}$ and $A = \{1\}$, the corresponding hyperplane arrangements in $\mathbb{R}^n$ are the braid arrangement $B_n$, the Catalan arrangement $C_n$, the semiorder arrangement $I_n$, the Shi arrangement $S_n$, and the Linial arrangement $L_n$ respectively. These arrangements are very well-studied [6, 36, 49, 55, 56, 62] and are instances of deformations of Coxeter arrangements called truncated affine arrangements [49]. Various aspects of truncated affine arrangements have been studied in great detail in [2, 3, 5, 49] and we refer the reader to them for further information. Remarkably, we have the following equalities in which the left-hand side is an evaluation of $B_n$ and the right-hand side is the number of regions in a Coxeter arrangement deformation.

$B_n(1, 1, 1, 1) = \text{number of regions in } C_n = \frac{n!}{n+1} \binom{2n}{n}$

$B_n(1, 1, 1, 0) = \text{number of regions in } S_n = (n+1)^{n-1}$

$B_n(1, 1, 0, 0) = \text{number of regions in } B_n = n!$

$B_n(1, 0, 0, 1) = \text{number of regions in } L_n = \frac{1}{2^n} \sum_{k=0}^{n} \binom{n}{k} (k+1)^{n-1}$

$B_n(1, \zeta_6, \bar{\zeta}_6, 1) = \text{number of regions in } I_n$.

In the last equality, $\zeta_6$ denotes a primitive sixth root of unity and $\bar{\zeta}_6$ denotes its complex conjugate. Section [5] of this article is devoted to a representation-theoretic understanding of these equalities.

Our primary object of study is a multivariate generalization of $B$ introduced by Gessel. Let $x = \{x_1, x_2, \ldots\}$ be a commuting set of indeterminates. With every $T \in \mathcal{T}_n^+$, we associate a monomial $x_T$ as follows. For a node $v \in T$ labeled $i$, let $x_v$ be $x_i$. Then

$$x_T = \prod_{v \in T} x_v.$$
Now consider the formal power series in \( x \) with coefficients in \( \mathbb{Q}[a_1, a_2, b_1, b_2] \),
\[
G := G(x; a_1, a_2, b_1, b_2) = \sum_{n \geq 1} \sum_{T \in T_n} a_1^{\text{rasc}(T)} a_2^{\text{rdes}(T)} b_1^{\text{lasc}(T)} b_2^{\text{ldes}(T)} x^n.
\]
It transpires that \( G \) is a symmetric function in \( x \) with coefficients in \( \mathbb{Q}[a_1, a_2, b_1, b_2] \). This non-obvious fact follows from the following functional equation satisfied by \( G \).

**Theorem 1.1** (Gessel). Let \( H(z) = \sum_{n \geq 0} h_n z^n \) where \( h_n \) denotes the \( n \)th complete homogeneous symmetric function. We have
\[
(1 + a_1 G)(1 + b_2 G) = H((a_1 b_2 - a_2 b_1)G + a_1 - a_2 - b_1 + b_2).
\]
Observe that the functional equation for \( B \) in Equation (1.1) can be obtained from that satisfied by \( G \) in Equation (1.2) by applying the homomorphism sending \( h_n \) to \( x^n/n! \). This homomorphism has the crucial feature of sending the coefficient of \( x_1 x_2 \cdots x_n \) in any symmetric function to the coefficient of \( x^n/n! \) in its image.

Given that \( G \) is a symmetric function, it is natural to ask for its expansion in the basis of Schur functions. This brings us to our first new result, which was originally conjectured by Gessel [30] in 1995.

**Theorem 1.2.** \( G \) is Schur-positive.

Here we mean that \( G \) may be expressed as a sum of Schur functions \( s_{\lambda} \) with coefficients in the semiring \( \mathbb{N}[a_1, a_2, b_1, b_2] \). Theorem 1.2 follows from another recursive functional equation satisfied by \( G \), which is also one of our main results.

**Theorem 1.3.** We have
\[
G = \sum_{n \geq 1} \sum_{\alpha} r_\alpha (a_1 b_2 G + a_1 + b_2)^{n-\ell(\alpha)} (a_2 b_1 G + a_2 + b_1)^{\ell(\alpha)-1},
\]
where \( r_\alpha \) denotes the ribbon Schur function indexed by the composition \( \alpha \) and \( \ell(\alpha) \) denotes the length of \( \alpha \).

In fact, Theorem 1.3 implies the much stronger fact that \( G \) may be expressed as a sum of ribbon Schur functions in the semiring \( \mathbb{N}[a_1, a_2, b_1, b_2] \). For \( n \geq 1 \), let \( G_n := G_n(x; a_1, a_2, b_1, b_2) \) denote the sum of the terms in \( G \) of total degree \( n \) in \( x \). The expansion for \( G_n \) when \( 1 \leq n \leq 5 \) is given in Appendix B. By solving the functional equation in Theorem 1.3 using a certain class of trees called marked trees, we give a combinatorial interpretation to the coefficients \( c_{i,j,k,l,\delta} \) in
\[
G_n = \sum_{i,j,k,l,\delta} c_{i,j,k,l,\delta} (a_1 b_2)^i (a_2 b_1)^j (a_1 + b_2)^k (a_2 + b_1)^l r_\delta.
\]
A fact worth noting about the expansion in Equation (1.3) is that the coefficient of \( r_\alpha \) for every \( \alpha \vdash n \) evaluates to the Catalan number \( \text{Cat}_{\alpha} := \frac{1}{n+1} \binom{2n}{n} \) upon setting all of \( a_1, a_2, b_1 \), and \( b_2 \) equal to 1. Additionally, the coefficients of both \( r_{(1,1,\ldots,1)} \) and \( r_{(n)} \) upon setting \( a_1 = q, \ a_2 = t \) and \( a_2 = q, \ b_1 = t \) respectively are the homogenized Narayana polynomials
\[
\text{Nar}_{n}(q, t) := \sum_{k=0}^{n-1} \frac{1}{n} \binom{n}{k} \binom{n-1}{k} t^k q^{n-1-k}.
\]
We provide three proofs of Theorem 1.3, each with its own merits. First, we give an algebraic proof that follows from Theorem 1.1 combined with a result of MacMahon [40].
Vol. 1, p. 186]. The rest of the paper focuses on establishing a deeper combinatorial understanding of the Schur-positive expansion given by Theorem 1.3. Our second proof, which forms the technical heart of our paper, establishes a further refinement of Theorem 1.2 and involves a weighted extension of a beautiful bijection due to Préville-Ratelle and Viennot [51] pertaining to generalized Tamari lattices. This refinement was also conjectured by Gessel [30], and we state it next.

**Theorem 1.4.** Fix a positive integer $n$. Let $v$ be a word of length $n - 1$ in the alphabet \{U, D\}, and let $\mathcal{T}_{n,v}^\ell$ denote the set of labeled trees on $n$ nodes with canopy $v$. We have that the generating function

$$G_{n,v} := G_{n,v}(x; a_1, a_2, b_1, b_2) = \sum_{T \in \mathcal{T}_{n,v}^\ell} a_1^{\text{rasc}(T)} a_2^{\text{rdes}(T)} b_1^{\text{lasc}(T)} b_2^{\text{ldes}(T)} x_T$$

is Schur-positive.

Details on the terminology used in Theorem 1.4 can be found in Section 2, and its proof is present in Theorem 4.12. Our third proof proceeds by solving a noncommutative lift of the functional equation in Theorem 1.1.

We connect our earlier results back to deformations of Coxeter arrangements focusing in particular on semiorder and Linial arrangements. Our main results in this setting are the following.

**Theorem 1.5.** The Frobenius characteristic of the natural $S_n$-action on the set of regions of the semiorder arrangement $\mathcal{I}_n$ is $G_n(x; 1, 1, \zeta_6, \zeta_6)$.

**Theorem 1.6.** There exists an $S_n$-action on the set of regions of the Linial arrangement $\mathcal{L}_n$ whose graded Frobenius characteristic is given by $G_n(x; a_1, 0, 0, b_2)$.

The proof of Theorem 1.5 utilizes a cycle index series computation relying on a result of Postnikov-Stanley [49], whereas the proof of Theorem 1.6 utilizes crucially a recent bijection of Bernardi [10] relating regions of $\mathcal{L}_n$ to a certain class of labeled trees that we call Bernardi trees.

This article is the full version of the extended abstract [31].

**Outline of the article.** In Section 2, we introduce our main combinatorial objects, and develop most of the notation we need. Subsection 2.4 provides a generating function proof of Theorem 1.3 and proves Theorem 1.2. Next we introduce tools necessary to prove Theorem 1.4. In Section 3 we describe the Préville-Ratelle–Viennot bijection between pairs of paths, which we refer to as glued pairs, and binary trees. In Section 4 we describe our weighted extension and after establishing various properties satisfied by our variant, we give a positive expansion for $G_{n,v}$ in terms of ribbon Schur functions in Theorem 4.12, thereby proving Theorem 1.4. In the same section, our Corollary 4.16 gives a natural noncommutative analogue of Theorem 1.3. We proceed to discuss the relation between our expansion for $G_n$ in Theorem 4.17 and a particular symmetric boolean decomposition of the lattice of noncrossing partitions, using a special case of a bijection due to P. Edelman [21]. We also draw parallels between our work and that of Simion-Ullman [58]. In Section 5, we discuss applications of our results to studying actions of the symmetric group on Coxeter deformations focusing in particular on semiorder and Linial arrangements. We conclude with further avenues in Section 6. Appendix A gives a third proof of Theorem 1.2 by considering a noncommutative lift of the functional equation in Theorem 1.1 to the algebra of noncommutative symmetric functions NSym introduced in the seminal paper [28]. Appendix B lists expansions for $B_n$ and $G_n$ for some small values of $n$. 
2.1. Compositions. A finite ordered list of positive integers $\alpha = (\alpha_1, \ldots, \alpha_\ell)$ is called a composition. If $\sum_{i=1}^\ell \alpha_i = n$, then we say that $\alpha$ is a composition of size $n$ and denote this by $\alpha \vdash n$. We call the $\alpha_i$ the parts of $\alpha$ and denote the number of parts of $\alpha$, also called the length of $\alpha$, by $\ell(\alpha)$. A partition $\lambda$ of size $n$, denoted by $\lambda \vdash n$, is a composition $(\lambda_1, \ldots, \lambda_k)$ of size $n$ satisfying $\lambda_1 \geq \cdots \geq \lambda_k$.

We define two operations on compositions. Given compositions $\alpha = (\alpha_1, \ldots, \alpha_\ell)$ and $\beta = (\beta_1, \ldots, \beta_m)$, we define the concatenation of $\alpha$ and $\beta$, denoted by $\alpha \bullet \beta$, to be the composition $(\alpha_1, \ldots, \alpha_\ell, \beta_1, \ldots, \beta_m)$. The near-concatenation of $\alpha$ and $\beta$, denoted by $\alpha \circ \beta$, is defined to be the composition $(\alpha_1, \ldots, \alpha_{\ell-1}, \alpha_\ell + \beta_1, \beta_2, \ldots, \beta_m)$. For example, if $\alpha = (2, 1, 3)$ and $\beta = (4, 1)$, then $\alpha \bullet \beta = (2, 1, 3, 4, 1)$ while $\alpha \circ \beta = (2, 1, 7, 1)$.

Recall the well-known bijection between compositions $\alpha = (\alpha_1, \ldots, \alpha_k)$ of $n$ and subsets $S \subseteq [n-1]$ given by $S = \{\alpha_1, \alpha_1 + \alpha_2, \ldots, \alpha_1 + \cdots + \alpha_{k-1}\}$. We will denote the set corresponding to $\alpha \vdash n$ by $\text{set}(\alpha)$, and in the opposite direction, given $S \subseteq [n-1]$, we will denote the corresponding composition of size $n$ by $\text{comp}(S)$. The inclusion order on subsets induces a natural poset structure on the set of compositions of size $n$. More precisely, given $\alpha, \beta \vdash n$, we say that $\alpha \leq \beta$ if and only if $\text{set}(\beta) \subseteq \text{set}(\alpha)$, and call $\leq$ the refinement order on compositions. For instance, consider $\alpha = (1, 2, 4, 2, 3, 2, 1)$ and $\beta = (3, 4, 7, 1)$, both compositions of size 15. Then $\text{set}(\beta) = \{3, 7, 14\}$ and $\text{set}(\alpha) = \{1, 3, 7, 9, 12, 14\}$. Clearly we have that $\text{set}(\beta) \subseteq \text{set}(\alpha)$, and therefore $\alpha \leq \beta$. We will denote this poset on compositions of size $n$ by $\text{Comp}_n$, and refer to it as the composition poset. Given compositions $\alpha, \beta \vdash n$, we will denote the interval in $\text{Comp}_n$ comprising compositions $\delta$ satisfying $\beta \leq \delta \leq \alpha$ by $[\beta, \alpha]$.

**Remark 2.1.** In our examples we will often omit commas and parentheses while writing compositions.

2.2. Symmetric functions. For notions related to the algebra of symmetric functions, denoted by $\text{Sym}$, that are not made explicit here, we refer the reader to [60]. An important class of symmetric functions is the set of ribbon Schur functions. These functions are special instances of skew Schur functions indexed by skew shapes that are ribbons, which are connected skew shapes that do not contain a $2 \times 2$ box. Assuming that we draw our skew shapes following the English convention, we can associate a composition to a ribbon by counting the number of boxes in every row of the ribbon from bottom to top. This association allows us to consider ribbon Schur functions as being indexed by compositions. We will refer to the ribbon Schur function indexed by a composition $\alpha$ as $r_\alpha$. An alternative definition of $r_\alpha$ is

$$r_\alpha := \sum x_{i_1} x_{i_2} \cdots x_{i_n},$$
where the sum is over all \( n \)-tuples \((i_1, \ldots, i_n)\) of positive integers satisfying \( i_j \leq i_{j+1} \) if \( j \notin \text{set}(\alpha) \) and \( i_j > i_{j+1} \) otherwise. Given a partition \( \lambda \) of \( n \), a standard Young tableau (henceforth SYT) of shape \( \lambda \) is a filling of the boxes of the Young diagram of \( \lambda \) with integers from \([n]\) using each label exactly once such that the labeling increases along rows and down columns. We denote the set of all SYTs of shape \( \lambda \) by \( \text{SYT}(\lambda) \). For \( T \in \text{SYT}(\lambda) \), we define \( \text{Des}(T) \) to be the set of all \( 1 \leq i \leq n-1 \) such that \( i \) belongs to a row above \( i+1 \) in \( T \). The following result describes the expansion of a ribbon Schur function in the basis of Schur functions.

**Proposition 2.2.** [66, Equation 2.2.4] Given a composition \( \alpha \vdash n \), we have

\[
 r_\alpha = \sum_{\lambda \vdash n} b_{\lambda \alpha} s_\lambda,
\]

where \( b_{\lambda \alpha} \) is the number of \( T \in \text{SYT}(\lambda) \) satisfying \( \text{comp}(\text{Des}(T)) = \alpha \). In particular, ribbon Schur functions are Schur-positive.

Let \( \text{ex} \) denote the homomorphism from \( \text{Sym} \) to \( \mathbb{Q}[[x]] \) mapping \( h_n \) to \( x^n/n! \) [60, Section 7.8]. This homomorphism is known as the exponential specialization and, as mentioned in the introduction, has the property that the coefficient of \( x^n/n! \) in the image \( \text{ex}(f) \) of a symmetric function \( f \) is the coefficient of \( x_1 \cdots x_n \) in \( f \) [60, Proposition 7.8.4].

### 2.3. Noncommutative symmetric functions.

Let \( X = \{x_1, x_2, \ldots \} \) be a set of countably many noncommuting indeterminates. The algebra of noncommutative symmetric functions, denoted by \( \text{NSym} \), is the free associative algebra \( \mathbb{Q}\langle h_1, h_2, \ldots \rangle \) contained in the noncommutative polynomial algebra, where

\[
h_n = \sum_{i_1 \leq i_2 \leq \cdots \leq i_n} x_{i_1} x_{i_2} \cdots x_{i_n},
\]

with \( h_0 = 1 \). As the definition above suggests, for \( i \geq 1 \), \( h_i \) is called the \( i \)th noncommutative complete homogeneous symmetric function. Given a composition \( \alpha = (\alpha_1, \ldots, \alpha_k) \), we define \( h_\alpha := h_{\alpha_1} \cdots h_{\alpha_k} \). Setting \( \text{deg}(h_\alpha) = i \) endows \( \text{NSym} \) with the structure of a graded algebra in which the \( n \)th degree graded piece, denoted by \( \text{NSym}^n \), is the \( \mathbb{C} \)-linear span of the \( h_\alpha \) for \( \alpha \vdash n \).

The noncommutative ribbon Schur functions form another special basis for \( \text{NSym} \). Given a composition \( \alpha \), define the noncommutative ribbon Schur function \( r_\alpha \) by

\[
r_\alpha = \sum x_{i_1} x_{i_2} \cdots x_{i_n},
\]

where the sum is over all \( n \)-tuples \((i_1, i_2, \cdots, i_n)\) of positive integers satisfying \( i_j \leq i_{j+1} \) if \( j \notin \text{set}(\alpha) \) and \( i_j > i_{j+1} \) otherwise. The noncommutative ribbon Schur functions multiply according to the following rule.

**Proposition 2.3.** [28, Proposition 3.13] Given compositions \( \alpha \) and \( \beta \), we have

\[
r_\alpha r_\beta = r_{\alpha \cdot \beta} + r_{\alpha \odot \beta}.
\]

We denote the natural projection from \( \text{NSym} \) to \( \text{Sym} \) sending \( h_n \) to \( h_n \) by \( \chi \). It is easy to see that \( \chi(r_\alpha) = r_\alpha \). Applying \( \chi \) to both sides of the equality in Proposition 2.3 gives us an analogous equality for ribbon Schur functions.
2.4. Words. Let $\mathbb{P}$ be the set of positive integers. Let $\mathbb{P}^+$ be the set of nonempty words on $\mathbb{P}$, which is the set of finite sequences of positive integers with positive length. If $w$ is a word with letters $w_1, w_2, \ldots, w_n$, we write $w = w_1 \cdots w_n$. To $w$, we associate the monomial $x_w = x_{w_1}x_{w_2}\cdots x_{w_n}$. An ascent of $w$ is an index $1 \leq i \leq n - 1$ such that $w_i < w_{i+1}$. A descent of $w$ is an index $1 \leq i \leq n - 1$ such that $w_i > w_{i+1}$. Let the descent set of $w$ be $\text{Des}(w) := \{1 \leq i \leq n - 1 \mid w_i > w_{i+1}\}$. The standardization of $w$, denoted by $\text{std}(w)$, is the permutation in $S_n$ obtained by replacing the entries of $w$ with $1, 2, \ldots, n$, keeping the same relative order, where repeated letters are considered as increasing from left to right. For example, the standardization of $112123$ is $124356$. Let $\text{asc}(w)$ and $\text{des}(w)$ be the number of ascents and descents in $w$ respectively. We will use a 2-parameter weighted power series analogue of the Eulerian polynomial,

\[
A(x; s, t) := \sum_{w \in \mathbb{P}^+} s^{\text{asc}(w)} t^{\text{des}(w)} x_w.
\]

It is easy to see that

\[
A(x; s, t) = \sum_{n \geq 1} \sum_{\alpha \vdash n} s^{n-\ell(\alpha)} t^{\ell(\alpha)-1} r_\alpha.
\]

By the homogenized version of a result of MacMahon [40, Vol. 1, p. 186], we have that

\[
A(x; s, t) = \sum_{n \geq 1} \sum_{\alpha \vdash n} s^{n-\ell(\alpha)} t^{\ell(\alpha)-1} r_\alpha.
\]

Letting $A := A(x; s, t)$ and clearing denominators, we obtain the compact formula

\[
H(s-t) = \frac{1+sA}{1+tA}.
\]

At this point we have all the tools we need to establish Theorem 1.3, which also implies Theorem 1.2.

