Non-Contiguous Pattern Containment in Binary Trees *

Lara Pudwell †

Department of Mathematics
and Computer Science
Valparaiso University
Valparaiso, IN 46383, USA

Lara.Pudwell@valpo.edu

Connor Scholten

Department of Mathematics
Grand Valley State University
Allendale, MI, 49401, USA

Tyler Schrock

Department of Mathematics
Troy University
Troy, AL, 36082, USA

Alexa Serrato

Department of Mathematics
Harvey Mudd College
Claremont, CA, 91711, USA

January 28, 2014

Abstract

In this paper we consider the enumeration of binary trees containing non-contiguous binary tree patterns. First, we show that any two ℓ-leaf binary trees are contained in the set of all n-leaf trees the same number of times. We give a functional equation for the multivariate generating function for number of n-leaf trees containing a specified number of copies of any path tree, and we analyze tree patterns with at most 4 leaves. The paper concludes with implications for pattern containment in permutations.

*Research was supported by the National Science Foundation (NSF DMS-0851721).

†Corresponding author

1 Introduction

Pattern avoidance has been studied in a number of combinatorial objects including permutations, words, partitions, and graphs. In this paper we consider such pattern questions in trees. Conceptually, tree T avoids tree t if there is no copy of t anywhere inside of T. Pattern avoidance in vertex-labeled trees has been studied in various contexts by Steyaert and Flajolet [19], Flajolet, Sipala, and Steyaert [10], Flajolet and Sedgewick [9], and Dotsenko [6] while Khoroshkin and Piontkovski [12] considered generating functions for general unlabeled tree patterns in a different setting.

In 2010, Rowland [16] explored contiguous pattern avoidance in binary trees (that is, rooted ordered trees in which each vertex has 0 or 2 children). He chose to work with binary trees because there is natural bijection between n-leaf binary trees and n-vertex trees. In 2012, Gabriel, Peske, Tay, and the first author [11] considered Rowland's definition of tree pattern in ternary, and more generally in m-ary, trees.

The patterns in [11] and [16] may be seen as parallel to consecutive patterns in permutations. In those papers, tree T was said to contain tree t as a (contiguous) pattern if t was a contiguous, rooted, ordered, subtree of T. In 2012, Dairyko, Tyner, Wynn, and the first author [5] considered non-contiguous patterns in binary trees in order to introduce an tree pattern analogue of classical permutation patterns. In particular, they showed that for any $n, \ell \in \mathbb{Z}^+$, any two ℓ-leaf non-contiguous binary tree patterns are avoided by equally many n-leaf trees and gave an explicit generating function for this enumeration.

In this paper, we follow the definition of tree pattern in [5] to mirror the idea of classical pattern avoidance in permutations. However, instead of focusing on trees that do not contain tree pattern t, we turn our attention to the number of trees with exactly k copies of tree pattern t, making pattern avoidance the special case where $k = 0$. Ultimately, we study the total number of copies of a given tree pattern in the set of all n-leaf trees to mirror the work of Bóna in [1, 2] where he considers the total number of copies of a given permutation pattern of length 3 in the set of all 132-avoiding permutations of length n.

All trees in this paper are rooted and ordered. We will focus on full binary trees, that is, trees in which each vertex has 0 or 2 (ordered children). Two children with a common parent are *sibling vertices*. A vertex with no children is a *leaf* and a vertex with 2 children is an *internal vertex*. A binary tree with

n leaves has $n-1$ internal vertices, and the number of such trees is given by the nth Catalan number (OEIS A000108). For simplicity of computation, we adopt the convention that there are zero rooted binary trees with zero leaves. The first few binary trees are shown in Figure 1, with names that will be referred to throughout the paper.

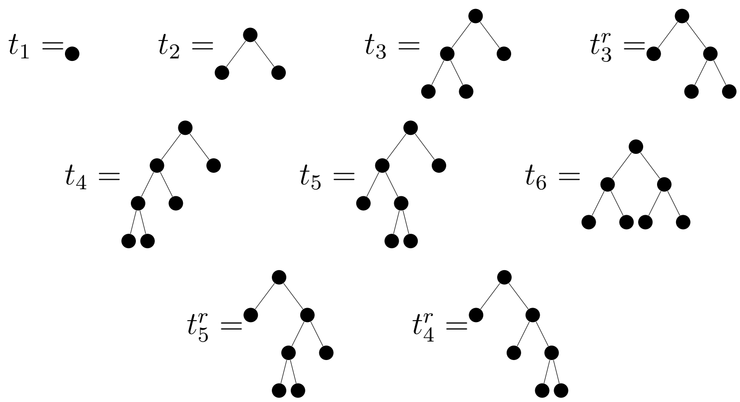

Figure 1: Binary trees with at most 4 leaves

2 Definitions and Notation

Tree T *contains* t as a (non-contiguous) tree pattern if t can be obtained from T via a finite sequence of edge contractions. Conversely, T *avoids* t if there is no sequence of edge contractions that produces t from T. For example, consider the three trees shown in Figure 2. T avoids t_4 as a contiguous pattern, but T contains t_4 non-contiguously (contract both dashed edges). On the other hand, T avoids t_6 both contiguously and non-contiguously since no vertex of T has a left child and a right child, both of which are internal vertices.

The definition of pattern in the previous paragraph is unambiguous for deciding the question "does T contain t?", but becomes more complicated when determining "how many copies of t are in T?" To remove ambiguity, we make the convention that if an edge between a parent vertex and a child vertex is contracted, then the edge from the parent to its other child must be contracted simultaneously.

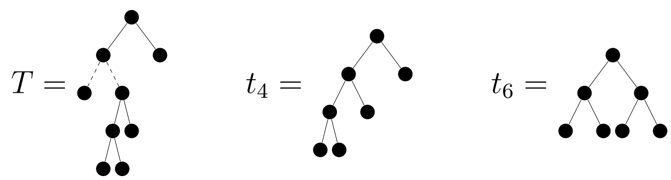

Figure 2: Three binary trees

Define $\mathrm{tr}_t(n,k)$ to be the number of n-leaf binary trees that contain exactly k copies of tree pattern t non-contiguously. For any tree T, let $\mathrm{co}_t(T)$ be the number of copies of t in T. We write $\mathbb{T}_n$ for the set of n-leaf binary trees and $\mathbb{T} = \cup_{n\geq 1}\mathbb{T}_n$. Further, let $\ell(t)$ be the number of leaves of t. We are particularly interested in determining $\mathrm{tr}_t(n,k)$ for various choices of t, n, and k. To this end, we define

$$F_t(x,y) := \sum_{T\in\mathbb{T}} x^{\ell(T)} y^{\mathrm{co}_t(T)} = \sum_{n\geq 1}\sum_{k\geq 0} \mathrm{tr}_t(n,k)x^n y^k.$$

In [5], the authors were concerned with pattern avoidance, so they focused on $\mathrm{tr}_t(n,0)$. They showed the following enumeration.

Theorem 1 (Dairyko, Tyner, Pudwell, Wynn, [5]). *Let $\ell \in \mathbb{Z}^+$ and let t be a binary tree pattern with ℓ leaves. Then*

$$F_t(x,0) = \sum_{n\geq 1} \mathrm{tr}_t(n,0)x^n = \frac{\sum_{i=0}^{\lfloor \frac{\ell-2}{2}\rfloor} (-1)^i \cdot \binom{\ell-(i+2)}{i} \cdot x^{i+1}}{\sum_{i=0}^{\lfloor \frac{\ell-1}{2}\rfloor} (-1)^i \cdot \binom{\ell-(i+1)}{i} \cdot x^{i}