**Proof of Theorem 1.3.** After we make the substitution $s = a_1b_2G + a_1 + b_2$ and $t = a_2b_1G + a_2 + b_1$, $H(s-t)$ becomes the right-hand side of Equation (1.2), which can be rewritten as

\[
(1 + a_1G)(1 + b_2G) = 1 + As.
\]

Since $(1 + a_1G)(1 + b_2G) = 1 + Gs$ and $(1 + a_2G)(1 + b_1G) = 1 + Gt$, Equation (2.4) is equivalent to

\[
\frac{1+Gs}{1+Gt} = \frac{1+As}{1+At},
\]

which implies that

\[
G = A = A(x; a_1b_2G + a_1 + b_2, a_2b_1G + a_2 + b_1),
\]

and Theorem 1.3 follows by Equation (2.2). \qed

The Schur-positivity of $G$ follows immediately by using Proposition 2.2 along with the commutative analogue of Proposition 2.3. However, this proof offers little insight into the combinatorial meaning of the ribbon Schur expansion given in the theorem. More importantly, it does not allow us to prove Theorem 1.4. Thus, the rest of the paper is dedicated to exploring the deeper combinatorial connections underlying Theorem 1.3.
2.5. Unlabeled trees and associated notions. A plane binary tree is a rooted tree in which every node has at most two children, of which at most one is called a left child and at most one is called a right child. We will denote the set of all plane binary trees by \( \mathcal{T} \) and that of plane binary trees on \( n \) nodes for \( n \geq 1 \) by \( \mathcal{T}_n \). Recall that we use the term tree to mean a plane binary tree. Elements of \( \mathcal{T} \) will be considered unlabeled trees. In other words, isomorphic elements of \( \mathcal{T} \) are considered to be the same. We will denote the set of nodes of \( T \) by Nodes(\( T \)), and the root node will be referred to as root(\( T \)). We will abuse notation on occasion and write \( v \in T \) when we mean \( v \in \text{Nodes}(T) \).

The nodes of a tree can be categorized as terminal nodes, which are nodes with no children, and internal nodes, which are nodes with at least one child. The set of terminal nodes of a tree \( T \) will be denoted by Term(\( T \)). The internal nodes of \( T \) can be classified 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. The sets of univalent nodes and bivalent nodes will be denoted by Bi(\( T \)) and Uni(\( T \)) respectively. A univalent node that has only a right (respectively left) child is called a right univalent node (respectively left univalent node). The set of right (respectively left) univalent nodes is denoted by Uni\(^r\)(\( T \)) (respectively Uni\(^l\)(\( T \)).

Given a tree \( T \), we will work with two different total orders on Nodes(\( T \)): preorder and inorder. To define them we need some notation. Given a node \( v \in T \), the subtree rooted at \( v \) is denoted by \( T_v \). The distance of \( v \) from root(\( T \)), denoted by \( d_T(\text{root}(T), v) \), is the number of edges in the unique path from root(\( T \)) to \( v \). We say that \( v \) is a descendant of a node \( u \) if \( u \) lies on the unique path from root(\( T \)) to \( v \). If \( v \) and \( u \) are distinct, then we call \( v \) a proper descendant of \( u \). The lowest common ancestor of nodes \( u \) and \( v \) is the unique node \( x \) such that both \( u \) and \( v \) are descendants of \( x \) and \( d_T(\text{root}(T), x) \) is maximal. The preorder \( \leq_p \) on \( T \) is defined to be the total order on Nodes(\( T \)) where for nodes \( u \) and \( v \) we have \( u \leq_p v \) if and only if exactly one of the conditions below holds.

1. \( v \) is a proper descendant of \( u \).
2. If \( w \) is the lowest common ancestor of \( u \) and \( v \), then \( v \) belongs to the left subtree of \( T_w \) and \( u \) belongs to the right subtree of \( T_w \).

We define the inorder \( \leq_i \) on \( T \) to be the total order on Nodes(\( T \)) where for distinct nodes \( u \) and \( v \) we have \( u \leq_i v \) if and only if exactly one of the conditions below holds.

1. \( u \) is a descendant of the left child of \( v \).
2. If \( w \) be the lowest common ancestor of \( u \) and \( v \), then \( v \) belongs to the right subtree of \( T_w \) and \( u \) belongs to the left subtree of \( T_w \).

Figure 1 gives an example each of preorder and inorder respectively.

**Figure 1.** The nodes labeled according to preorder on the left, and according to inorder on the right.

The preorder \( \leq_p \) on \( T \in \mathcal{T}_n \) allows us to associate a composition of \( n \) with it, which we call the type of the tree and denote by \( \mathbf{c}(T) \). To compute \( \mathbf{c}(T) \), let \( v_1 \leq_p \cdots \leq_p v_n \) be the nodes of \( T \) in preorder. Assume further that \( v_{i_1} \leq_p \cdots \leq_p v_{i_k} \) are all the elements of
Term(T). Note that \( 1 \leq i_1 < \cdots < i_k = n \). We now define \( c(T) := \text{comp}\{i_1, \ldots, i_{k-1}\} = (i_1, i_2 - i_1, i_3 - i_2, \ldots, i_k - i_{k-1}) \). Clearly \( c(T) \vdash n \). The reader can verify that for the tree \( T \) in Figure 1, we have \( c(T) = (3, 2, 3, 1) \).

Even though we will not be working with directed trees explicitly, we will work under the convention that the edge \( pq \) refers to the edge joining \( p \) and \( q \) in the tree and \( p \) is closer to the root than \( q \). In this case, \( p \) must be an internal node. If \( q \) is a right (respectively left) child of \( p \), we refer to the edge \( pq \) as a right descent (respectively left ascent) of \( p \).

Next we define an important subset of \( T \) that will feature in many of our results. It is the subset comprising trees satisfying the condition that every univalent node is a right univalent node. We refer to such trees as right-leaning trees and denote the set of right-leaning trees by \( RT \). The set of trees in \( RT \) with \( n \) nodes will be denoted by \( RT_n \).

In our pictorial depiction of trees, we can append two unlabeled leaves to all the terminal nodes, as well as one unlabeled leaf to each internal node with one child, thus obtaining a “complete binary tree”, which will help us to define the canopy of a tree. We denote the completion of a tree \( T \) by \( \mathcal{T} \). For the tree \( T \) in Figure 1, we obtain the tree \( \mathcal{T} \) in Figure 2 upon completion. To obtain \( \mathcal{T} \) from \( T \in \mathcal{T}_n \), we have to append \( n + 1 \) leaves. These leaves can be classified as left leaves and right leaves. If we traverse the tree \( \mathcal{T} \) in inorder and record the leaves in addition to the nodes by denoting left leaves by \( D \) and right leaves by \( U \), we obtain a word of length \( n + 1 \) in the alphabet \( \{U, D\} \). Since this word always begins with a \( D \) and ends with a \( U \), we lose no information by removing its first and last letters. The resulting word of length \( n - 1 \) is called the canopy of the tree and denoted by \( \text{can}(T) \).

![Figure 2](image.png)

**Figure 2.** The completion \( \mathcal{T} \) of a tree \( T \). In this case \( \text{can}(T) = UDUUUDDUD \).

### 2.6. Labeled trees.

A labeled plane binary tree (or simply a labeled tree) is a tree whose nodes have labels drawn from the set of positive integers \( \mathbb{P} \). We denote the set of labeled trees by \( \mathcal{T}^\ell \) and the set of labeled trees on \( n \) nodes for \( n \geq 1 \) by \( \mathcal{T}_n^\ell \). Given \( T \in \mathcal{T}_n^\ell \), we denote by \( \text{sh}(T) \) the unlabeled tree obtained by removing the labels on the nodes of \( T \). Given a node \( u \) in \( T \), we refer to the label on \( u \) as \( u^\ell \). We associate two reading words with \( T \): the preorder reading word denoted by \( \text{pr}(T) \), and the inorder reading word denoted by \( \text{in}(T) \). If \( v_1, \ldots, v_n \) are the nodes of \( T \) in preorder, then \( \text{pr}(T) := v_1^\ell \cdots v_n^\ell \). On the other hand, if \( v_1, \ldots, v_n \) are the nodes of \( T \) in inorder, then \( \text{in}(T) := v_1^\ell \cdots v_n^\ell \).

For a labeled tree, we have a refined classification for its edges. Suppose that \( q \) is the right child of \( p \). If \( p^\ell \leq q^\ell \), then \( pq \) is a right ascent. Otherwise it is a right descent. Now suppose that \( q \) is the left child of \( p \). If \( p^\ell < q^\ell \), then \( pq \) is a left ascent. Otherwise it is a left descent. Using this classification, we associate a weight \( \text{wt}(T) \) to a labeled tree \( T \) as follows,

\[
\text{wt}(T) := a_1^{\text{rasc}(T)} a_2^{\text{rdes}(T)} b_1^{\text{lasc}(T)} b_2^{\text{ldes}(T)}.
\]

Recall that \( \text{rasc}(T) \) (respectively, \( \text{rdes}(T), \text{lasc}(T) \) and \( \text{ldes}(T) \)) is the number of right ascents (respectively, right descents, left ascents and left descents) in \( T \). For the labeled
Similarly, we define the nodes that are either marked or terminal, then let \( T \) be the marked tree. We will depict marked trees by thickening the boundary of the marked nodes. The set of marked trees will be denoted by \( \mathcal{T} \) and the subset comprising marked trees on \( n \) nodes for \( n \geq 1 \) will be denoted by \( \mathcal{T}_n \). We can define labeled marked trees in the obvious manner, and we will denote the analogues of the two subsets defined in the preceding statement by \( \mathcal{T}^m \) and \( \mathcal{T}_n^m \) respectively. We will abuse notation and, given \( T \in \mathcal{T}_n \), denote by \( sh(T) \) the unlabeled tree obtained by forgetting the marking. In a similar vein, we will denote by \( c(T) \) the type of the unlabeled tree obtained by forgetting the marking.

The nodes of a marked tree can be classified according to the marking and number of children. Let \( Bi^m(T) \) (respectively \( Bi^u(T) \)) denote the set of marked bivalent nodes (respectively unmarked bivalent nodes) in \( T \). Also, let \( Uni^m(T) \) (respectively \( Uni^u(T) \)) denote the set of marked right (respectively left) univalent nodes. Similarly, let \( Uni^m(T) \) (respectively \( Uni^u(T) \)) denote the set of unmarked right (respectively left) univalent nodes. Finally, the set of marked (respectively unmarked) univalent nodes is denoted by \( Uni^m(T) \) (respectively \( Uni^u(T) \)). As we did for labeled trees, we can associate a weight \( wt(T) \) with a marked tree \( T \) as follows.

\[
wt(T) := (a_1b_2)^{Bi^u(T)}(a_2b_1)^{Bi^m(T)}a_1^{Uni^u(T)}b_1^{Uni^m(T)}a_2^{Uni^m(T)}b_2^{Uni^u(T)}.
\]

We associate two compositions to a marked tree akin to the notion of the type of a tree. Let \( \hat{T} \in \hat{\mathcal{T}}_n \) and let \( T := sh(\hat{T}) \). Suppose that \( v_1, \ldots, v_n \) are the nodes of \( \hat{T} \) in preorder. The lower type of \( \hat{T} \), denoted by \( \hat{c}(\hat{T}) \), is defined as follows. If \( v_{i_1} \prec v_{i_2} \prec \cdots \prec v_{i_k} \) are the nodes that are either marked or terminal, then \( \hat{c}(\hat{T}) := (i_1, i_2 - i_1, i_3 - i_2, \cdots, i_k - i_{k-1}) \). Similarly, we define the upper type of \( \hat{T} \), denoted by \( \hat{c}(\hat{T}) \), as follows. If \( v_{j_1} \prec \cdots \prec v_{j_m} \) are the marked nodes and \( m \geq 1 \), then \( \hat{c}(\hat{T}) := (j_1, j_2 - j_1, \ldots, j_m - j_{m-1}, n - j_m) \). Otherwise, \( \hat{T} \) has no marked nodes and we set \( \hat{c}(\hat{T}) := (n) \). Clearly, both \( c(T) \) and \( \hat{c}(\hat{T}) \) are compositions of \( n \), but more importantly, note that \( \hat{c}(\hat{T}) \preceq c(T) \). Thus, there are uniquely determined compositions \( \beta^{(1)}, \ldots, \beta^{(r)} \) such that \( \hat{c}(\hat{T}) = \beta^{(1)} \odot \cdots \odot \beta^{(r)} \) and \( (|\beta^{(1)}|, \ldots, |\beta^{(r)}|) = c(T) \). We refer to this sequence of compositions \( \beta^{(1)}, \ldots, \beta^{(r)} \) as the de-composition sequence of \( \hat{T} \). Observe further that \( \hat{c}(\hat{T}) = \beta^{(1)} \odot \cdots \odot \beta^{(r)} \). Consider the marked tree \( \hat{T} \) in Figure 3 with nodes \( v_1 \) through \( v_9 \) in preorder. The marked nodes are \( v_2, v_6, \) and \( v_7 \) and the terminal nodes are \( v_3, v_5, v_8, \) and \( v_9 \). Thus, \( \hat{c}(\hat{T}) = (2, 1, 2, 1, 1, 1, 1) \) and \( \hat{c}(\hat{T}) = (2, 4, 1, 2) \). If \( T := sh(\hat{T}) \), then \( c(T) = (3, 2, 3, 1) \) and hence, the de-composition sequence of \( \hat{T} \) is \((2, 1), (2, 1, 1, 1), (1)\).
At this point, we are ready to state an easy result whose commutative image will be used later on. The result follows readily from Proposition \[2.3.\]

**Lemma 2.4.** If \((\beta^{(1)}, \ldots, \beta^{(r)})\) is the de-composition sequence of \(\hat{T}\), then

\[
\mathbf{r}_{\beta^{(1)}} \cdots \mathbf{r}_{\beta^{(r)}} = \sum_{\delta \in [\hat{c}(\hat{T}), \hat{c}(\hat{T})]} \mathbf{r}_\delta.
\]

An important feature of the product in Lemma \[2.4.\] is that each \(\mathbf{r}_\delta\) for \(\delta \in [\hat{c}(\hat{T}), \hat{c}(\hat{T})]\) appears exactly once in the expansion. Observe further that, for each fixed choice of \(T \in \mathcal{T}_n\), the composition poset \(\operatorname{Comp}_n\) is the disjoint union of the intervals \([\hat{c}(\hat{T}), \hat{c}(\hat{T})]\) where \(\hat{T}\) ranges over all marked trees satisfying \(\text{sh}(\hat{T}) = T\). More concisely, we have

\[
\operatorname{Comp}_n = \bigsqcup_{\text{sh}(\hat{T}) = T} [\hat{c}(\hat{T}), \hat{c}(\hat{T})].
\]

This disjoint union property is equivalent to saying that every composition of size \(n\) belongs to a unique interval of the type \([\hat{c}(\hat{T}), \hat{c}(\hat{T})]\) for some choice of marking on \(T\). We now make the preceding statement precise.

Let \(v_1, \ldots, v_n\) be the nodes of \(T\) in preorder. Let \(\alpha \vdash n\) and \(S = \text{set}(\alpha)\). For every \(j \in S\), mark the node \(v_j\) if and only if \(v_j\) is not a terminal node. This gives us the unique marked tree \(\hat{T}\) such that \(T = \text{sh}(\hat{T})\) and \(\alpha \in [\hat{c}(\hat{T}), \hat{c}(\hat{T})]\). Let \(\mathcal{T}_\alpha\) denote the set of all marked trees on \(n\) nodes such that \(\alpha \in [\hat{c}(\hat{T}), \hat{c}(\hat{T})]\). From the disjoint union property in Equation \[2.5.\], it follows that \(|\mathcal{T}_\alpha| = \operatorname{Cat}_n\) for all \(\alpha\). Note that if \(\alpha = (1^n)\), then \(\mathcal{T}_\alpha\) is the set of marked trees on \(n\) nodes all of whose internal nodes are marked. Whereas, if \(\alpha = (n)\), then \(\mathcal{T}_\alpha\) is the set of marked trees none of whose internal nodes are marked, so \(\mathcal{T}_{(n)}\) is \(\mathcal{T}_n\). The marked tree \(\hat{T}\) in Figure 4 belongs to \(\mathcal{T}_\alpha\) where \(\alpha = (2, 1, 3, 1, 2)\). Note that \(\text{set}(\alpha) = \{2, 3, 6, 7\}\) and, following the procedure outlined earlier, to obtain our marked tree we mark the internal nodes \(v_2, v_6\) and \(v_7\) but leave the terminal node \(v_3\) unmarked.

We denote the set of **right-leaning marked binary trees** on \(n\) nodes by \(\mathcal{R}\mathcal{T}_n\). In addition to the usual notion of weight inherited from marked trees, we attach another modified weight \(\text{mwt}(\hat{T})\) to an element \(\hat{T} \in \mathcal{R}\mathcal{T}_n\) as follows,

\[
\text{mwt}(\hat{T}) := (a_1 b_2)^{\text{Bi}^u(\hat{T})} (a_2 b_1)^{\text{Bi}^m(\hat{T})} (a_1 + b_2)^{\text{Un}^u(\hat{T})} (a_2 + b_1)^{\text{Un}^m(\hat{T})}.
\]

### 3. The Préville-Ratelle–Viennot bijection

In this section, we provide a direct combinatorial proof of Theorem \[1.3.\]. We use a weighted extension of the Push-Gliding algorithm of Préville-Ratelle and Viennot from \[51.\] in order to define a weight-preserving bijection between the set of labeled binary trees and a set of combinatorial objects whose weighted multivariate generating function is equal to the right-hand side of Theorem \[1.3.\]. We will then see how the refined Schur-positivity statement in Theorem \[1.4.\] also follows from this weight-preserving bijection.
First, we make some remarks so that the reader can appreciate the context. The main result in [51] is a generalization of \( m \)-Tamari lattices to posets of arbitrary paths. Prévillé-Ratelle and Viennot were motivated by a question arising from what has come to be known as “rational Catalan combinatorics.” Bergeron introduced \( m \)-Tamari lattices in his study of higher trivariate diagonal harmonics. More details on this aspect are present in [9].

Given a positive integer \( m \) and nonnegative integer \( n \), an \( m \)-Dyck path of size \( n \) is a lattice path beginning at the origin and ending at \( (mn, n) \) which stays weakly above the line \( y = mx \), and the allowed steps are north steps (corresponding to a \((0,1)\)) and east steps (corresponding to a \((1,0)\)). The \textit{m-Tamari lattice} \( \text{Tam}_m^n \) is defined on the set of \( m \)-Dyck paths of size \( n \) where the partial order \( \leq_{\text{Tam}} \) is defined [12, Definition 2] as follows. Let \( P \) and \( Q \) be two \( m \)-Dyck paths of size \( n \). We have \( P \leq_{\text{Tam}} Q \) if there exists in \( P \) an east step \( a \) followed by a north step \( b \) such that \( Q \) is obtained by swapping \( a \) and \( F \) where \( F \) is the shortest factor of \( P \) beginning with \( b \) that is an \( m \)-Dyck path, potentially translated. An example of \( \leq_{\text{Tam}} \) is given in Figure 5 where the east step and the factor that get translated are highlighted. When \( m = 1 \), \( \text{Tam}_n^m \) coincides with the classical Tamari lattice [64].

\[ \leq_{\text{Tam}} \]

\[ \text{Figure 5.} \ A \text{ cover relation in } \text{Tam}_2. \]

Bergeron defined a symmetric group action on labeled \( m \)-Dyck paths as well as intervals in \( \text{Tam}_n^m \), and conjecturally linked this representation with trivariate diagonal coinvariant spaces. The character of this representation was computed in [13]. Bergeron’s conjectures have initiated a lot of study in relation to Tamari lattices, ranging from areas as diverse as enumeration of planar maps [22] to representation theory associated with rectangular parking functions and labeled parallelogram polyominoes [7, 8]. Furthermore, the geometry and topology of \( \text{Tam}_n^m \) have also garnered attention [16, 43].

Returning to the topic of this article, the work of Preville-Ratelle and Viennot suggests that we can interpret the generating functions \( G_{n,v} \) of Theorem 1.4 as symmetric functions with coefficients in \( \mathbb{Q}[a_1, a_2, b_1, b_2] \) associated to generalized Tamari lattices. The Schur-positivity of these symmetric functions which will be established in Theorem 4.12 hints at the existence of a graded \( \mathfrak{S}_n \)-module associated to generalized Tamari lattices whose graded Frobenius characteristic coincides with \( G_{n,v} \).

3.1. Binary paths and PG sequences. We proceed to describe the Push-Gliding algorithm (henceforth PG algorithm) of Préville-Ratelle and Viennot and our extension after setting up the necessary notation. Given a positive integer \( m \), a \textit{binary path} \( v \) of \textit{length} \( m - 1 \) is an ordered pair \( ((v_1, \ldots, v_m), w_v) \) where \( (v_1, \ldots, v_m) \) is an ordered sequence of nodes and \( w_v = w_1 \ldots w_{m-1} \) is a word of length \( m - 1 \) in the alphabet \( \{U, D\} \). We call \( v_1 \) (respectively \( v_m \) ) the \textit{initial node} (respectively \textit{final node}) of \( v \). A \textit{push-gliding sequence}, or \textit{PG sequence}, of \textit{length} \( m - 1 \) is a word \( u \) of length \( m - 1 \) in the alphabet \( \{P, G\} \). Given a PG sequence \( u \) and a binary path \( v \) of the same length, we say that the ordered pair \((u, v)\) forms a \textit{glued pair} if the following conditions hold.
(1) The number of $P$s in any prefix of $u$ is weakly greater than the number of $U$s in the prefix of the same length in $w$.

(2) The total number of $P$s in $u$ is equal to the total number of $U$s in $v$.

We identify the binary path $v$ with the sequence of nodes $v_1$ through $v_m$ situated in the plane where we join $v_i$ and $v_{i+1}$ for $1 \leq i \leq m - 1$ either by an up step (corresponding to a translation by $(1, 1)$) if $w_i = U$ or by a down step (corresponding to a translation by $(1, -1)$) if $w_i = D$. We will abuse notation and refer to this embedding in the plane as $v$ as well.

Consider a PG sequence $u$. By reading $u$ from left to right, we may construct a walk in the plane starting from $v_1$ where we take an up step (respectively down step) for every $P$ (respectively $G$) that we read. The condition that $p_{u, v}$ is a glued pair is equivalent to stating that this walk begins at $v_1$, ends at $v_m$, and stays weakly above the binary path corresponding to $v$. Given this correspondence, we conclude that our notion of glued pairs is the same as the notion of weakly non-crossing paths considered by [51, Section 1.4]. In view of this, an equivalent variation of the PG algorithm in [51] takes a glued pair $(u, v)$ as input and outputs a unique binary tree $PG(u, v)$. For a fixed binary path $v$, Préville-Ratelle and Viennot endow the set of glued pairs of the form $(u, v)$ with the structure of a lattice [51, Proposition 11] and, following them, we denote this lattice by $\text{Tam}(v)$. We extend their algorithm to labeled trees and binary paths, where the labels are drawn from the set of positive integers.

The right branch of a tree $T$ is the maximal sequence of nodes $u_1, \ldots, u_m$ in $T$ such that $u_1$ is the root node and $u_{i+1}$ is the right child of $u_i$ for $1 \leq i \leq m - 1$. The left branch of $T$ can be defined similarly. A tail $\tau$ attached to $T$ is a binary path such that its initial node is a node on the right branch of $T$ and $w_\tau$ is a nonempty word beginning with a $U$. We emphasize here that the only node that is common to both the tree $T$ and the tail $\tau$ is the initial node of $\tau$. An example of a tree $T$ with a tail $\tau$ attached to it is given in Figure 6.

Nodes shaded red and bolded edges correspond to the nodes and edges in the tree $T$. The nodes and edges that lie in the shaded region depict the tail $\tau$. In particular, the node $\tau_1$ is common to both the tail and the tree.

![Figure 6. A tree $T$ with an attached tail $\tau = ((\tau_1, \tau_2, \tau_3, \tau_4, \tau_5), UDDU)$.](image)

3.2. The PG Algorithm. This section introduces the PG algorithm of [51] which gives a bijection between glued pairs and plane binary trees. At every step in the algorithm, we keep track of a binary tree and a tail attached to the tree. The algorithm involves two subroutines Glide and Push both of which take as input a tree with an attached tail and output an updated tree and tail by either “gliding” the tail toward the root of the tree or “pushing” an initial segment of the tail into the tree. A pictorial description is given in Figures 7 and 8 and a formal description in Algorithms 1 and 2. As before, the nodes shaded red and bolded edges correspond to the nodes and edges in the tree at each step, whereas the nodes and edges that lie in the shaded region depict the tail at each step.
Algorithm 1 Gliding the tail toward the root of the tree (Glide)

Input: $T \in \mathcal{T}$ and $\tau = ((\tau_1, \ldots, \tau_m), w_\tau)$ a tail attached to $T$ such that $\tau_1 \neq \text{root}(T)$.
Output: $T \in \mathcal{T}$ and $\tau'$ a tail attached to $T$.

1: function Glide$(T, \tau)$
2: Let $\beta$ be the node in $T$ whose right child is $\tau_1$;
3: Let $\tau' := ((\beta, \tau_2, \tau_3, \ldots, \tau_m), w_\tau)$ be the new tail obtained from $\tau$ by replacing the starting node $\tau_1$ with $\beta$ but keeping the same sequence of $U$ and $D$ steps;
4: return $(T, \tau')$;

Algorithm 2 Pushing part of the tail into the tree (Push)

Input: $T \in \mathcal{T}$ and $\tau = ((\tau_1, \ldots, \tau_m), w_\tau)$ a tail attached to $T$.
Output: $T' \in \mathcal{T}$ and $\tau'$ a tail attached to $T'$.

1: function Push$(T, \tau)$
2: Let $k$ be the unique nonnegative integer such that $w_\tau = UD^kw'$ where $w'$ is either empty or begins with a $U$;
3: Let $T''$ be the tree rooted at $\tau_2$ with left subtree $T_{\tau_1}$ and (potentially empty) right subtree constructed by defining $\tau_{i+1}$ to be the right child of $\tau_i$ for $2 \leq i \leq k + 1$;
4: Let $T'$ be the tree obtained from $T$ by replacing $T_{\tau_1}$ with $T''$;
5: Let $\tau' = ((\tau_{k+2}, \tau_{k+3}, \ldots, \tau_m), w')$ be the binary path obtained by considering the final segment of $\tau$ starting at $\tau_{k+2}$;
6: return $(T', \tau')$;

The PG algorithm of Prévillé-Ratelle–Viennot utilizes these subroutines as follows. Given a glued pair $(u, v)$, the PG sequence $u$ read from left to right gives us the order in which we apply the Push and Glide subroutines in the algorithm. A formal description is given in Algorithm 3, and Figure 9 shows an example of how the PG algorithm transforms a glued pair into a binary tree.

The PG algorithm has the following crucial properties.

**Proposition 3.1** (Prévillé-Ratelle–Viennot [51]). Let $v = ((v_1, \ldots, v_m), w_v)$ be a binary path of length $m - 1 \geq 0$. Given a glued pair $(u, v)$, let $T = \text{PG}(u, v)$. Then

(i) The canopy of the tree $\text{PG}(u, v)$ output by the PG algorithm is $w_v$.
(ii) We have $v_1 \leq_i v_2 \leq_i \cdots \leq_i v_m$ in $T$.

Note that after running the algorithm, $u' = G^r$, where $r$ is the number of edges in the right branch of the output tree $\text{PG}(u, v)$ and $G^r$ denotes the word of length $r$ that only consists of $G$s. Observe further that Proposition 3.1 gives us a way of generating trees on $n$ nodes with a prescribed canopy $v$, which is also the reason for our interest in the PG algorithm.
Algorithm 3 The Push-Gliding algorithm (PG)

**Input:** A binary path $v = ((v_1, \ldots, v_m), w_v)$ and PG sequence $u$ such that $(u, v)$ is a glued pair.

**Output:** $T \in \mathcal{T}_m$.

1. function $\text{PG}(u, v)$
2. Let $k$ be the unique nonnegative integer such that $w_v = D^k w'$ where $w'$ is either the empty word or begins with a $U$;
3. Let $T$ be the tree rooted at $v_1$ such that $v_i$ is the right child of $v_{i-1}$ for $2 \leq i \leq k+1$;
4. Let $\tau := ((v_{k+1}, v_{k+2}, \ldots, v_m), w')$;
5. while the length of $\tau$ is $\geq 1$ do
6. if $u$ begins with $P$ then
7. $(T, \tau) \leftarrow \text{Push}(T, \tau)$;
8. else
9. $(T, \tau) \leftarrow \text{Glide}(T, \tau)$;
10. Remove the first letter of $u$;
11. return $T$;

Figure 9. The PG algorithm performed on a glued pair $(u, v)$.
4. Weighted extension of the Preville-Ratelle–Viennot bijection

We define labeled analogs of various objects considered in Subsection 3.1, beginning with binary paths. Given a positive integer \( m \), a labeled binary path \( v \) of length \( m - 1 \) is an ordered pair \( (\sigma_1, \ldots, \sigma_m, \ell, \phi) \) where \( (v_1, \ldots, v_m) \) is an ordered sequence of labeled nodes and \( \ell = \ell(v_1 \ldots v_m) \) is a word of length \( m - 1 \) in the alphabet \( \{U, D\} \). As usual, our labels are drawn from \( P \) and \( \ell, \phi \) is the label on node \( v_i \). Let the inorder reading word of \( v \), denoted by \( \text{in}(v) \), be \( \ell_1 \ldots \ell_n \). We identify \( v \) with the sequence of labeled nodes \( v_1 \) through \( v_m \) situated in the plane, where we join \( v_i \) and \( v_{i+1} \) for \( 1 \leq i \leq m - 1 \) either by an up step if \( \ell_i = U \) or by a down step if \( \ell_i = D \). To associate a weight based on its orientation as indicated in Figure 10 where, for the sake of convenience, we have set \( x := v_1 \) and \( y := v_{i+1} \). The inequalities along the edges tell us how \( x^\ell \) and \( y^\ell \) compare and in turn determine the weights of the edges.

![Figure 10. Determining the weight of a labeled edge.](image)

We define the weight of the labeled binary path \( v \) to be the product of the weights on its edges, and denote it by \( \text{wt}(v) \). Observe that the weights in Figure 10 are consistent with the orientation of a labeled tree, as defined in the introduction, based on their orientation. Thus, if we define the weight of a labeled tree to be the product of the weights along its edges, then this new weight in fact coincides with the one in Subsection 2.6.

Given a PG sequence \( u \) and a labeled binary path \( v \), we abuse notation and call the ordered pair \( (u, v) \) a glued pair if \( (u, \bar{v}) \) is a glued pair, where \( \bar{v} \) is the binary path obtained from \( v \) by omitting the labels on the nodes. A labeled tail \( \tau \) attached to a labeled tree \( T \) is a labeled binary path such that its initial node is a node on the right branch of \( T \) and \( w_\tau \) is a nonempty word beginning with \( U \). Figure 11 gives an example of a labeled tree \( T \) with a labeled tail \( \tau \) attached to it. We define the weight of the ordered pair \( (T, \tau) \) comprising a labeled tree and a labeled tail, denoted by \( \text{wt}(T, \tau) \), to be \( \text{wt}(T) \text{wt}(\tau) \). For the pair \( (T, \tau) \) in Figure 11, we have \( \text{wt}(T, \tau) = (a_1^2 a_2^3 b_1 b_2^0)(a_1 a_2 b_2^0) = a_1^3 a_2^4 b_1 b_2^0 \).

![Figure 11. A tree T with an attached tail \( \tau = ((\tau_1, \tau_2, \tau_3, \tau_4, \tau_5), UDDDU) \).](image)

Motivated by Theorem 1.3, we interpret the right-hand side as the multivariate generating function of the set of sequences of labeled trees and binary paths, which we now make precise. Let \( \mathcal{P}T \ell \) be the set of sequences \( (v^{(1)}, T_1, v^{(2)}, \ldots, \phi(m-1), T_m-1, v^{(m)}) \) for \( m \geq 1 \) such that for each \( i \) satisfying \( 1 \leq i \leq m \), we have that \( v^{(i)} \) is a labeled binary path and...
weights are preserved. \( \tau \) for each \( i \), if \( x \) is the final node of \( v_i \) and \( y \) is the initial node of \( v_{(i+1)} \), then

\[
\gamma_i := \begin{cases} 
a_1b_2 & \text{if } x^i \leq y^i \\
a_2b_1 & \text{if } x^i > y^i. \end{cases}
\]

We define the **inorder reading word** of \( S = (v_{(1)} T_1, \ldots, T_{m-1}, v_{(m)}) \in \mathcal{PT}^\ell \), denoted by \( \text{in}(S) \), to be the concatenation of the inorder reading words of the labeled binary paths and labeled trees in the order in which they appear in the sequence from left to right. It is easy to verify that the right-hand side of Theorem \ref{1.3} is equal to the weighted generating function for the set of labeled trees and binary paths,

\[
\sum_{S \in \mathcal{PT}^\ell} \text{wt}(S)\text{in}(S).
\]

We will prove the following reformulation of Theorem \ref{1.3} and use it to prove Theorem \ref{1.4}.

**Theorem 4.1.** There is a bijection \( \Phi : \mathcal{T}^\ell \to \mathcal{PT}^\ell \) such that \( \text{wt}(T) = \text{wt}(\Phi(T)) \) and \(\text{in}(T) = \text{in}(\Phi(T))\).

Note that if we begin with a glued pair \((u, v)\) where \(v\) is a labeled binary path, then the PG algorithm outputs a labeled binary tree \(T := \text{PG}(u, v) \in \mathcal{T}^\ell\). Therefore, the PG algorithm provides a bijection between \(\mathcal{T}^\ell\) and the set of glued pairs \((u, v)\) such that \(v\) is a labeled binary path. By Proposition \ref{3.1} the inorder reading word of \(T\) equals the inorder reading word of \(v\). To construct the map \(\Phi\), we extend the PG algorithm to keep track of the weights.

### 4.1. The extended PG algorithm

We describe our analogues of the routines \texttt{Push} and \texttt{Glide}. These are the routines \texttt{wGlide} and \texttt{wPush}, illustrated in Figures \ref{fig:extended} and \ref{fig:push}, except this time we track the weights using the notion of **infected nodes**. Note that routines \texttt{Push} and \texttt{Glide} can be extended to labeled trees and tails. We will abuse notation and write \texttt{Push} and \texttt{Glide} for these extended routines. Applying a \texttt{Push} or \texttt{Glide} to a labeled tree \(T\) with a labeled tail \(\tau\) may change the weight \(\text{wt}(T, \tau)\). Crucially, these changes are local in the sense that they only involve the nodes \(\beta, \tau_1, \tau_2\), where \(\tau_1, \tau_2\) are the first two nodes of \(\tau\), and \(\beta\) is the node in \(T\) whose right child is \(\tau_1\). Furthermore, all other edge weights are preserved.

The routine \texttt{wGlide} first performs a \texttt{Glide} on the tree \(T\) and tail \(\tau\) to obtain a new tail \(\tau'\) attached to \(T\). Let \(\tau_1\) be the initial node in \(\tau\). Then, if \(\text{wt}(T, \tau) \neq \text{wt}(T, \tau')\), then all nodes in the subtree of \(T\) rooted at \(\tau_1\) are added to the “infected set”, and we no longer track the weights of the edges in that subtree. Similarly, the routine \texttt{wPush} first performs a \texttt{Push} on the tree \(T\) and tail \(\tau\) to obtain a tail \(\tau'\) attached to a new tree \(T'\). Then, if \(\text{wt}(T, \tau) \neq \text{wt}(T', \tau')\), all nodes in the subtree rooted at \(\tau_1\) are added to the “infected set.”

However, there are two exceptional cases we will need to consider in order to construct \(\Phi\) so that it is a bijection. First, if \(\beta^i > \tau_1^i > \tau_2^i\) and a \texttt{Glide} is performed, add all nodes in the subtree of \(T\) rooted at \(\tau_1\) to the “infected set.” Second, if \(\beta^i \leq \tau_1^i \leq \tau_2^i\) and a \texttt{Push} is performed, add all nodes in the subtree of \(T\) rooted at \(\tau_1\) to the “infected set.”

More succinctly, it can be shown that if a \texttt{Glide} is performed, we infect the nodes in the subtree rooted at \(\tau_1\) if and only if \(\text{std}(\beta^i) \in \{321, 312, 132\}\). If a \texttt{Push} is performed, we
inflict the nodes in the subtree rooted at \( \tau_1 \) if and only if \( \text{std}(\beta \ell_1 \tau_1) \in \{123, 213, 231\} \). A formal description of these new routines is given in Algorithms 4 and 5. Using the routines \( \text{wGlide} \) and \( \text{wPush} \), we formally describe our weighted analogue of the PG algorithm, referred to as \( \text{ExtendedPG} \), in Algorithm 6.

**Algorithm 4 \( \text{wGlide} \)**

**Input:** \( T \in \mathcal{T} \), \( \tau = ((\tau_1, \ldots, \tau_m), w_\tau) \) a labeled tail attached to \( T \) such that \( \tau_1 \neq \text{root}(T) \), and \( I \subseteq \text{Nodes}(T) \).

**Output:** \( T' \in \mathcal{T} \), \( \tau' \) a labeled tail attached to \( T \), and \( I' \subseteq \text{Nodes}(T) \).

\[
\begin{align*}
1: & \quad \text{function } \text{wGlide}(T, \tau, I) \\
2: & \quad \text{Let } \beta \text{ be the node in } T \text{ whose right child is } \tau_1; \\
3: & \quad (T', \tau') \leftarrow \text{Glide}(T, \tau); \\
4: & \quad \text{if } w(T, \tau) \neq w(T', \tau') \text{ or } \beta \ell > \tau_1 > \tau_2 \text{ then} \quad \triangleright \text{Case (G1)} \\
5: & \quad \text{return } (T, \tau', I' \cup \text{Nodes}(T_{\tau_1})); \\
6: & \quad \text{else} \quad \triangleright \text{Case (G2)} \\
7: & \quad \text{return } (T, \tau', I);
\end{align*}
\]

**Algorithm 5 \( \text{wPush} \)**

**Input:** \( T \in \mathcal{T} \), \( \tau = ((\tau_1, \ldots, \tau_m), w_\tau) \) a labeled tail attached to \( T \), and \( I \subseteq \text{Nodes}(T) \).

**Output:** \( T' \in \mathcal{T} \), \( \tau' \) a labeled tail attached to \( T' \), and \( I' \subseteq \text{Nodes}(T') \).

\[
\begin{align*}
1: & \quad \text{function } \text{wPush}(T, \tau, I) \\
2: & \quad (T', \tau') \leftarrow \text{Push}(T, \tau); \\
3: & \quad \text{if } \tau_1 \text{ is the right child of a node } \beta \text{ in } T \text{ then} \\
4: & \quad \text{if } w(T, \tau) \neq w(T', \tau') \text{ or } \beta \ell \leq \tau_1 \leq \tau_2 \text{ then} \quad \triangleright \text{Case (P1)} \\
5: & \quad \text{return } (T', \tau', I' \cup \text{Nodes}(T_{\tau_1})); \\
6: & \quad \text{else} \quad \triangleright \text{Case (P2)} \\
7: & \quad \text{return } (T', \tau', I); \\
8: & \quad \text{return } (T', \tau', I); \quad \triangleright \text{Case (P)}
\end{align*}
\]

Figure 12 demonstrates the execution of Algorithm 6 through an example. Uninfected tree nodes are shaded dark red, while the infected tree nodes are shaded light green. The nodes belonging to the labeled tail are in the highlighted region and are circled. The edges that constitute the (partial) tree are thickened. Our next lemma allows us to track the weight \( w(T, \tau) \).

**Lemma 4.2.** In Cases (P2) and (G2) of \( \text{wPush} \) and \( \text{wGlide} \), we have that \( w(T, \tau) = w(T', \tau') \). In Case (P1), the combined weight of the edges \( \beta \tau_2 \) and \( \tau_1 \tau_2 \) in \( T' \) is \( a_1 b_2 \) if \( \beta \ell \leq \tau_2 \) or \( a_2 b_1 \) if \( \beta \ell > \tau_2 \). Similarly, in Case (G1), the combined weight of the edges \( \beta \tau_2 \) in \( \tau' \) and \( \beta \tau_1 \) in \( T' \) is \( a_1 b_2 \) if \( \beta \ell \leq \tau_2 \) or \( a_2 b_1 \) if \( \beta \ell > \tau_2 \).

**Proof.** Note that Case (G2) applies in \( \text{wGlide} \) if and only if \( \text{std}(\beta \ell_1 \tau_1) \in \{123, 213, 231\} \), and Case (G1) applies if and only if \( \text{std}(\beta \ell_1 \tau_1) \in \{321, 312, 132\} \). Similarly, Case (P2) applies in \( \text{wPush} \) if and only if \( \text{std}(\beta \ell_1 \tau_1) \in \{321, 312, 132\} \), and Case (P1) applies if and only if \( \text{std}(\beta \ell_1 \tau_1) \in \{123, 213, 231\} \). The lemma can be checked in each of these cases. \( \square \)
Algorithm 6 ExtendedPG

Input: A labeled binary path \( v = ((v_1, \ldots, v_m), w_v) \) and PG sequence \( u \) such that \( (u, v) \) is a glued pair.

Output: \( T \in \mathcal{T}^d \) and a set \( I \subseteq \text{Nodes}(T) \) of “infected” nodes.

1: function \text{ExtendedPG}(u, v)
2: Let \( k \) be the unique nonnegative integer such that \( w_v = \mathcal{D}^k w' \) where \( w' \) is either the empty word or begins with a \( U \);
3: Let \( T \) be the labeled tree rooted at \( v_1 \) such that \( v_i \) is the right child of \( v_{i-1} \) for \( 2 \leq i \leq k + 1 \);
4: Let \( \tau := ((v_{k+1}, v_{k+2}, \ldots, v_m), w') ;
5: Let \( I := \emptyset ;
6: \text{while} \: \text{the length of} \: \tau \: \text{is} \: \geq 1 \: \text{do}
7: \quad \text{if} \: u \: \text{begins with a} \: P \: \text{then}
8: \quad \quad (T, \tau, I) \leftarrow \text{wPush}(T, \tau, I);
9: \quad \quad \text{Remove} \: P \: \text{from the front of} \: u;
10: \quad \text{else}
11: \quad \quad (T, \tau, I) \leftarrow \text{wGlide}(T, \tau, I);
12: \quad \quad \text{Remove} \: G \: \text{from the front of} \: u;
13: \quad \text{return} \: (T, I);

The next remark is very crucial as it demonstrates how infected and noninfected nodes carry information about the PG sequence applied to obtain a particular labeled tree. Furthermore, the idea described next is key to Lemma 4.9 where we show that our map \( \Phi : \mathcal{T}^d \rightarrow \mathcal{PP}^d \) defined in Definition 4.7 is indeed a bijection.

Remark 4.3. For every labeled binary path \( v \), there exists a unique PG sequence \( u \) such that \( (u, v) \) is a glued pair and \text{ExtendedPG}(u, v) \) has no infected nodes. To construct such a \( u \), we start with \( v \) and iteratively build \( u \) by running the PG algorithm and ensuring that our choice of push or glide at each step is such that it does not result in infection. Given the criteria for infecting nodes in \( \text{wPush} \) and \( \text{wGlide} \), this choice is unique. We make the preceding discussion precise. Begin with \( u \) being the empty word. At every step, if the tail is attached to the root, then apply \( \text{wPush} \) and append a \( P \) to \( u \). Otherwise, the tail is not attached to the root. In this case, apply \( \text{wPush} \) and append a \( P \) to \( u \) if \( \text{std}(\beta \tau_1 \tau_2) \in \{321, 312, 132\} \), or apply \( \text{wGlide} \) and append a \( G \) to \( u \) if \( \text{std}(\beta \tau_1 \tau_2) \in \{123, 213, 231\} \). At the end of the algorithm, append \( G^r \) to \( u \) where \( r \) is the number of nodes in the right branch of the final tree. The operations were chosen so that only Cases (P2) or (G2) apply at each step.

Next we prove some preliminary results leading up to Definition 4.7.

Lemma 4.4. Let \( (u, v) \) be a glued pair where \( v \) is a labeled binary path. Let \( T \) and \( \tau \) be the current labeled tree and labeled tail at any step while computing \text{ExtendedPG}(u, v) \), and let \( \tau_1 \) denote the initial node of \( \tau \). Then the nodes in the right and left branches of \( T \) preceding and including \( \tau_1 \) in inorder are uninfected. In particular, the first and last node of \( v \) are uninfected in the final output \text{ExtendedPG}(u, v).

Proof. If one of these nodes is infected, then it must have been the \( \tau_1 \) node in some application of (P1) or (G1), which is impossible. \( \square \)
In fact, one can establish the stronger claim that all nodes in the right branch of $T$ are uninfected. Our next lemma concerns the subgraph formed by the infected nodes in the tree output by $\text{ExtendedPG}$.

**Lemma 4.5.** Let $(u, v)$ be a glued pair where $v$ is a labeled binary path, and let $(T, I) = \text{ExtendedPG}(u, v)$. The subgraph of $T$ induced by the set $I$ of infected nodes is a disjoint union of subtrees $T_1, T_2, \ldots, T_m$ of $T$ for some $m \geq 1$. The nodes in these subtrees have the property that $\text{Nodes}(T_i)$ forms an interval in the inorder on $T$ for each $1 \leq i \leq m - 1$, and no two of these intervals are adjacent.

**Proof.** Since the algorithm $\text{ExtendedPG}$ infects entire subtrees at each stage in the algorithm, it follows that the final set of infected nodes $I$ will be a disjoint union of subtrees of $T$. The fact that the node sets of each of these subtrees forms an interval in the inorder on $T$ follows immediately from properties of inorder. It remains to show that no two of these intervals are adjacent. We will show that this is true for the current tree and infected set at any stage in the algorithm by analyzing the two routines $\text{wGlide}$ and $\text{wPush}$.

Let $T, \tau$ and $I$ be the current tree, attached tail and infected set of nodes at any step in the algorithm $\text{ExtendedPG}$. Let the subset of nodes in the right branch of $T$ preceding and
including \( \tau_1 \) in inorder be \( t_1, \ldots, t_k = \tau_1 \). By Lemma 4.4, these nodes must all be uninfected at this stage.

First we analyze when the \texttt{wGlide} routine is performed. In this case, we must have \( p \geq 2 \) as \( \tau_1 \) cannot be the root of \( T \). If Case (G2) applies, it does not alter either \( T \) or \( I \). On the other hand, if Case (G1) applies, then this operation infects nodes corresponding to the tree \( T_p \) rooted at \( t_p \). The nodes in \( T_p \) form a final segment in inorder of \( T \), and these nodes are separated from the other infected nodes by the uninfected node \( t_{p-1} \).

Next we analyze when the \texttt{wPush} routine is performed. If \( p = 1 \), then this routine does not infect any vertices in \( T \), so suppose \( p \geq 2 \). As before, let \( T_p \) be the subtree of \( T \) rooted at \( t_p \). Let \( (\tau_2, \tau_3, \ldots, \tau_{k+2}) \) be the maximal sequence of nodes in \( \tau \) starting at \( \tau_2 \) joined by \( D \) steps. If Case (P2) applies, then infected subtrees of \( T \) remain infected subtrees of \( T' \). If Case (P1) applies, then this operation infects \( T_p \). The nodes in this subtree are separated by the other infected nodes in \( T' \) by the uninfected node \( t_{p-1} \). Therefore, the new tree and set of infected nodes after any push or glide operation satisfy the claim stated in the lemma.

Before we define the map \( \Phi \), a remark on another bijection between trees and glued pairs that does not involve the PG algorithm is in order.

\textbf{Remark 4.6.} We refer the reader to \cite{51} Section 2 for a straightforward description of the inverse correspondence between trees and glued pairs, which we describe informally next. It uses an alternative definition of canopy and requires us to walk counterclockwise along the boundary of our tree and associate to it a word in an alphabet of four letters \( \{a, \bar{a}, b, \bar{b}\} \), where the choice of \( a \) and \( b \) is determined by whether we walk towards the left child or towards the right child, while the choice of \( \bar{a} \) and \( \bar{b} \) is determined by whether one is walking from a left child to its parent node or from a right child to its parent node. The subword consisting of only \( \bar{a} \) and \( \bar{b} \) gives us the PG sequence \( u \), and the subword consisting of only \( a \) and \( b \) gives us the canopy \( v \). More precisely, to obtain a word in the alphabet \( \{U, D\} \) (respectively \( \{P, G\} \)) in order to be consistent with our notation, we have to convert \( \bar{a} \) to \( U \) (respectively \( P \)) and \( b \) to \( D \) (respectively \( \bar{b} \) to \( G \)). One extends the preceding correspondence in a natural way to one between labeled trees \( T \) and glued pairs \( (u, v) \) where \( v \) is labeled. We now proceed to define the map \( \Phi : \mathcal{T}^t \to \mathcal{P}\mathcal{T}^t \).

Given a tree \( T \in \mathcal{T}^t \), let \( (u, v) \) be the corresponding glued pair such that \( v \) is labeled with the word \( \text{in}(T) \). Now run \texttt{ExtendedPG} on the input \( (u, v) \) and recover the tree \( T \) along with its set of infected nodes \( I \). Then by our two previous lemmas, the nodes in \( I \) correspond to a set of nodes in \( v \) which form nonadjacent intervals in \( v \).

\textbf{Definition 4.7.} Let \( v^{(1)}, v^{(2)}, \ldots, v^{(m)} \) be the labeled binary subpaths of \( v \) corresponding to uninfected nodes in the order in which they appear in \( v \) from left to right. Also, let \( T_1, T_2, \ldots, T_{m-1} \) be the subtrees of \( T \) corresponding to infected nodes such that \( \text{root}(T_1) \prec_i \cdots \prec_i \text{root}(T_{m-1}) \). We define the map \( \Phi \) using this decomposition as follows,

\[
\Phi(T) = (v^{(1)}, T_1, v^{(2)}, \ldots, v^{(m-1)}, T_{m-1}, v^{(m)}).
\]

In the example in Figure 12, we have \( m = 3 \) and the words \( \text{in}(v^{(1)}) \), \( \text{in}(v^{(2)}) \), and \( \text{in}(v^{(3)}) \) are 2, 134, and 55 respectively. Also, \( T_1 \) consists of a single node labeled 1 and \( T_2 \) is the green subtree consisting of three nodes labeled 342 in inorder. The decomposition of the output tree in our running example given by \( \Phi \) is shown in Figure 13.
Figure 13. The image of the tree in Figure 12 under \( \Phi \).

For good measure, we also depict the decomposition of the canopy according to the infected and uninfected nodes in Figure 14. We have thickened the edges giving the PG sequence and canopy for the infected tree(s). We are ready to complete the proof of Theorem 4.1.

**Figure 14.** The canopy decomposed according to infection.

Proof of Theorem 4.1. It is immediate from the construction that \( \text{in}(T) = \text{in}(\Phi(T)) \). Thus, it suffices to prove that \( \text{wt}(T) = \text{wt}(\Phi(T)) \) and that \( \Phi \) is a bijection, which follow from Lemmas 4.8 and 4.9 below, respectively.

For both the lemmas that follow, to emphasize the dependence on the tree \( T \), we will denote the corresponding glued pair by \((u(T), v(T))\). Furthermore, if at any point we talk about nodes \( \beta \), \( \tau_1 \) and \( \tau_2 \), the reader will benefit from revisiting Figures 7 and 8.

**Lemma 4.8.** The map \( \Phi \) is weight-preserving; that is,

\[
\text{wt}(T) = \prod_{i=1}^{m} \text{wt}(v^{(i)}) \prod_{i=1}^{m-1} \gamma_i \text{wt}(T_i).
\]

**Proof.** We first show that if \( p \) and \( q \) are consecutive nodes, read from left to right, in some \( v^{(i)} \) joined by a \( U \) step, then the weight of the left edge of \( q \) in \( T \) is the same as the weight of the edge between \( p \) and \( q \) in \( v(T) \). To this end, note that in the case under consideration \( p \) and \( q \) correspond to two uninfected consecutive nodes in \( v(T) \) joined by a \( U \) step. Since the edge joining \( p \) and \( q \) is a \( U \) step, this edge will remain adjacent to \( q \) throughout the algorithm ExtendedPG as the left edge of \( q \). Furthermore, \( p \) must be in the left subtree of \( T_q \). Since \( p \) is uninfected at the end of the algorithm, the left edge of \( q \) must only participate in the operations corresponding to Cases \((G2)\) or \((P2)\), each of which preserves the weight of this edge by Lemma 4.2. Therefore, the weight of the edge between \( p \) and \( q \) in \( v(T) \) matches that of its corresponding left edge in \( T \).

Now we consider the case where \( p \) and \( q \) are consecutive nodes joined by a \( D \) step in some \( v^{(i)} \). Then \( p \) and \( q \) will correspond to two uninfected consecutive nodes in \( v(T) \) joined by a \( D \) step, and the edge between them will remain incident with \( q \) throughout the algorithm ExtendedPG as the right edge of \( p \). Furthermore, the uninfected node \( q \) must be in the right subtree of \( T_p \). Thus, the right edge of \( p \) must only participate in weight-preserving operations throughout the algorithm, and we again conclude that the weight of the edge between \( p \) and \( q \) in \( v(T) \) matches that of its corresponding right edge in \( T \).

Therefore, the only remaining edge weights unaccounted for are the weights of the edges between the nodes \( \beta \), \( \tau_1 \), and \( \tau_2 \) at the time of the infection of the root \( \tau_1 \) of each maximal
infected subtree $T_i$. It can be seen that $\beta$ corresponds to the final node in $v(i)$ and $\tau_2$ corresponds to the initial node in $v(i+1)$. At the time of infection of $T_i$, an operation corresponding to either Case ($G1$) or Case ($P1$) involving these three nodes is performed. By Lemma 4.2, the combined weight of these two edges is subsequently $a_1b_2$ if $\beta^i \leq \tau^i_2$ or $a_2b_1$ if $\beta^i > \tau^i_2$. In either case, this weight is precisely $\gamma_i$. Furthermore, by maximality of the infected subtree, these edges must participate only in weight-preserving operations for the rest of the algorithm.

□

Lemma 4.9. The map $\Phi : T^\ell \to \mathcal{PT}^\ell$ is a bijection.

Proof. Suppose $T \in T^\ell$ and $\Phi(T) = (v(1), T_1, \ldots, v(m-1), T_{m-1}, v(m))$ for some $m \geq 1$. Since we know that $\text{in}(T) = \text{in}(\Phi(T))$, it suffices to show that the underlying tree $T$ can be recovered from $\Phi(T)$ uniquely. Then once we know the tree structure, the labeling is uniquely determined by the word $\text{in}(T)$. The key idea was hinted at in Remark 4.3. The state of a node, that is, whether it is infected or not, determines whether a $\text{wGl}ide$ or a $\text{wPush}$ was performed in the iterative construction of the PG sequence.

First, it can easily be seen from the definition of canopy that

$$v(T) = v(1)(Dv(T_1)U)v(2) \ldots v(m-1)(Dv(T_{m-1})U)v(m).$$

In Equation (4.1), we are identifying all labeled binary paths with their corresponding sequence of $U$ and $D$ steps. Next, we recover the tree $T$ itself. Since we now the canopy $v(T)$ and the infected subtrees $T_i$ given by running $\text{ExtendedPG}$, it suffices to show there is only one PG sequence $u(T)$ such that running the algorithm on $(u(T), v(T))$ yields these infected subtrees. We iteratively reconstruct $u(T)$ as follows.

First, the algorithm constructs a tree from the nodes in $v(1)$. None of these nodes can be infected, so we are forced to apply only operations corresponding to Cases ($G2$) or ($P2$) according to std($\beta^i\tau^i_1\tau^i_2$), or a push ($P$) whenever the tail is attached to the root of the current tree. This continues until the first node of $v(T_1)$ is part of the right branch of the current tree. Second, in order for $T_1$ to form a subtree in the final output tree, we are then forced to follow the operations corresponding to the PG sequence $u(T_1)$ of $T_1$ in order to construct $T_1$. Once $T_1$ is constructed, the tail will then be attached to root($T_1$) \footnote{To see this, recall that the PG algorithm terminates with $u' = Gr$ where $r$ is the number of edges in the right branch. In the case under consideration, these $r$ glides serve the purpose of gliding the attached labeled tail all the way to root($T_1$) as claimed.} so that the initial node $\tau_1 = \text{root}(T_1)$. We need the tree $T_1$ to become infected, so we are forced to apply operations corresponding to either Case ($G1$) or Case ($P1$) next, again depending on std($\beta^i\tau^i_1\tau^i_2$). After this operation, we once again have that $\tau_1$ is not part of any tree $T_i$, hence we are forced to apply only ($G2$) and ($P2$) operations until the first node of $v(T_2)$ is part of the current tree. This continues until the tail attached to the tree consists of a single node. We leave it to the reader to verify that this procedure gives an inverse to our map $\Phi$.

This concludes our proof of Theorem 4.1 from which Theorem 1.3 follows immediately. Far more importantly, we can also prove Theorem 1.4, which is what we consider in Subsection 4.2.

Remark 4.10. Note that our bijection $\Phi$ has an interesting unlabeled analogue. If we apply our bijection above to labeled trees on $n$ nodes with a fixed inorder reading word, say the identity permutation in $S_n$, then we get a bijection between $T_n$ and $\mathcal{PT}_n$, the set
of sequences of interlacing paths and trees where both are unlabeled. Of course, one can establish that \(|PT_n| = \text{Cat}_n\) by using an argument relying on generating functions.

4.2. The case of fixed canopy. For a word \(v\) in \(\{U, D\}\) of length \(n - 1\), recall that \(G_{n,v}\) is the weight generating function summing over all labeled binary trees with canopy \(v\). In this subsection, we prove Theorem 1.4 which states that \(G_{n,v}\) is Schur-positive. In fact, we give an explicit expansion for \(G_{n,v}\) in terms of ribbon Schur functions using marked trees, and Theorem 4.1 is crucial to this end.

We need the notion of the boundary word of a tree. Given \(T \in \mathcal{T}_n\), for every bivalent node, label its left edge by a \(D\) and right edge by a \(U\). Then, for every right (respectively left) univalent node, label its right (respectively left) edge by a \(D\) (respectively \(U\)). The boundary word of \(T\), denoted by bw\((T)\), is defined recursively as follows. Read the label of the left edge of the root (if it exists), followed by the boundary word of the left subtree, then read the label of the right edge of the root, and finally read the boundary word of the right subtree. The notion of boundary word extends naturally to labeled trees and marked trees as well.

Remark 4.11. Observe that in our definition of the boundary word, we give left edges from left univalent nodes the label \(U\) while left edges from bivalent nodes get the label \(D\). This choice, although incongruous, has been made so that in going from Equation (4.5) to Equation (4.6) in the proof of Theorem 4.12, the indeterminates \(a_1\), \(a_2\) continue to index weights on right edges, while the variables \(b_1\), \(b_2\) continue to index weights on left edges. One could, in theory, write our results by making the uniform choice of all left edges labeled \(D\) and all right edges labeled \(U\), but we will refrain from doing so.

Given a marked tree \(\hat{T} \in \hat{T}_n\), we call \(T \in T_n^{\ell}\) compatible with \(\hat{T}\) if and only if the following conditions are satisfied.

1. \(\text{sh}(T) = \text{sh}(\hat{T})\).
2. The right edge of any bivalent or right univalent node \(p\) in \(T\) is a strict right descent if and only if the corresponding node in \(\hat{T}\) is marked.
3. The left edge of any left univalent node \(p\) in \(T\) is a strict left descent if and only if the corresponding node in \(\hat{T}\) is marked.

Let \((\beta^{(1)}, \ldots, \beta^{(k)})\) be the decomposition sequence of \(\hat{T}\). Upon reading trees in \(T_n^{\ell}\) compatible with \(\hat{T}\) in preorder, we obtain

\[
\sum_{T \text{ compatible with } \hat{T}} x_T = r_{\beta^{(1)}} r_{\beta^{(2)}} \ldots r_{\beta^{(k)}}.
\]

Equation (4.2) is useful in establishing our next result, which proves Theorem 1.4 from the introduction. As the proof is slightly involved, the reader will benefit from referring to Figure 15 at various points in the proof.

Theorem 4.12. For \(n \geq 1\) and \(v\) a word of length \(n - 1\) in \(\{U, D\}\), we have that

\[
G_{n,v} = \sum_{\substack{T \in T_n \\text{bw}(T) = v}} \text{wt}(T) r_{\beta^{(1)}} r_{\beta^{(2)}} \ldots r_{\beta^{(k)}},
\]

where \((\beta^{(1)}, \ldots, \beta^{(k)})\) is the decomposition sequence of \(\hat{T}\). Hence, \(G_{n,v}\) is Schur-positive.
Proof. Recall that
\[ G_{n,v} = \sum_{T \in \mathcal{T}_n^v} \text{wt}(T)x_T, \]
where \( \mathcal{T}_n^v \) denotes the set of labeled trees on \( n \) nodes with canopy \( v \). By using our weight-preserving bijection \( \Phi \) from Theorem 4.1, we can rewrite Equation (4.3) as
\[ G_{n,v} = \sum_{T \in \mathcal{T}_n^v} \text{wt}(\Phi(T))x_T. \]

Thus, we have that \( G_{n,v} \) is equal to the weight generating function summing over those elements \( (v^{(1)}, T_1, \ldots, T_{m-1}, v^{(m)}) \in \mathcal{P}_n \mathcal{F}^\ell \) that satisfy
\[ v = v^{(1)}(Dv(T_1)U)v^{(2)} \ldots v^{(m-1)}(Dv(T_{m-1})U)v^{(m)}. \]
Here, as in the proof of Lemma 4.9, we are abusing notation and identifying the labeled binary paths with their corresponding sequence of \( U \) and \( D \) steps. Our proof of the claim proceeds by performing a series of transformations starting from a fixed labeled tree. In fact, all transformations except the last are bijections. Our final transformation gives a marked tree whose weight and boundary word coincide with the weight and canopy of our original labeled tree. We now give details.

Fix a labeled tree \( T \in \mathcal{T}_n^v \) to begin with, and let \( \Phi(T) = (v^{(1)}, T_1, \ldots, T_{m-1}, v^{(m)}) \). We transform \( \Phi(T) \) into a labeled right-leaning tree whose edges are also labeled to keep track of the canopy. More precisely, given \( \Phi(T) \in \mathcal{P}_n \mathcal{F}^\ell \), we recursively associate with it a tree \( \hat{T} \) as follows. Suppose the trees \( \hat{T}_i \) have already been constructed for \( 1 \leq i \leq m-1 \). Let the length of the right branch of \( \hat{T} \) be equal to the sum of the lengths of the \( v^{(i)} \) for \( 1 \leq i \leq m \), and let the sequence of labels along the right branch from root to terminal node be
\[ \text{in}(v^{(1)}) \text{in}(v^{(2)}) \ldots \text{in}(v^{(m)}). \]
Let \( pq \) be an edge in the right branch where \( q \) is the right child of \( p \). If \( p \) and \( q \) correspond to consecutive nodes in some \( v^{(i)} \), label the edge \( U \) or \( D \) depending on whether it forms a \( U \) or \( D \) step in \( v^{(i)} \). Otherwise, \( p \) corresponds to the final node of some \( v^{(i)} \), and \( q \) corresponds to the initial node of \( v^{(i+1)} \), for some \( 1 \leq i \leq m-1 \). In this case, label the edge with a \( U \). Additionally, attach root \( \hat{T}_i \) as a left child of \( p \) and label the edge \( p \text{root}(\hat{T}_i) \) with a \( D \). Repeating this for all \( 1 \leq i \leq m-1 \) gives us \( \hat{T} \). It is clear how to recover \( \Phi(T) \) from it. Note also how the canopy \( v \) is encoded by the edge labeling on \( \hat{T} \).

To describe the weight of \( \Phi(T) \) in terms of \( \hat{T} \), we need some notation. Denote the set of bivalent nodes whose right edge is a weak ascent (respectively strict descent) by \( \text{Bi}^a(\hat{T}) \) (respectively \( \text{Bi}^d(\hat{T}) \)). Also, denote the set of right univalent nodes whose right edge is a weak ascent (respectively strict descent) labeled with a \( U \) by \( \text{Un}^u_1(\hat{T}) \) (respectively \( \text{Un}^u_1(\hat{T}) \)). Finally, denote the set of right univalent nodes whose right edge is a weak ascent (respectively strict descent) labeled with a \( D \) by \( \text{Un}^u_1(\hat{T}) \) (respectively \( \text{Un}^u_1(\hat{T}) \)). Then we have that
\[ \text{wt}(\Phi(T)) = (a_1b_2)^{|\text{Bi}^a(\hat{T})|}(a_2b_1)^{|\text{Bi}^d(\hat{T})|}a_1^{||\text{Un}^u_1(\hat{T})||}b_2^{||\text{Un}^u_2(\hat{T})||}a_2^{||\text{Un}^u_1(\hat{T})||}b_1^{||\text{Un}^u_1(\hat{T})||}. \]

In the next step, we transform \( \hat{T} \) into a labeled tree \( T' \) (without edge labels) such that the boundary word of \( T' \) is \( v \) according to the following procedure. For a right univalent node \( p \) in \( \hat{T} \), if the right edge of \( p \) connecting it to \( q \) is labeled \( U \), then transform it into a left edge (and make \( \hat{T}_q \) the left subtree of \( p \)), and subsequently omit all edge labels to
obtain \( T' \). One can verify that \( \text{bw}(T') \) indeed coincides with the canopy \( v \) of \( T \) and that this transformation is also invertible.

Consider now the unique marked tree \( \hat{T} \) such that \( T' \) is compatible with it. This marked tree also satisfies \( \text{bw}(\hat{T}) = v \). In terms of statistics on marked trees, we can rewrite Equation (4.5) as

\[
\text{wt}(\Phi(T)) = (a_1 b_2) |\text{Bi}^u(\hat{T})||\text{Bi}^m(\hat{T})| a_1 |\text{Uni}^m(\hat{T})| b_1 |\text{Uni}^{im}(\hat{T})| a_2 |\text{Uni}^{im}(\hat{T})| b_2 |\text{Uni}^u(\hat{T})|,
\]

and the expression on the right-hand side is precisely \( \text{wt}(\hat{T}) \). Thus, by performing this series of transformations from \( T \) to \( \hat{T} \), we obtain a bijection

\[
T_{n,v}^f \leftrightarrow \bigcup_{\substack{T \in T_n^f \text{ compatible with } \hat{T}}} \{ T \in T_n^f \text{ compatible with } \hat{T} \}.
\]

Given how the weights behave under this bijection, we can rewrite Equation (4.4) as

\[
G_{n,v} = \sum_{T \in T_n^f} \sum_{\text{T compatible with } \hat{T}} \text{wt}(\hat{T})x_T.
\]

From Equation (4.2), the claim follows. \( \square \)

The various stages in the proof of Theorem 4.12 are shown in Figure 15, beginning with the labeled tree of Figure 12. We adhere to the notation established in that proof in labeling our objects. The double-headed arrows correspond to bijections. Observe that the weight of the final marked tree \( \hat{T} \) agrees with the initial labeled tree \( T \). Furthermore, note that \( \text{bw}(\hat{T}) \) equals the canopy of \( T \) (both equaling \( \text{DUDUDDUDD} \) in turn). For the instance under discussion, our techniques in Theorem 4.12 prove that the set of all labeled trees with canopy \( v = \text{DUDUDDUDD} \) that map to the marked tree \( \hat{T} \) at the bottom of Figure 15 contributes the Schur-positive summand \( \text{wt}(\hat{T})r_{15r_{11}r_{1}r_{1}} = (a_1 b_2)(a_2 b_1)^2 a_2^2 b_2 r_{15r_{11}r_{1}r_{1}} \) to \( G_{10,v} \).

We discuss the explicit Schur expansion next. Given a composition \( \alpha = (\alpha_1, \ldots, \alpha_k) = n \), consider compositions \( \beta^{(1)}, \ldots, \beta^{(k)} \) such that \( \beta^{(i)} \models \alpha_i \) for \( 1 \leq i \leq k \). Let \( \gamma = \beta^{(1)} \cdots \cdot \beta^{(k)} \). By using the commutative analogue of Lemma 2.4 followed by the expansion in Proposition 2.2, we obtain the equality

\[
r_{\beta^{(1)}} r_{\beta^{(2)}} \cdots r_{\beta^{(k)}} = \sum_{Y \in \text{SYT}(n)_{\text{set}(\gamma) \subseteq \text{des}(Y) \subseteq \text{set}(\gamma)}} s_{\text{sh}(Y)}.
\]

We are using the letter \( Y \) to denote our SYTs instead of \( T \), which is more conventional, since we have reserved \( T \) for trees. We are abusing notation and using \( \text{sh}(Y) \) to denote the shape of the tableau \( Y \). Furthermore, \( \text{SYT}(n) \) denotes the set of all SYTs with \( n \) boxes. Using Equation (4.8), we obtain the following explicit Schur expansion as a corollary to Theorem 4.12.

**Corollary 4.13.** For \( n \geq 1 \) and \( v \) a word of length \( n - 1 \) in \( \{U, D\} \), we have

\[
G_{n,v} = \sum_{\substack{T \in T_n \text{ compatible with } \hat{T} \\text{ and } \text{bw}(T) = v}} \text{wt}(\hat{T}) \left( \sum_{Y} s_{\text{sh}(Y)} \right),
\]
where the inner sum ranges over all $Y \in \text{SYT}(n)$ satisfying $\text{set}(\bar{c}(\hat{T})) \setminus \text{set}(c(\hat{T})) \subseteq \text{des}(Y) \subseteq \text{set}(\bar{c}(\hat{T}))$.

**Example 4.14.** Consider $n = 3$ and $v = DU$. The six marked trees with boundary word is $DU$ are shown in Figure 16 along with their weights. By Theorem 4.12, we have the following expansion for $G_{3,DU}$.

\[
G_{3,DU} = a_1b_2r_3 + a_2b_2r_{12} + a_1b_1r_{21} + a_2b_1r_{111} + a_1b_2r_2 \cdot r_1 + a_2b_1r_{11} \cdot r_1
\]
\[
= (a_1b_2r_3 + a_2b_2r_{12} + a_1b_1r_{21} + a_2b_1r_{111}) + (a_1b_2(r_3 + r_{21}) + a_2b_1(r_{12} + r_{111})).
\]

Furthermore, Corollary 4.13 implies that

\[
G_{3,DU} = a_1b_2s_3 + (a_2b_2 + a_1b_1)s_{21} + a_2b_1s_{111} + a_1b_2s_3 + (a_1b_2 + a_2b_1)s_{21} + a_2b_1s_{111}.
\]
Upon setting \( a_1 = a_2 = b_1 = b_2 = 1 \) in Example 4.14, we obtain the symmetric function \( 2(s_3 + 2s_{21} + s_{111}) \). This is twice the Frobenius characteristic of the regular representation of \( S_3 \). It is easy to see that in the general case we have that \( G_{n,v}(x; 1, 1, 1, 1) \) is \( |\text{Tam}(v)| \) times the Frobenius characteristic of the regular representation of \( S_n \).

To conclude this subsection, we give a combinatorial description of the coefficient of \( r_\alpha \) for a given \( \alpha \vdash n \) in the expansion of \( G_{n,v} \) given by Theorem 4.12. Using the commutative image of the equality in Lemma 2.4 and the disjoint union property in Equation (2.5), we have the following corollary.

**Corollary 4.15.** Given \( \alpha \vdash n \), the coefficient of \( r_\alpha \) in \( G_{n,v} \) is given by

\[
\sum_{T \in \mathcal{T}_n, \text{bw}(T) = v} \text{wt}(\hat{T}).
\]

For instance, from the six marked trees in Figure 16, the second and sixth trees from the left are all the elements of \( \mathcal{T}_{1,2} \) and have boundary word \( DU \). Thus, the coefficient of \( r_{(1,2)} \) in \( G_{3,DU} \) is \( a_2b_2 + a_2b_1 \).

In the introduction, we claimed that the coefficient of \( r_{(1^n)} \) in \( G_n \) is the homogenized Narayana polynomial in the variables \( a_2 \) and \( b_1 \). To see this, we use the fact that \( G_n = \sum G_{n,v} \) as \( v \) varies over all words of length \( n - 1 \) in \( \{U, D\} \). Since \( \mathcal{T}_{1^n} \) consists of marked trees on \( n \) nodes where all internal nodes are marked, we conclude that the coefficient of \( r_{(1^n)} \) in \( G_n \) is

\[
\sum_{T \in \mathcal{T}_n} (a_2b_1)^{|\text{Bi}(T)|}a_2^{\text{Unii}(T)}b_1^{\text{Unid}(T)}.
\]

This equals \( \sum_{T \in \mathcal{T}_n} r(T)^a(T) \) where \( r(T) \) and \( l(T) \) denote the number of right and left edges in \( T \) respectively. It is well-known that the Narayana number \( \text{Nar}(n, k) \) counts the number of trees in \( \mathcal{T}_n \) with \( k \) right edges, and we are done. A similar argument reveals that the coefficient of \( r_{(n)} \) is the homogenized Narayana polynomial in the variables \( a_1 \) and \( b_2 \).

**4.3. Noncommutative version.** Let \( x_i \) for \( i \geq 1 \) be noncommutative variables. Given a word \( w = w_1 \ldots w_n \in \mathbb{P}^+ \), define the noncommutative monomial \( x_w \) to be \( x_{w_1} \ldots x_{w_n} \). Let \( s \) and \( t \) be parameters which do not commute with the \( x_i \). For \( w \in \mathbb{P}^+ \), let \( x_w(s, t) \) be the noncommutative monomial given by inserting \( s \) for each ascent \( w_i \leq w_{i+1} \) and \( t \) for each descent \( w_i > w_{i+1} \), in between \( x_{w_i} \) and \( x_{w_{i+1}} \) in the monomial \( x_w \). For instance, if \( w = 21131 \), then \( x_w(s, t) = x_2tx_1sx_1sx_3tx_1 \).

Using our weight-preserving map \( \Phi \) to obtain an element of \( \mathcal{PT}^\ell \) from a given labeled tree along with the fact that the PG algorithm respects inorder reading words, we obtain the following noncommutative analogue of the functional equation in Theorem 1.3.

**Corollary 4.16.** If

\[
G := \sum_{T \in \mathcal{T}_n} \text{wt}(T)x_{\text{in}(T)},
\]

then

\[
G = \sum_{w \in \mathbb{P}^+} x_w(a_1b_2G + a_1 + b_2, a_2b_1G + a_2 + b_1).
\]

We use the commutative image of the functional equation for \( G \) in the previous corollary to obtain an expansion for \( G \) in terms of ribbon Schur functions. The details are omitted as they are very similar to those in Theorem 4.12 except that the edge-labeling on the
right-leaning trees does not play a role anymore. This is to be expected as we are taking the sum over all $G_{n,v}$ to obtain $G_n$, and $G = \sum_{n \geq 1} G_n$.

**Theorem 4.17.** For $n \geq 1$, we have that

$$G_n = \sum_{T \in \mathcal{T}_n} \text{mwt}(T) r_{\beta(1)} r_{\beta(2)} \cdots r_{\beta(k)},$$

where $(\beta(1), \ldots, \beta(k))$ is the de-composition sequence of $T$.

Our expansion in Theorem 4.17 suggests the weights in the expansion of $G_{n,v}$ given in Theorem 4.12 group together nicely to give an expansion involving $a_1 b_2$, $a_1 + b_2$, $a_2 b_1$, $a_2 + b_1$. In particular, we infer that the coefficient of $r_{\{1^n\}}$ in $G_n$, the homogenized Narayana polynomial in variables $a_2$ and $b_1$, is in fact a polynomial with positive integer coefficients in the “variables” $a_2 b_1$ and $a_2 + b_1$. Thus, what we are observing is a generalization of the phenomenon of $\gamma$-nonnegativity, and this phenomenon is perhaps best understood by studying a particular symmetric boolean decomposition of the lattice of noncrossing partitions, drawing inspiration from work of Simion-Ullman [58], Blanco-Petersen [11], and Hersh [37].

4.4. A symmetric boolean decomposition of the lattice of noncrossing partitions. We recall the relevant definitions briefly. A partition $\pi$ of $[n]$ is a collection of pairwise disjoint nonempty subsets $B_1, \ldots, B_k$ whose union is $[n]$. We write this as $\pi := B_1/B_2/\cdots/B_k$. The subsets $B_1, \ldots, B_k$ are the blocks of $\pi$, and the number of blocks is denoted by $b_k(\pi)$. We depict a partition of $[n]$ using its arc diagram: We consider $n$ nodes representing the integers $1$ through $n$ from left to right, and connect two nodes by an arc if the corresponding integers belong to the same block. If $i$ and $j$ belong to the same block in $\pi$, then we denote this equivalence relation by $i \sim_\pi j$. We omit the subscript if the partition is clear from context. The set of partitions of $[n]$, denoted by $\Pi_n$, can be endowed with the structure of a graded lattice by defining a partial order as follows. Given partitions $\sigma$ and $\tau$, we say that $\sigma \leq_{\Pi_n} \tau$ if each block in $\sigma$ is contained in a block in $\tau$. In particular, a partition $\tau$ covers a partition $\sigma$ in $\Pi_n$ if $\tau$ is obtained by merging two distinct blocks in $\sigma$. We say that $\sigma$ is finer than $\tau$ or equivalently that $\tau$ is coarser than $\sigma$. The rank of $\pi \in \Pi_n$ is given by $n - b_k(\pi)$. The partition of $[n]$ into singleton sets gives the unique minimal element in $\Pi_n$, and the partition of $[n]$ consisting of a single block gives the unique maximal element.

A partition $\pi$ of $[n]$ is said to be noncrossing if there do not exist $1 \leq a < b < c < d \leq n$ such that $a \sim c$, $b \sim d$ and $a \not\sim b$. The set of noncrossing partitions of $n$, denoted by $\text{NC}(n)$, inherits a graded lattice structure from that on $\Pi_n$ and shares the rank function. For the many interesting properties of $\text{NC}(n)$, the reader is referred to the beautiful survey by Simion [57] and references therein. For a more recent survey on the relevance of $\text{NC}(n)$ in various areas of mathematics, the reader is referred to [41].

The cardinality of $\text{NC}(n)$ is $\text{Cat}_n$, and this is easily revealed by the following bijection between $\text{NC}(n)$ and $\mathcal{T}_n$ which is a special case of one due to Edelman [21] Theorem 1.1]. Given $T \in \mathcal{T}_n$, let $v_1, \ldots, v_n$ be its vertices in preorder. Define $\text{nc}(T)$ to be the finest partition of $[n]$ with the property that distinct positive integers $1 \leq i < j \leq n$ are in the same block if $v_j$ is the left child of $v_i$. An example of Edelman’s bijection is given in Figure [17].

An interesting feature of $\text{NC}(n)$ is that it possesses a symmetric boolean decomposition, which is a generalization of the notion of symmetric chain decomposition and informally means that $P$ can be decomposed into a finite number of boolean lattices with center
of symmetry around the middle rank (or ranks) of \( P \). Part of the interest in symmetric boolean decompositions of posets is because they imply the \( \gamma \)-nonnegativity of the rank generating function. Note that \( \gamma \)-nonnegativity in turn implies the unimodality of the coefficients of the rank generating function. We refer the reader to [17] Chapter 4 for the precise definitions of a symmetric boolean decomposition of a finite ranked poset and what it means for a polynomial to be \( \gamma \)-nonnegative. Our expansion of \( G_n \) in Theorem [4.17] with weights polynomial in \( a_1b_2, a_1 + b_2, a_2b_1, a_2 + b_1 \) is a reflection of a particular symmetric boolean decomposition of \( \text{NC}(n) \). Each \( G_{n,v} \) corresponds to the contribution of a boolean lattice in this decomposition, and the sum of the \( G_{n,v} \) as canopy \( v \) varies corresponds to the contribution of \( \text{NC}(n) \). We give the details next.

Given \( T \in \mathcal{T}_n \), we have the following decomposition of \( \text{Nodes}(T) \) into disjoint subsets:

\[
\text{Nodes}(T) = \text{Bi}(T) \bigcup \text{Uni}^i(T) \bigcup \text{Uni}^r(T) \bigcup \text{Term}(T).
\]

Let \( v_1, \ldots, v_n \) be the nodes of \( T \) in preorder. Observe that \( v_n \) is a terminal node of \( T \). The decomposition in Equation (4.9) allows us to associate a word \( w = w_1 \cdots w_{n-1} \) in the alphabet \( \{b, e, l, r\} \) with \( T \) as follows. For \( 1 \leq i \leq n-1 \), we define \( w_i := b \) \( (e, l, r \) respectively) if \( v_i \) is a bivalent node (terminal or end node, left univalent node, and right univalent node respectively). We can define this word using the noncrossing partition corresponding to \( T \) as well. Let \( \text{nc}(T) = B_1/\cdots/B_k \) for some positive integer \( k \). Let \( \max(T) = \{\max B_1, \ldots, \max B_k\} \) and \( \min(T) = \{\min B_1, \ldots, \min B_k\} \). Then

\[
w_i = \begin{cases} 
  b & i \not\in \max(T), i + 1 \in \min(T) \\
  e & i \not\in \max(T), i + 1 \in \min(T) \\
  r & i \not\in \max(T), i + 1 \in \max(T) \\
  l & i \sim i + 1
\end{cases}
\]

We call \( w \) the \( SU \)-word associated to \( T \), or equivalently, to \( \text{nc}(T) \). An example of a \( SU \)-word is given in Figure [17].

**Remark 4.18.** The reader is invited to compare our approach to that of Simion-Ullman [58] where they give a constructive proof a symmetric chain decomposition of \( \text{NC}(n) \). Their proof, in fact, gives a symmetric boolean decomposition. They attach a word in the alphabet \( \{b, e, l, r\} \) which is slightly different from ours. In particular, observe that the definition of when \( w_i = l \) or \( r \) is switched. The word attached by Simion-Ullman possesses an interesting asymmetry where whether \( i \) belongs to the same block as 1 plays a role in determining \( w_i \). On the other hand, our definition is quite symmetric. Another symmetric boolean decomposition (different from ours and Simion-Ullman’s) for \( \text{NC}(n) \) is given by Blanco-Petersen [11]. We are following them in calling the word associated to a noncrossing partition the \( SU \)-word, even though their word is defined differently from ours.

![Figure 17](image-url)

**Figure 17.** A tree and its corresponding noncrossing partition and \( SU \)-word.

We make note of some basic aspects of the \( SU \)-word associated with \( T \in \mathcal{T}_n \), similar to [58] Observations 2.3 and 2.4.
Lemma 4.19. The number of instances of $b$, $e$, $r$, and $l$ in the $SU$-word $w$ associated to $T$ is $|Bi(T)|$, $|\text{Term}(T)| - 1$, $|\text{Uni}^l(T)|$, and $|\text{Uni}^r(T)|$ respectively. Furthermore the subword comprising all instances of $b$ and $e$ in $w$ forms a well-bracketed expression, which implies that $|Bi(T)| = |\text{Term}(T)| - 1$. Finally, the number of blocks in $\text{nc}(T)$ is $|\text{Term}(T)| + |\text{Uni}^r(T)|$.

Proof. The first claim is immediate. Observe that for any tree $T \in \mathcal{T}_n$, there is a unique complete binary tree $T'$ homeomorphic to it which one obtains by shrinking a path comprising univalent nodes except for its end nodes. Our claims clearly hold for $T'$. All that remains to be checked is that they continue to hold upon subdividing edges, that is, inserting univalent nodes. This is easily verified.

Given a right (respectively left) univalent node $v$ in a tree $T \in \mathcal{T}_n$, let $\phi_v(T)$ be the tree obtained by making the right (respectively left) subtree the left (respectively right) subtree. The $SU$-word associated with $\phi_v(T)$ is determined by replacing the $r$ (or $l$) corresponding to the node $v$ in the $SU$-word of $T$ with $l$ (respectively $r$). We call the action of $\phi_v$ on a tree $T$ a flip on node $v$. Two trees are flip-equivalent if one can be obtained from the other by performing flips. In terms of noncrossing partitions, a flip amounts to merging two blocks if $v$ happened to be a right univalent node, otherwise it amounts to splitting a block into two blocks. Thus, applying a flip on a right univalent node is equivalent to going up the lattice $NC(n)$ by one rank. For instance, the noncrossing partition corresponding to the tree $T$ in Figure 17 is $\{1, 6, 9\}/\{2, 4\}/\{3\}/\{5, 7, 8\}$. The noncrossing partition corresponding to $\phi_{v_4}(T)$ is $\{1, 6, 9\}/\{2, 4, 5\}/\{3\}/\{7, 8\}$ while the noncrossing partition corresponding to $\phi_{v_7}(T)$ is $\{1, 6, 9\}/\{2, 4\}/\{3\}/\{5\}/\{7\}/\{8\}$. Note that $v_4$ is a right univalent node while $v_7$ is a left univalent node.

Remark 4.20. Our notion of flip is essentially the unlabeled analogue of the “modified Foata-Strehl action” studied by Bränden [14] and independently by Getu, Shapiro and Woan [33]. This action is a variant of a group action on permutations originally introduced by Foata and Strehl [23].

Observe that if $T \in \mathcal{RT}_n$, then the $SU$-word of $T$ does not contain any instances of the letter $l$. Applying a sequence of flips on only the right univalent nodes, which is equivalent to changing some subset of the instances of $r$ in the $SU$-word of $T$ into $l$, and considering the corresponding noncrossing partitions gives a copy of some boolean lattice inside $NC(n)$. We denote this lattice by $\mathcal{B}_T$. As $T$ varies over set of right-leaning trees on $n$ nodes, we get boolean lattices $\mathcal{B}_T$ in $NC(n)$ which are all disjoint. Thus, to establish our claim that flips give rise to a symmetric boolean decomposition, it suffices to show that the boolean lattices $\mathcal{B}_T$ have the right center of symmetry.

To this end, observe that $\text{nc}(T)$ for $T \in \mathcal{RT}_n$ is the unique minimal element in the boolean lattice $\mathcal{B}_T$. The rank of $\text{nc}(T)$ is $n - |\text{Term}(T)| - |\text{Uni}^r(T)|$ by Lemma 4.19. Let $T'$ be the tree obtained by performing flips on all right univalent nodes of $T$. Thus, $T'$ is a tree without any right univalent nodes. It is, in fact, the reflection of the original tree about a vertical line passing through the root of the original tree. The unique maximal element in $\mathcal{B}_T$ is $\text{nc}(T')$ and its rank is $n - |\text{Term}(T')| = n - |\text{Term}(T)|$. Thus, the sum of the ranks of $\text{nc}(T)$ and $\text{nc}(T')$ is $2n - (2|\text{Term}(T)| + |\text{Uni}^r(T)|)$ which equals $n - 1$. This completes the proof.

We finish this subsection by explaining how the above decomposition explains the weights present in Theorem 4.17. Since $G_n = \sum G_{n,v}$ as the canopy $v$ varies over all words of length
n – 1 in \{U, D\}, using the expansion in Theorem 4.12 we obtain the following equality.

\( G_n = \sum_{T \in \mathcal{T}_n} \text{wt}(\hat{T}) r_{\beta(1)} r_{\beta(2)} \cdots r_{\beta(k)}. \)  

Here, the compositions \( \beta(i) \) are as in the statement of Theorem 4.17. Note that we can naturally lift the notion of flips to marked trees and talk about boolean lattices \( \mathcal{B}_L \) associated with marked trees. Therefore, the sum in Equation (4.10) can be rewritten as

\[
G_n = \sum_{T \in \mathcal{R} \mathcal{T}_n} \sum_{R \in \mathcal{B}_P} \text{wt}(\hat{T}') r_{\beta(1)} r_{\beta(2)} \cdots r_{\beta(k)}
\]

Since flips do not alter either the number of bivalent vertices, marked or unmarked, or the decomposition sequence, we obtain the expansion in Theorem 4.17 by using the fact that \( \sum_{T' \in \mathcal{B}_P} \text{wt}(T') = m \text{wt}(\hat{T}). \)

4.5. An improved extended PG algorithm. In the proof of Theorem 4.12 we associate an edge-labeled right-leaning labeled tree with an element of \( \mathcal{P} \mathcal{F}_\ell \) as follows. Beginning with \( T \in \mathcal{T}_{n,v}' \), using our extended PG algorithm we construct a unique element \( \Phi(T) = (v^{(1)}, T_1, \ldots, v^{(m-1)}, T_{m-1}, v^{(m)}) \in \mathcal{P} \mathcal{F}_\ell \). This is subsequently mapped to a unique element \( \hat{T} \). The procedure for computing \( \hat{T} \) is recursive and involves computing \( \Phi(T_i) \) for \( 1 \leq i \leq m - 1 \). At this point, it appears that we have to run our extended PG algorithm multiple times to compute the \( \Phi(T_i) \). The aim of this subsection is to show that this is not the case and that a single run of the extended PG algorithm suffices to compute \( \hat{T} \) starting from \( T \). Given that all the results we present next are contingent on an elementary analysis of the PG algorithm, we keep our presentation brief.

Consider the following minor change in the routines \( \text{wPush} \) and \( \text{wGlide} \): In Cases (\( \mathcal{P}1 \)) and (\( \mathcal{G}1 \)), instead of infecting the entire subtree rooted at \( \tau_1 \), we only infect the node \( \tau_1 \). We refer to these modified routines as \( \text{mwPush} \) and \( \text{mwGlide} \) respectively. Using these new routines instead of the old ones in ExtendedPG results in a variation we call ModifiedExtendedPG. Note that the tree with infection output by ModifiedExtendedPG immediately yields what ExtendedPG would yield. We simply need to consider those infected nodes that are not proper descendants of any other infected nodes and subsequently

\[
\text{Figure 18. The symmetric boolean decomposition of NC(4) given by flips.}
\]
think of the subtree rooted at such nodes to be entirely infected. This gives the forest of infected subtrees used to associate a unique element of $\mathcal{PT}^\ell$ with $T$.

An analysis of when $\text{ModifiedExtendedPG}$ infects a node gives us the following lemma.

**Lemma 4.21.** Consider a tree $T \in \mathcal{T}_n^\ell$, and let $y \in \text{Nodes}(T)$ be such that it does not belong to either the left branch or the right branch of $T$ (recall that none of the nodes on the left or right branch can be infected). The node $y$ is infected upon the execution of algorithm $\text{ModifiedExtendedPG}$ if and only if one of the following criteria hold.

1. $y$ is the right child of a node $x$, $z$ is the smallest node greater than $y$ that is not a descendant of $y$, and $\text{std}(x^\ell y^\ell z^\ell) \in \{132, 312, 321\}$.
2. $y$ is the left child of a node $z$, $x$ is the greatest node less than $y$ that is not a descendant of $y$, and $\text{std}(x^\ell y^\ell z^\ell) \in \{231, 213, 123\}$.

All our comparison of nodes are performed using the order $\preceq_i$ given by inorder on $T$.

Comparing the two points in Lemma 4.21 with the depiction of Glide and Push routines in Figures 7 and 8 tells us that the first type of infection is the result of a glide while the second type of infection is the result of a push. Figure 19 shows the result of executing our modified algorithm.

![Figure 19](image-url)

**Figure 19.** The infected nodes are shaded light blue (respectively light orange) if their infection is the result of a glide (respectively push).

Let $\Phi(T) = (v^{(1)}, T_1, \ldots, v^{(m-1)}, T_{m-1}, v^{(m)}) \in \mathcal{PT}^\ell$ for some $m \geq 1$. Upon infecting nodes according to the criteria stated in Lemma 4.21, note that the infected nodes that are not proper descendants of any other infected node are the roots of the subtrees $T_i$ for $1 \leq i \leq m - 1$. To compute $\Phi(T_i)$ we first need to return the infected nodes on the left and right branch of $T_i$ back to being uninfected. Every other node that was infected in the original tree continues to be infected after execution of our algorithm $\text{ModifiedExtendedPG}$. This follows from Lemma 4.21. Thus, the infected nodes occurring in the execution of $\text{ModifiedExtendedPG}$ on the original labeled tree $T$ carry all the information required to compute $\Phi(T_i)$ for all $1 \leq i \leq m - 1$. Now we can recursively build the element $\hat{T}$ as outlined in the proof of Theorem 4.12.

The shrewd reader might have observed that this modified algorithm can be optimized further, especially as we “disinfect” the left and right branches of the $T_i$. In fact, the only infected nodes that really play a role are those that do not lie on the left or right branch of a subtree $T_y$ where $y$ is an infected node. Figure 20 demonstrates this fact by decomposing a labeled tree into a forest of subtrees where all trees but one are rooted at infected nodes satisfying the condition earlier in this paragraph. Each of these trees is then read in inorder.
and the word so obtained forms a path that goes solely right in $\hat{T}$. This concludes the analysis of our weighted generalization of the PG algorithm.

Figure 20. The bijection between labeled trees and edge-labeled right-leaning labeled trees using our optimized PG algorithm.

5. $\mathfrak{S}_n$-modules from deformations of Coxeter arrangements

We briefly introduce terminology pertaining to hyperplane arrangements. For a detailed introduction, we refer the reader to [52, 61]. A hyperplane arrangement is a finite collection of affine hyperplanes in a vector space. Let $\mathcal{A}$ be a hyperplane arrangement in a finite-dimensional vector space $V$ over $\mathbb{R}$. We denote the number of connected components of $V - \bigcup_{H \in \mathcal{A}} H$, also called regions of $\mathcal{A}$, by $r(\mathcal{A})$. We denote the set of regions of $\mathcal{A}$ by Regions($\mathcal{A}$). We provide a representation-theoretic meaning to some of the equalities in the introduction relating specializations of $G(x; a_1, a_2, b_1, b_2)$ to the number of regions in various Coxeter deformations, focusing on the semiorder and Linial arrangements in particular. While there is an obvious $\mathfrak{S}_n$-action on regions of the semiorder arrangement $I_n$, the question of finding one on regions of the Linial arrangement $L_n$ is a bit more subtle.

5.1. Semiorder arrangements. Recall that the semiorder arrangement $I_n$ is the hyperplane arrangement in $\mathbb{R}^n$ given by the hyperplanes $H_{ij} : x_i - x_j = 1$ for $1 \leq i \neq j \leq n$. Note that this set of hyperplanes is stable under the natural action of $\mathfrak{S}_n$. This implies that Regions($I_n$) inherits an action of $\mathfrak{S}_n$. We show that the Frobenius characteristic of this action is a specialization of $G_n$. Figure 21 demonstrates this for $I_3$. In this case, we can compute the Frobenius characteristic of this action to be $h_3 + 2h_{21} + 2h_{111}$. In terms of ribbon Schur functions, this may be written as $5r_3 + 3(r_{21} + r_{12}) + 2r_{111}$. Note that as ribbon Schur functions do not form a basis for the ring of symmetric functions, this expansion in terms of ribbon Schur functions is not unique.

Recall that our expansion for $G(x; a_1, a_2, b_1, b_2)$ in terms of ribbon Schur functions involves coefficients that are polynomials in $a_1b_2$, $a_1 + b_2$, $a_2b_1$, and $a_2 + b_1$. Consider the case $a_1 = b_2 = 1$, $a_2 = \zeta_6$, and $b_1 = \overline{\zeta_6}$ where $\zeta_6$ is a primitive sixth root of unity and $\overline{\zeta_6}$ denotes its complex conjugate. Then we have that $a_1b_2$, $a_2 + b_1$, and $a_2b_1$ all equal 1, whereas $a_1 + b_2$ equals 2. Thus, for this specialization, we have $\text{mwt}(\hat{T}) = 2^{\text{Univ}(\hat{T})}$ for $\hat{T} \in \mathcal{RT}_n$. 


Using Theorem 4.17 along with Lemma 2.4 we obtain

\begin{equation}
G_n(x; 1, \zeta_6, \bar{\zeta}_6, 1) = \sum_{T \in \mathcal{RT}_n} 2^{|\text{Un}^u(T)|} \left( \sum_{\delta \in \left( \mathcal{E}(T), \bar{\mathcal{E}}(T) \right)} r_\delta \right).
\end{equation}

While it is not immediate from the expansion in Equation (5.1), it turns out that the coefficient of $r_\alpha$ on the right-hand side of Equation (5.1) depends solely on $\ell(\alpha)$. This hints at a combinatorially interesting expansion of $G_n(x; 1, \zeta_6, \bar{\zeta}_6, 1)$ in terms of the Frobenius characteristics of certain remarkable characters of symmetric groups known as Foulkes characters.

For $0 \leq k \leq n - 1$, let $F_{n,k}$ denote the symmetric function defined to be the sum of all ribbon Schur functions indexed by compositions with exactly $k + 1$ parts, that is,

$$F_{n,k} = \sum_{\alpha \vdash n, \ell(\alpha) = k+1} r_\alpha.$$  

Note that the dimension of the $S_n$-module $\chi_{n,k}$ corresponding to $F_{n,k}$ is $A_{n,k}$, the Eulerian number enumerating the number of permutations in $S_n$ with $k$ descents. The $\chi_{n,k}$, known as the Foulkes characters, were introduced by Foulkes [25] in his study of descents in permutations and have many interesting properties. They show up in various areas such as counting permutations by descents and cycle types [32], enumerating alternating permutations according to cycle type [59], analysis of the carrying process [19, 44]. The notion of Foulkes character has been generalized to other reflection groups in [42].

For $m \geq 1$, let $\text{Mot}_m$ denote the $m$th Motzkin number which we define to be the number of trees in $\mathcal{RT}_m$. The reader should note that this is slightly different from the usual definition, which is that the $m$th Motzkin number is the number of trees in $\mathcal{RT}_{m+1}$. Let $M(x)$ denote the generating function for the Motzkin numbers,

$$M(x) = \sum_{m \geq 1} \text{Mot}_m x^m,$$
and let $C(x)$ denote the generating function for the Catalan numbers,

$$C(x) = \sum_{m \geq 1} \text{Cat}_m x^m.$$  

It is easy to combinatorially establish the relation

$$(5.2) \quad M \left( \frac{x}{1-x} \right) = C(x),$$

as elements of $T_n$ can be obtained by performing the following operation on a subset of nodes in a right-leaning tree $T$: If $v$ is a node with right subtree $T_w$, then we replace $v$ with a path that goes left and make $T_w$ the right subtree of the last node on this path. Equation (5.2) implies the following result that will come handy later.

$$(5.3) \quad M(e^x - 1) = C(1 - e^{-x}).$$

By decomposing binary trees into the root and the left and right subtrees, which might be potentially empty, we see that $C(x)$ satisfies the functional equation

$$C(x) = x(1 + C)^2.$$  

By using Equation (5.2) again, we see that $M(x)$ satisfies the functional equation

$$(5.4) \quad M(e^x - 1) = C(1 - e^{-x}).$$

**Theorem 5.1.** For $n \geq 1$, we have the following expansions

$$G_n(x; 1, \zeta_6, \overline{\zeta_6}, 1) = \sum_{\alpha \vdash n} \text{Mot}_\ell(\alpha) h_\alpha = \sum_{j=1}^{n} \left( \sum_{k=0}^{n-j} \binom{n-j}{k} \text{Mot}_{n-k} \right) F_{n,j-1}.$$  

**Proof.** Throughout this proof, let $H = \sum_{i \geq 0} h_i$. To obtain the $h$-expansion for $G_n$, we use the functional equation for $G$ from Theorem 1.1. When $a_1 = b_2 = 1$, $a_2 = \zeta_6$ and $b_1 = \overline{\zeta_6}$, we get that

$$(5.4) \quad G = (1 + G + G^2)(H - 1).$$

A comparison of the functional equation in Equation (5.4) with that satisfied by the ordinary generating function $M(x)$ for the Motzkin numbers reveals that

$$G = \sum_{m \geq 1} \text{Mot}_m (H - 1)^m$$  

$$(5.5) \quad = \sum_{m \geq 1} \sum_{\ell(\alpha) = m} \text{Mot}_m h_\alpha.$$  

Focusing on terms indexed by compositions of size $n$ in Equation (5.5) yields the first equality in the statement of the theorem. To obtain the expansion in terms of ribbon Schur functions, we utilize the fact that $h_\alpha = \sum_{\alpha \leq \beta} r_\beta$. Then the last equality in Equation (5.5) can be transformed to

$$G = \sum_{m \geq 1} \sum_{\ell(\alpha) = m} \text{Mot}_m \sum_{\alpha \leq \beta} r_\beta$$  

$$= \sum_{m \geq 1} \sum_{\beta = m} r_\beta \left( \sum_{\alpha \leq \beta} \text{Mot}_\ell(\alpha) \right)$$  

$$= \sum_{m \geq 1} \sum_{\beta = m} r_\beta \left( \sum_{k=0}^{m-\ell(\beta)} \binom{m - \ell(\beta)}{k} \text{Mot}_{m-k} \right).$$
Note that the innermost sum is only dependent on $\ell(\beta)$. Collecting terms corresponding to compositions of a fixed size and grouping them according to their lengths gives us the second equality in the statement of the theorem. □

Note that $G_3(x; 1, \zeta_6, \zeta_6, 1) = \mathrm{Mot}_1 h_3 + \mathrm{Mot}_2 (h_{12} + h_{21}) + \mathrm{Mot}_3 h_{111} = h_3 + 2h_{21} + 2h_{111}$, which is the Frobenius characteristic of the $S_3$-action on $I_3$ as shown in Figure 21. To compute the Frobenius characteristic in the general case we will need the following result.

**Lemma 5.2** ([49, Lemma 7.6]). Let $\sigma \in S_n$ be a permutation with $k$ cycles. Then the number of regions in $I_n$ fixed by $\sigma$ is equal to the number of regions in $I_k$.

We remark here that the statement in Lemma 5.2 differs slightly from that in [49] as Postnikov-Stanley consider the hyperplane arrangement $I_n$ projected onto the hyperplane $x_1 + \cdots + x_n = 0$ as their definition of the semiorder arrangement. This induces a harmless shift in indices and does not affect the mathematical content.

**Theorem 5.3.** $G_n(x; 1, \zeta_6, \zeta_6, 1)$ is the Frobenius characteristic of the action of $S_n$ on Regions($I_n$).

**Proof.** Our proof uses the fact that the cycle index series of the $S_n$-action on Regions($I_n$) is also the Frobenius characteristic. In view of Lemma 5.2 we find that the cycle index series is given by

$$Z_n = \sum_{\lambda \vdash n} r(I_{\ell(\lambda)}) \frac{p_\lambda}{z_\lambda}$$

$$= \sum_{k=1}^{n} r(I_k) \sum_{\lambda \vdash n \atop \ell(\lambda) = k} \frac{p_\lambda}{z_\lambda}.$$ 

Let $Z := \sum_{n \geq 1} Z_n$. Then we have that

$$Z = \sum_{n \geq 1} \sum_{k=1}^{n} r(I_k) \sum_{\lambda \vdash n \atop \ell(\lambda) = k} \frac{p_\lambda}{z_\lambda}$$

$$= \sum_{k \geq 1} r(I_k) \left( \sum_{j \geq 1} \frac{p_j}{j} \right)^k.$$ 

(5.6)

Now let $H = \sum_{i \geq 0} h_i$. Then we have $\sum_{j \geq 1} p_j/j = \log H$. Using this in Equation (5.6) gives

$$Z = \sum_{k \geq 1} r(I_k) \frac{(\log H)^k}{k!}.$$ 

(5.7)

We have the expansion

$$\frac{(\log H)^k}{k!} = \sum_{m \geq 0} (-1)^{m-k} \text{Stir}(m, k) \frac{(H - 1)^m}{m!}$$

where $(-1)^{m-k} \text{Stir}(m, k)$ is the signed Stirling number enumerating permutations in $S_m$ having exactly $k$ cycles in their cycle factorization. Using this equality, we can rephrase
Equation 5.7 as
\[
Z = \sum_{k \geq 1} r(I_k) \sum_{m \geq 0} (-1)^{m-k} \text{Stir}(m, k) \frac{(H - 1)^m}{m!} 
\]
(5.8)
\[
= \sum_{m \geq 1} \frac{(H - 1)^m}{m!} \sum_{k=1}^{m} (-1)^{m-k} r(I_k) \text{Stir}(m, k),
\]
where, in changing the order of summation, we have used the fact that \(\text{Stir}(0, k) = 0\) for \(k \geq 1\). By [62, Theorem 2.3], the following equality holds.

\[
\sum_{m \geq 1} r(I_m) \frac{x^m}{m!} = C(1 - e^{-x}).
\]
(5.9)

Equation 5.3 states that \(M(e^x - 1) = C(1 - e^{-x})\), and hence the left hand side of Equation 5.9 also equals \(M(e^x - 1)\). This implies that \(m! \text{Mot}_m = \sum_{k=1}^{m} (-1)^{m-k} r(I_k) \text{Stir}(m, k)\)
and using this in Equation 5.8, we conclude that
\[
Z = \sum_{m \geq 1} \text{Mot}_m (H - 1)^m = G.
\]
(5.10)
The second equality in Equation 5.10 follows from Equation 5.5. This in turn implies that the cycle index series \(Z\) is equal to \(G_n(x; 1, 6, 6, 1)\), thereby finishing the proof. \(\square\)

Clearly, the dimension of the module corresponding to the \(S_n\)-action on Regions\((I_n)\) equals \(r(I_n)\). By applying the exponential specialization \(ex\) to \(G(x; 1, 6, 6, 1)\) we may obtain an expression for \(r(I_n)\) which we omit as it is not more concise than known formulae. By [62, Theorem 2.1] this expression also coincides with the number of labeled semiorders on \([n]\).

Remark 5.4. An alternative to the earlier specialization is the following. Set \(a_2 = b_1 = 1\), \(a_1 = 6\), and \(b_2 = 6\). We then obtain
\[
G_n(x; 6, 1, 1, 6) = \sum_{\mathcal{T} \in \mathcal{RT}_n} 2^{\left|\text{Uni}^m(\mathcal{T})\right|} \left( \sum_{\delta \in [\mathcal{E}(\mathcal{T}), \overline{\mathcal{E}(\mathcal{T})}]} r_{\delta} \right).
\]
(5.11)

Similar arguments as before together with the functional equation for the generating function of the Catalan numbers give the expansion
\[
G_n(x; 6, 1, 1, 6) = \sum_{k=1}^{n} \left( \sum_{j=0}^{n-k} (-1)^j \binom{n-k}{j} \text{Cat}_{n-j} \right) F_{n,k-1}.
\]
(5.12)

It is not obvious that the coefficients in the expansion in Equation 5.12 are positive integers, but this follows from comparing it to the expression coming from right-leaning trees in Equation 5.11.

We conclude this subsection with a generalization of Theorem 5.3. Given a positive integer \(p\), define the \(p\)-semiorder arrangement \(I_{n,p}\) to be the hyperplane arrangement in \(\mathbb{R}^n\) defined by the hyperplanes \(x_i - x_j = \pm 1, \pm 2, \cdots, \pm p\) for \(1 \leq i < j \leq n\). As before, the symmetric group \(S_n\) acts on Regions\((I_{n,p})\) and one can ask about the corresponding Frobenius characteristic. We provide a brief description of its computation next, omitting
details. Let \( C_p(x) = \sum_{m \geq 1} \text{Cat}_{m,p} x^m \) denote the generating function for the Fuss-Catalan numbers \( \text{Cat}_{m,p} := 1/(pm+1)((p+1)^m/m) \). These numbers enumerate the rooted plane \((p + 1)\)-ary trees on \( m \) nodes, the set of which we denote by \( \mathcal{T}_m^p \). Let \( \text{Mot}_{m,p} \) denote the cardinality of the set \( \mathcal{R}\mathcal{T}_m^p \) consisting of \((p + 1)\)-ary trees on \( m \) nodes with the property that every internal node has at least one child. Note that for a \((p + 1)\)-ary tree, an internal node has up to \( p + 1 \) children, yet there is a still a well-defined notion of left and right child provided either exists. Consider \( M_p(x) := \sum \text{Mot}_{m,p} x^m \). As was the case earlier, we have that

\[
M_p \left( \frac{x}{1-x} \right) = C_p(x)
\]

which implies \( M_p(e^x - 1) = C_p(1 - e^{-x}) \).

Crucially for us, Lemma 5.2 continues to hold for \( \mathcal{I}_{n,p} \) as well. Thus, we can repeat the cycle index series computation of Theorem 5.3. We only need the relation analogous to Equation 5.9. Indeed, by [62, Theorem 2.3], we have that

\[
\sum_{m \geq 1} r(\mathcal{I}_{m,p}) \frac{x^m}{m!} = C_p(1 - e^{-x}).
\]

In arriving at the above equality we have used the fact that the number of regions in the \( p \)-Catalan arrangement defined by the hyperplanes \( x_i - x_j = 0, \pm 1, \ldots, \pm p \) for \( 1 \leq i < j \leq m \) is given by \text{Cat}_{m,p} (see [62, Section 2] or [1, Section 5]). Thus, we obtain the following theorem whose \( p = 1 \) case is Theorem 5.3.

**Theorem 5.5.** For \( p \geq 1 \), the Frobenius characteristic of the action of \( S_n \) on Regions(\( \mathcal{I}_{n,p} \)) is given by \( M_p(H - 1) \), where \( H = \sum_{i \geq 0} h_i \). Additionally, it expands positively in terms of the \( F_{n,k} \).

5.2. Linial arrangements and local binary search trees. We turn our attention to studying the Linial arrangement \( \mathcal{L}_n \) and defining an \( S_n \)-action on its regions. Observe that, unlike in the case of the semiorder arrangement, the symmetric group \( S_n \) does not stabilize the set of hyperplanes defining \( \mathcal{L}_n \). Hence it is not immediate how to construct an \( S_n \)-action on Regions(\( \mathcal{L}_n \)). Another well-studied arrangement with the property that the set consisting of its defining hyperplanes is not stable under the \( S_n \)-action is the Shi arrangement \( S_n \). In spite of this limitation, one can define an \( S_n \)-action on Regions(\( S_n \)) by using one of the many ways to index its regions by parking functions of length \( n \), and then using the natural \( S_n \)-action on them. Drawing inspiration from this, we use certain labeled trees that we call Bernardi trees instead of parking functions to index regions of \( \mathcal{L}_n \), and then construct a natural \( S_n \)-action on Bernardi trees to derive one on Regions(\( \mathcal{L}_n \)).

First, we need to understand how the symmetric function \( G_n \) relates to \( \mathcal{L}_n \).

The problem of enumerating \( r(\mathcal{L}_n) \) was first considered by Postnikov [48], inspired by a question of Linial and Ravid. He showed that \( r(\mathcal{L}_n) \) equals the number of intransitive trees and gave a bijection between intransitive trees and local binary search trees. It is the latter that carries a description in terms of ascents and descents in labeled trees. Consider the case where \( a_1 = b_2 = 0 \). In the setting of the introduction, this corresponds to considering labeled trees that only have left ascents and right descents, both of which correspond to strict inequalities. We refer to such trees as local binary search trees (henceforth LBS trees) after Stanley. We will construct another subset of \( \mathcal{T}_n^\ell \) that is equinumerous with standard LBS trees on \( n \) nodes and use it to define an \( S_n \)-action on Regions(\( \mathcal{L}_n \)). It is worth emphasizing that the same subset of trees has been considered by Bernardi [10] to
solve the long-standing problem of finding a bijection between Linial regions and standard LBS trees.

By writing $G_n$ as a sum of the $G_{n,v}$ where $v$ runs over all possible canopies of length $n - 1$ and using Theorem $4.12$ we have

$$G_n(x; 0, a_2, b_1, 0) = \sum_{T \in T_n^{(1^m)}} (a_2 b_1)^{|\text{Bi}^m(T)|} a_2^{|\text{Uni}^m(T)|} b_1^{|\text{Uni}^m(T)|} r_{\beta(1)} \cdots r_{\beta(k)},$$

where $(\beta(1), \ldots, \beta(k))$ is the de-composition sequence associated with $\hat{T}$. For any $\hat{T} \in T_n^{(1^m)}$, the de-composition sequence consists solely of compositions of the form $(1^m)$ for some choice of positive integer $m$. Since $r_{1^m} = e_m$ for any positive integer $m$, Equation (5.14) can be written as

$$G_n(x; 0, a_2, b_1, 0) = \sum_{T \in T_n} a_2^r(T) b_1^{\ell(T)} e_c(T)$$

where $r(T)$ and $\ell(T)$ denote the number of right edges and left edges in $T$ respectively. Since $r(T) + \ell(T) = n - 1$ for any $T \in T_n$, we can set $b_1 = 1$. Additionally we set $a_2$ equal to an indeterminate $q$. We are ready to describe our $\mathfrak{S}_n$-module whose graded Frobenius characteristic is $G_n(x; 0, q, 1, 0)$.

A Bernardi tree is a standard labeled binary tree satisfying the condition that every internal node has a label that is greater than the label of its right child provided it exists, otherwise it is greater than the label of its left child. Let $T_n^B$ denote the set of Bernardi trees on $n$ nodes. Observe how the definition of these trees naturally comes from applying the homomorphism $\text{ex}$ to the right-hand side of Equation (5.15). Since $G_n(x; 0, a_2, b_1, 0)$ is the generating function corresponding to LBS trees on $n$ nodes, applying $\text{ex}$ to the left-hand side establishes that the number of standard LBS trees on $n$ nodes is equal to the cardinality of $T_n^B$. Figure 22 shows all trees in $T_3^B$.

![Figure 22. The Bernardi trees on 3 nodes.](image)

**Remark 5.6.** For the reason behind the name Bernardi trees, we refer the reader to [10] Example 1.1] where Bernardi gives a bijection between $T_n^B$ and Regions($\mathcal{L}_n$). In the notation of [10], the bijection associates to $T \in T_n^B$ the region of $\mathcal{L}_n$ defined by the inequalities $x_i - x_j < 1$ where $1 \leq i < j \leq n$ and either $\text{drift}(i) \leq \text{drift}(j)$ or $\text{drift}(i) = \text{drift}(j) + 1$ and $i < p_j$. Here, drift($v$) as the number of ancestors of $v$ (including $v$) that are right children. Figure 23 shows the regions of $\mathcal{L}_3$ indexed by the corresponding Bernardi trees according to this bijection.

For $T \in T_n^B$, let $v_1 \prec_p \cdots \prec_p v_n$ be the nodes of $T$ in preorder. Among these nodes, let $v_1 \prec_p \cdots \prec_p v_n$ be all the terminal nodes and set $i_0 := 1$. Observe that the preorder reading word $\text{pre}(T) = v_1^\ell \cdots v_n^\ell$ can be factorized as $W_1 \cdots W_k$, where for $j \geq 1$ we have

$$W_j = v_{a+1}^\ell v_{a+2}^\ell \cdots v_{b-1}^\ell v_b^\ell,$$

where we have set $a := i_{j-1}$ and $b = i_j$ for ease of notation. Given the definition of $T_n^B$, we know that each $W_j$ is strictly decreasing when read from left to right. For the tree
\( T \in T^B_9 \) in Figure 24 on the left, the shaded regions give us the words \( W_1 = 731, W_2 = 86, W_3 = 952 \) and \( W_4 = 4 \) and their concatenation gives us \( \text{pre}(T) = 731 \ 86 \ 952 \ 4 \).

The factorization of \( \text{pre}(T) \) as \( W_1 \ldots W_k \) as described earlier allows us to define an obvious \( S_n \)-action on \( T^B_n \) as follows. Given \( \sigma \in S_n \), define \( \sigma(W_i) \) to be the word obtained by replacing every letter in \( W_i \) by its image under \( \sigma \), and then sorting the resulting word so that it is strictly decreasing when read from left to right. Now define \( \hat{\sigma}(W) \) to be \( \sigma(W_1) \cdots \sigma(W_k) \), and let \( \sigma(T) \) be the unique labeled tree such that \( \text{sh}(\sigma(T)) = \text{sh}(T) \) and \( \text{pre}(\sigma(T)) = \hat{\sigma}(W) \). It is easily checked that \( \sigma(T) \in T^B_n \) and that the above description gives an \( S_n \)-action on \( T^B_n \). Furthermore, if we denote by \( T^B_{n,k} \) the set of Bernardi trees with exactly \( k \) right edges, then the above action also gives one on \( T^B_{n,k} \).

In Figure 24 for the tree \( T \) on the left, we have \( \sigma(T) \) on the right where \( \sigma = (38) \) in cycle notation. Note that for the instance under discussion, we have \( \hat{\sigma}(W) = 871 \ 63 \ 952 \ 4 \).

Thus, \( \sigma(T) \) is the unique tree whose preorder reading word is 871639524 and whose underlying shape is that of \( T \). Had we chosen \( \sigma = (37) \), then \( \sigma(T) \) would be \( T \) itself.

Let \( \mathbb{C}T^B_n \) (respectively \( \mathbb{C}T^B_{n,k} \)) denote the vector space generated by formal linear combinations of trees in \( T^B_n \) (respectively \( T^B_{n,k} \)). Then both \( \mathbb{C}T^B_n \) and \( \mathbb{C}T^B_{n,k} \) are \( S_n \)-modules,
and we have the following equality of $S_n$-modules
\[ \mathbb{C}T_n^B = \bigoplus_{k=0}^{n-1} \mathbb{C}T_{n,k}^B. \]
Thus, we can think of $\mathbb{C}T_n^B$ as being graded by the number of right edges.

**Theorem 5.7.** The graded Frobenius characteristic of the $S_n$-module $\mathbb{C}T_n^B$ is given by
\[ \sum_{T \in T_n} q^{r(T)} h_c(T). \]

Using Bernardi’s bijection between $T_n^B$ and Regions($L_n$), our $S_n$-action lifts to an action on Regions($L_n$) whose graded Frobenius characteristic is given by $\omega(G_n(x; 0, q, 1, 0))$.

We leave it to the reader to verify that $\omega(G_n(x; 0, q, 1, 0)) = G_n(x; q, 0, 0, 1)$, and hence, the graded Frobenius characteristic of our $S_n$-action on Regions($L_n$) is given by $G_n(x; q, 0, 0, 1)$. Note also that by applying the homomorphism $e_t$ to $G_n(x; 1, 0, 0, 1)$, we can compute the cardinality of $T_n^B$. This gives another formula for $r(L_n)$, which is not in any way more concise than known formulae and hence is omitted.

5.3. **Intransitive trees and $\gamma$-nonnegativity.** We turn our discussion to another notion of importance both in algebraic combinatorics and discrete geometry, that of $\gamma$-nonnegativity. We already alluded to it during our discussion of symmetric boolean decompositions of the lattice of noncrossing partitions. We say that a polynomial $P(t) = \prod_{j=0}^{[\frac{n}{2}]} \gamma_{n,j} t^j (1 + t)^{n-2j}$ where $\gamma_{n,j} \geq 0$. We refer the reader to [47, Chapter 4] for a book exposition and [4] for further problems regarding $\gamma$-nonnegativity. For a recent survey on the relevance and prevalence of $\gamma$-nonnegativity and real-rootedness of polynomials arising naturally in combinatorics, the reader is referred to [14]. Our focus here is the connection between intransitive trees of Postnikov [48] and regions of Linial arrangements.

Following Postnikov [48], an intransitive tree on $n$ nodes is a tree whose nodes are labeled with distinct positive integers from $[n]$ such that the label of a node is either greater than labels of its neighbors, in which case we call it a right vertex, or is less than the labels of its neighbors, in which case we call it a left vertex. Note that the trees considered by Postnikov are neither plane nor rooted, and they do not have to be binary. We refer the reader to [48] for further details on the terminology. Let $f_n(t) := \sum_{k \geq 1} f_{nk} t^k$ where $f_{nk}$ is the number of intransitive trees on $[n+1]$ with $k$ right vertices. Consider the generating function
\[ F(t, x) = \sum_{n \geq 0} f_n(t) \frac{x^n}{n!}. \]
By [48] Theorem 3, we have that $F := F(t, x)$ satisfies the functional equation
\[ F(F + t - 1) = te^{x(F + t)}. \]
We note that in Postnikov’s statement of the above functional equation, the roles of $x$ and $t$ are switched. Consider the functional equation satisfied by $\tilde{B} := 1 + a_2 B(x; 0, a_2, 1, 0)$. 

\[ F(t, x) = \sum_{n \geq 0} f_n(t) \frac{x^n}{n!}. \]
From Equation (1.1) we have

\[ r(B + a_2 - 1) = a_2 e^{x(B + a_2)}. \]

By comparing Equations (5.18) and (5.19), we obtain the following proposition.

**Proposition 5.8.** For \( n \geq 1 \), the number of intransitive trees on \([n+1]\) with \( k \) right vertices equals the number of standard LBS trees on \([n]\) with \( k - 1 \) right descents.

Upon setting \( b_1 = 1 \) in Equation (5.15), we obtain

\[ G_n(x; 0, a_1, 1, 0) = \sum_{T \text{ LBS trees on } n \text{ nodes}} a_r(T) x_T = \sum_{T \in R_n} a_2^{[Bi(T)]} (1 + a_2)^{|Uni(T)|} e_{c(T)}, \]

where the last equality follows from Theorem 4.17. Applying the homomorphism \( e_x \) to both sides implies

\[ \sum_{T \text{ a standard LBS tree on } n \text{ nodes}} a_r(T) x_T = \sum_{T \in R_n} a_2^{[Bi(T)]} (1 + a_2)^{|Uni(T)|} \left( \frac{n}{c(T)} \right), \]

where \( \binom{n}{\alpha} \) denotes the multinomial coefficient \( \frac{n!}{\alpha_1! \cdots \alpha_k!} \) for composition \( \alpha = (\alpha_1, \ldots, \alpha_k) = n \).

Thus, we have established that the distribution of right descents over standard LBS trees is \( \gamma \)-nonnegative. Combining this with Proposition 5.8, we have the following theorem.

**Theorem 5.9.** For \( n \geq 1 \), the distribution of right descents over the set of standard LBS trees on \( n \) nodes is \( \gamma \)-nonnegative. Equivalently, the polynomials \( f_n(t) \) in Equation (5.17) considered by Postnikov are \( \gamma \)-nonnegative. As a corollary, we have that the sequence of coefficients \( \{f_{nk}\}_{k \geq 1} \) is unimodal.

In the spirit of the theme in [14], we offer the following stronger conjecture.

**Conjecture 5.10.** The polynomials \( f_n(t) \) are real-rooted with all roots negative for all \( n \geq 1 \). In particular, the coefficients of \( f_n(t) \) form a log-concave sequence.

### 6. Final remarks

We conclude this article with some avenues to pursue and various remarks containing isolated results.

1. We briefly consider some special cases that were not covered in Section 5. Our first specialization concerns the Shi arrangement \( \mathcal{S}_n \). Set \( a_1 = a_2 = b_2 = 1 \) and \( b_1 = 0 \). This corresponds to considering trees that do not have any right ascents. If we restrict our attention to such trees on \( n \) nodes endowed with the standard labeling, then there are \((n+1)^{n-1}\) of them (see [40] or [45, p. 7]). Additionally, the functional equation in Theorem 1.1 is equivalent to

\[ (1 + G) = \sum_{i \geq 0} h_i (1 + G)^i. \]

Solving this functional equation [35, Section 4.1] implies that \( G_n(x; 1, 1, 0, 1) \) is the Frobenius characteristic of the natural \( \mathfrak{S}_n \)-action on parking functions of length \( n \) (see also [63, Proposition 2.2]), which we denote by \( PF_n \). Using one of the many bijections between Regions(\( \mathcal{S}_n \)) and parking functions of length \( n \), we may obtain an \( \mathfrak{S}_n \)-action on Regions(\( \mathcal{S}_n \)). Thus, we can think of \( G_n(x; 1, 1, 0, 1) \) as the Frobenius characteristic of this action.
Another case of interest is when $a_1 = b_1 = 1$ and $a_2 = b_2 = 0$. We have that $G_{n,v}(x; 1, 0, 1, 0)$ is the generating function over increasing binary trees on $n$ nodes with canopy $v$. We leave it to the reader to verify that Theorem 4.12 implies that $G_{n,v}(x; 1, 0, 1, 0)$ is a single ribbon Schur function $r_\alpha$ where $\alpha$ is the composition associated with the canopy $v = v_1 \cdots v_{n-1}$ define as follows. Consider the subset $S$ of $[n-1]$ where $i \in S$ if and only if $v_i = U$, and define $\alpha := \text{comp}(S)$. That there are $n!$ standard increasing binary trees on $n$ nodes is simply a reflection of the fact that $\sum_{\alpha} r_\alpha$ is the Frobenius characteristic of the regular representation of $\mathfrak{S}_n$.

Another specialization is when $a_1 = a_2 = 1$ and $b_1 = b_2 = 0$. We have $G_n(x; 1, 1, 0, 0)$ is the generating function over words of length $n$. This is well-known to be the Frobenius characteristic of the regular representation of $\mathfrak{S}_n$. It is also the Frobenius characteristic of the coinvariant algebra of the polynomial ring on $n$ generators, which is isomorphic to the regular representation as a symmetric group module. If we keep the $a_2$ parameter, this is equivalent to recording descents in the generating function over words. In [34], Gnedin-Gorin-Kerov construct a filtration of symmetric group submodules of the coinvariant algebra, defined in terms of its Garsia-Stanton descent basis [27]. They show that $G_n(x; 1, q, 0, 0)$ is the Frobenius characteristic of the associated graded module corresponding to this filtration. For general results pertaining to multigraded Frobenius characteristics of the coinvariant algebra according the descents, the reader is referred to [35]. It would be interesting to draw connections between this work and ours.

In the introduction, for every subset $A$ of $S := \{-1, 0, 1\}$, we associated the arrangement in $\mathbb{R}^n$ consisting of all hyperplanes $x_i - x_j = a$ for all $a \in A$ and $1 \leq i < j \leq n$. Let us denote this arrangement by $\mathcal{A}_A$. We know that for each choice of $A \subset S$, there is a specialization of $a_1, a_2, b_1, b_2$ such that $B_n(a_1, a_2, b_1, b_2)$ equals $r(\mathcal{A}_A)$. For this choice of specialization, let $G_n^A$ (respectively $G_n^B$) denote the corresponding generating function over labeled binary trees with $n$ nodes (respectively, those with canopy $v$). Consider distinct subsets $X$ and $Y$ of $S$ such that $X \subset Y$. Our results show that the obvious inequality $r(\mathcal{A}_X) < r(\mathcal{A}_Y)$ is a reflection of a curious Schur-positivity phenomenon. Indeed, Theorem 4.12 implies $G_n^Y - G_n^X$ is Schur-positive. In fact, if neither $X$ nor $Y$ corresponds to the semioriented arrangement, then the stronger claim that $G_n^Y - G_n^X$ is Schur-positive is true by Theorem 4.12. To emphasize that this is non-obvious, consider $Y = \{0, 1\}$ and $X = \{1\}$, so that $\mathcal{A}_Y$ and $\mathcal{A}_X$ correspond to Shi and Linial arrangements respectively. Then $G_2^Y - G_2^X = (h_1 + h_2) - (2h_2) = h_1 - h_2$, which is not $h$-positive, but is Schur-positive. More generally, we have that $G_n^Y$, which is equal to $PF_n$, and $G_n^X$ are both sums of certain homogeneous symmetric functions over an indexing set of Catalan objects. It would be interesting to obtain an alternative combinatorial interpretation for the Schur-positivity of $G_n^Y - G_n^X$.

We return to the topic of $\gamma$-nonnegativity. Equation (1.3) states that $G_n$ can be written in terms of ribbon Schur functions with coefficients in $\mathbb{N}[a_1 + b_2, a_1 b_2, a_2 + b_1, a_2 b_1]$. Suppose that $G_n(x; a_1, a_2, b_1, b_2) = \sum_{\alpha \vdash n} c_{\alpha} r_\alpha$ where the $c_{\alpha}$ belong to $\mathbb{N}[a_1 + b_2, a_1 b_2, a_2 + b_1, a_2 b_1]$. Applying the homomorphism $ex$ yields

$$B_n(a_1, a_2, b_1, b_2) = \sum_{\alpha \vdash n} c_{\alpha} |\{\pi \in \mathfrak{S}_n \mid \text{Des}(\pi) = \text{set}(\alpha)\}|.$$

Note that this equality involves a sort of “double” $\gamma$-nonnegativity: one in terms of $a_1 b_2$ and $a_1 + b_2$, and the other in terms of $a_2 b_1$ and $a_2 + b_1$. It would be interesting
to identify other instances of such a phenomenon. There has been a great amount of interest in $\gamma$-nonnegativity recently, in no small measure due to Gal’s conjecture [26] which is essentially a statement in topology, and the implications it has for the Charney-Davis conjecture [17]. For more work in this direction, see [50]. We do not know of any topological motivation for our “double” $\gamma$-nonnegativity, and we intend to explore this in the future.

(4) For a fixed positive integer $n$, there are $2^{n-1}$ words $v$ of length $n-1$ in the alphabet $\{U, D\}$ and as many symmetric functions $G_{n,v}$. These are too many to form a basis for the degree $n$ component of the ring of symmetric functions with coefficients in $\mathbb{Q}[a_1, a_2, b_1, b_2]$. Lifting the expansion in Theorem 4.12 na"{i}vely to the setting of NSym by replacing the ribbon Schur functions with the noncommutative ribbon Schur functions gives us a noncommutative analogue of $G_{n,v}$, which we denote by $G_{n,v}$. The dimension of the degree $n$ component of NSym, denoted by NSym$^n$, is $2^{n-1}$, and it would be interesting to know if the $G_{n,v}$ for varying $v$ of length $n-1$ give rise to a basis for NSym$^n$. If yes, one might also be interested in other combinatorial aspects such as a combinatorial interpretation to the structure coefficients in the expansion of $G_{n,v} \cdot G_{m,v}$ in the conjectural $G$-basis. Since the $G_{n,v}$ belong to NSym, it might also be interesting to understand them from the perspective of the representation theory of the 0-Hecke algebra in type $A$. Finally, is there a combinatorial Hopf algebra structure hiding behind our setup?

(5) Our work involved the construction of generalized Tamari lattices by Préville-Ratelle–Viennot, who in turn were motivated by considerations of Bergeron concerning diagonal harmonics. It would be interesting to relate our work to the study of higher diagonal harmonics. On a related note, glued pairs considered in this article are in bijection with parallelogram polyominoes, and in [8] the combinatorics of the labeled analogues was studied. We are yet to explore the connection between the work in [8] and ours. Another natural question to consider is that of extending our results to labeled $k$-ary trees for $k \geq 3$. Note that while we have an analogue of $B(x; a_1, a_2, b_1, b_2)$ for $k$-ary trees where $k \geq 3$, it is unclear how to obtain an analogue of $G_n(x; a_1, a_2, b_1, b_2)$ as there does not appear to be natural definition for ascent-descent statistics in this generalized setting. In the case $k \geq 3$, we do not have access to procedures such as the PG algorithm, and we expect any approach to involve a different set of techniques.

Appendix A. A Noncommutative Lift of Gessel’s Functional Equation

The functional equation in Theorem 1.1 can be manipulated to obtain the one below, which is more amenable to a recursive computation.

(A.1) \[ G = (1 + a_2 G)(1 + b_1 G) \sum_{n \geq 1} ((a_1 b_2 - a_2 b_1)G + a_1 - a_2 - b_1 + b_2)^{n-1} h_n \]

Consider now the following noncommutative analogue of the functional equation [A.1].

(A.2) \[ G = \sum_{n \geq 1} h_n (1 + a_2 G)(1 + b_1 G)((a_1 b_2 - a_2 b_1)G + (a_1 - a_2 - b_1 + b_2))^{n-1} \]

We will interpret (A.2) as describing the generating function of certain decorated trees with weighted edges.
A.1. A poset on set partitions associated with a tree. Fix a tree $T \in \mathcal{T}_n^l$ and consider the set of partitions of Nodes($T$) where each block consists solely of nodes that occur on a path that goes right and increases weakly along it. We will refer to this set as $\Pi_T$. This set can be endowed with a partial order by merging blocks whenever the resulting partition satisfies the criterion mentioned above. As is easily verified, the poset thus obtained is isomorphic to a boolean lattice of subsets ordered by inclusion. The unique minimal element is the partition where every block is a singleton set. On the other hand, the unique maximal element in this poset is the coarsest set partition satisfying the criterion mentioned earlier. Suppose the nodes of $T$ are $v_1, \ldots, v_n$ in preorder. We mark all internal nodes $v_i$ that are elements with maximum $i$ in their respective blocks and omit the labels on the tree to obtain a marked tree $\tilde{T}$. Note that $c(\tilde{T})$ uniquely determines the maximal element in $\Pi_T$. Furthermore, we can identify elements in $\Pi_T$ with compositions $\alpha \leq c(\tilde{T})$ and will work with this identification below. Henceforth, by a composition belonging to $\Pi_T$ we mean that, under the earlier identification, the composition maps to a set partition in $\Pi_T$.

Consider the tree $T$ on the left in Figure 25 where the nodes in each connected highlighted region correspond to a block in the maximal element of $\Pi_T$. Thus, the maximal element of $\Pi_T$ corresponds to the composition $(1, 2, 2, 1, 1)$. On the right is a different shading on the same tree where the associated composition is $(1, 2, 1, 2, 1, 1)$.

A.2. Decorated trees. Given $T \in \mathcal{T}_n^l$, a decorated tree is an ordered pair $(T, \beta)$ where $\beta \in \Pi_T$. To solve the noncommutative functional equation in Equation (A.2) we associate a weight $\text{wt}(T, \beta)$ with $(T, \beta)$. Note that, by a procedure outlined earlier, the pair $(T, \beta)$ determines a unique marked tree on $\text{sh}(T)$ which we call $\tilde{T}_\beta$. Note that all left univalent nodes will always be marked. We use the following rules to assign a weight to an internal node $v$ of $\tilde{T}_\beta$.

$$\text{wt}(v) = \begin{cases} 
  a_2b_1 & \text{v a marked bivalent node} \\
  a_1b_2 - a_2b_1 & \text{v an unmarked bivalent node} \\
  b_1 & \text{v a left univalent node} \\
  a_2 & \text{v a marked right univalent node} \\
  a_1 - a_2 - b_1 + b_2 & \text{v an unmarked right univalent node} 
\end{cases}$$

We define $\text{wt}(T, \beta)$ to be the product of weights of the internal nodes. Let $v_1^l, \ldots, v_n^l$ be the labels of the nodes read in preorder. The noncommutative monomial associated with $T$ is $X_T := X_{v_1^l}X_{v_2^l} \cdots X_{v_n^l}$. For instance, we have that $X_T = X_7X_1X_4X_1X_2X_3X_5X_2X_4$ where $T$ is a tree in Figure 25.

Proposition A.1. We have the following expansion for $G$.

$$G = \sum_{n \geq 1} \sum_{T \in \mathcal{T}_n^l} \sum_{\beta \in \Pi_T} \text{wt}(T, \beta) X_T.$$
Proof. For brevity, we set $\gamma = a_1 b_2 - a_2 b_1$ and $\delta = a_1 - a_2 - b_1 + b_2$. Equation (A.2) implies that $G$ is a sum of terms of the form

$$h_n(1 + a_2 G)(1 + b_1 G)(\gamma G + \delta) \cdots (\gamma G + \delta).$$

Picking either 1 or $\delta$ in the product in Equation (A.3) corresponds to a leaf in Figure 26 and the weight gets transferred to the internal node that is the parent of the leaf. The case of choosing weights of the form $a_2 G$, $b_1 G$, and $\gamma G$ is similar. The order in which terms appear in Equation (A.3) ensures that the associated monomials are of the form $X_T$. $\square$

![Figure 26. A pictorial representation for the term in Equation A.3](image)

Henceforth, let $G_n$ denote the summand of $G$ corresponding to a fixed $n$. The sum in Proposition A.1 can be simplified vastly. First, we utilize the “boolean” nature of the poset $\Pi_T$ to perform the first stage of simplification. This comes from endowing a fixed $T \in \mathcal{T}^\ell_n$ with the structure of a decorated tree in all possible ways and summing the associated weights. If $\hat{T}$ denotes the marked tree corresponding to the maximal element in $\Pi_T$, then

$$\sum_{\beta \in \Pi_T} \text{wt}(T, \beta) = (a_2 b_1)^{|\text{Bi}^m(\hat{T})|}(a_1 b_2)^{|\text{Bi}^u(\hat{T})|}a_2^{|\text{Uni}^m(\hat{T})|}b_1^{|\text{Uni}^u(\hat{T})|}(a_1 - b_1 + b_2)^{|\text{Uni}^u(T)|}.$$

The expression on the right-hand side depends only on $\hat{T}$ rather than the decorated trees associated with $T$, and we will denote it by $w'(T)$.

Our second stage of simplifications uses the flip-equivalence defined in Subsection 4.4. The flip-equivalence class of $T$ has a unique right-leaning tree as a representative $\hat{T}$. Furthermore, the flips that take $T$ to $T'$ also take $\hat{T}$ to a marked tree $\hat{T}'$ without altering the marking. Thus, we have

$$\sum_{\hat{T} \sim T} w'(T) = (a_2 b_1)^{|\text{Bi}^m(\hat{T}')|}(a_1 b_2)^{|\text{Bi}^u(\hat{T}')|}(a_2 + b_1)^{|\text{Uni}^m(\hat{T}')|}(a_1 + b_2)^{|\text{Uni}^u(\hat{T}')|}.$$

Since flips preserve preorder reading words, we have $X_{T_1} = X_{T_2}$ for trees $T_1$ and $T_2$ that are flip-equivalent, and we obtain the following expansion for $G_n$.

$$G_n = \sum_{T \in \mathcal{RT}_n^\ell} (a_2 b_1)^{|\text{Bi}^m(T)|}(a_1 b_2)^{|\text{Bi}^u(T)|}(a_2 + b_1)^{|\text{Uni}^m(T)|}(a_1 + b_2)^{|\text{Uni}^u(T)|}X_T.$$

On using the notion of compatibility from Subsection 4.2 and Lemma 2.4, we obtain the following expansion for $G_n$ in the basis of noncommutative ribbon Schur functions.
Theorem A.2.
\[
G_n = \sum_{T \in \mathcal{RT}_n} (a_2b_1)^{|\text{Bi}^m(T)|} (a_1b_2)^{|\text{Bi}^n(T)|} (a_2 + b_1)^{|\text{Uni}^m(T)|} (a_1 + b_2)^{|\text{Uni}^n(T)|} \left( \sum_{\delta \in [\hat{c}(T), \hat{d}(T)]} r_\delta \right).
\]

Appendix B. The values of $B_n$ and $G_n$ for small $n$

For the sake of brevity, the expansions below assume that $\alpha = a_1b_2$, $\beta = a_1 + b_2$, $\gamma = a_2b_1$ and $\delta = a_2 + b_1$.

| $n$  | $B_n(x; a_1, a_2, b_1, b_2)$                                                                 |
|------|-----------------------------------------------------------------------------------------------|
| 1    | $r_1$                                                                                         |
| 2    | $\delta r_{11} + \beta r_2$                                                                  |
| 3    | $(\delta^2 + \gamma)r_{111} + (\beta \delta + \gamma)r_{112} + (\beta \delta + \alpha)r_{21} + (\beta^2 + \alpha)r_3$ |
| 4    | $(\delta^3 + 3\gamma \delta)r_{1111} + (\beta \delta^2 + \beta \gamma + 2\gamma \delta)r_{1112} + (\beta^2 \delta^2 + \beta \gamma + \alpha \delta + \gamma \delta)r_{121} + (\beta^2 \delta^2 + 2\beta \gamma + \alpha \delta + 3\alpha \gamma + \gamma^2)r_{1122} + (\beta \delta^3 + 3\beta \gamma \delta + 3\gamma \delta^2 + 2\gamma^2)^2 r_{11112}$ |
| 5    | $(\delta^4 + 6\gamma \delta^2 + 2\gamma^2)r_{11111} + (\beta \delta^3 + 3\beta \gamma \delta + 3\gamma \delta^2 + 2\gamma^2)^2 r_{11112}$ |

Expansion for $G_n(x; a_1, a_2, b_1, b_2)$ in terms of ribbon Schurs

| $n$  | $G_n(x; a_1, a_2, b_1, b_2)$                                                                 |
|------|-----------------------------------------------------------------------------------------------|
| 1    | $r_1$                                                                                         |
| 2    | $\delta r_{11} + \beta r_2$                                                                  |
| 3    | $(\delta^2 + \gamma)r_{111} + (\beta \delta + \gamma)r_{112} + (\beta \delta + \alpha)r_{21} + (\beta^2 + \alpha)r_3$ |
| 4    | $(\delta^3 + 3\gamma \delta)r_{1111} + (\beta \delta^2 + \beta \gamma + 2\gamma \delta)r_{1112} + (\beta^2 \delta^2 + \beta \gamma + \alpha \delta + \gamma \delta)r_{121} + (\beta^2 \delta^2 + 2\beta \gamma + \alpha \delta + 3\alpha \gamma + \gamma^2)r_{1122} + (\beta \delta^3 + 3\beta \gamma \delta + 3\gamma \delta^2 + 2\gamma^2)^2 r_{11112}$ |
| 5    | $(\delta^4 + 6\gamma \delta^2 + 2\gamma^2)r_{11111} + (\beta \delta^3 + 3\beta \gamma \delta + 3\gamma \delta^2 + 2\gamma^2)^2 r_{11112}$ |

Table of values for $B_n(x; a_1, a_2, b_1, b_2)$

| $n$  | $B_n(x; a_1, a_2, b_1, b_2)$                                                                 |
|------|-----------------------------------------------------------------------------------------------|
| 1    | $r_1$                                                                                         |
| 2    | $\delta r_{11} + \beta r_2$                                                                  |
| 3    | $(\delta^2 + \gamma)r_{111} + (\beta \delta + \gamma)r_{112} + (\beta \delta + \alpha)r_{21} + (\beta^2 + \alpha)r_3$ |
| 4    | $(\delta^3 + 3\gamma \delta)r_{1111} + (\beta \delta^2 + \beta \gamma + 2\gamma \delta)r_{1112} + (\beta^2 \delta^2 + \beta \gamma + \alpha \delta + \gamma \delta)r_{121} + (\beta^2 \delta^2 + 2\beta \gamma + \alpha \delta + 3\alpha \gamma + \gamma^2)r_{1122} + (\beta \delta^3 + 3\beta \gamma \delta + 3\gamma \delta^2 + 2\gamma^2)^2 r_{11112}$ |
| 5    | $(\delta^4 + 6\gamma \delta^2 + 2\gamma^2)r_{11111} + (\beta \delta^3 + 3\beta \gamma \delta + 3\gamma \delta^2 + 2\gamma^2)^2 r_{11112}$ |
References

[1] C. Athanasiadis, Characteristic polynomials of subspace arrangements and finite fields, Adv. Math. 122 (1996), 193–233.

[2] C. Athanasiadis, Deformations of Coxeter hyperplane arrangements and their characteristic polynomials, Adv. Stud. Pure Math. 27 (2000), 1–26.

[3] C. Athanasiadis, Extended Linial hyperplane arrangements for root systems and a conjecture of Postnikov and Stanley, J. Algebraic Combin. 10 (1999), 207–225.

[4] C. Athanasiadis, Gamma-nonnegativity in Combinatorics and Geometry, slides: http://users.uoa.gr/~caath/Strobl.pdf.

[5] O. Bernardi, Deformations of braid arrangements and Trees, arxiv preprint: https://arxiv.org/abs/1604.06554.

[6] S. Blanco and T. K. Petersen, Counting Dyck paths by area and rank, Ann. Comb. 18 (2014), 171–197.

[7] M. Bousquet-Mélou, É. Fusy and L.-F. Préville-Ratelle, The number of intervals in the m-Tamari lattices, Electron. J. Combin. 18 (2011), 26 pp.

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

[9] P. Diaconis and J. Fulman, Foulkes characters, Eulerian idempotents, and an amazing matrix, J. Algebraic Combin. 36 (2012), 425–440.

[10] B. Drake, An inversion theorem for labeled trees and some limits of areas under lattice paths, Phd. Dissertation, Brandeis University (2008).

[11] H. Foukou, Eulerian numbers, Newcomb’s problem and representations of symmetric groups, Discrete Math. 30 (1980), 3–49.
[60] R. Stanley, Enumerative Combinatorics, Vol. 2, Cambridge Studies in Adv. Math., 62, Cambridge Univ. Press, Cambridge, 1999.

[61] R. Stanley, Hyperplane arrangements, http://www-math.mit.edu/~rstan/arrangements/arr.html

[62] R. Stanley, Hyperplane arrangements, interval orders, and trees, Proc. Nat. Acad. Sci. U.S.A. 93 (1996), 2620–2625.

[63] R. Stanley, Parking functions and noncrossing partitions, Electron. J. Combin. 4 (1997), 14 pp.

[64] D. Tamari, The algebra of bracketings and their enumeration, Nieuw Arch. Wisk. 10 (1962), 131–146.

[65] V. Tewari, Gessel polynomials, rooks, and extended Linial arrangements, arxiv preprint: https://arxiv.org/abs/1604.06894

[66] M. L. Wachs, Poset topology: tools and applications, Geometric Combinatorics, IAS/Park City Math. Ser., 13, Amer. Math. Soc., Providence, RI, 2007, pp. 497–615.

Department of Mathematics, Brandeis University, Waltham, MA 02453, USA
E-mail address: gessel@brandeis.edu

Department of Mathematics, University of Washington, Seattle, WA 98195, USA
E-mail address: stgriff@math.washington.edu

Department of Mathematics, University of Washington, Seattle, WA 98195, USA
E-mail address: vasut@math.washington.edu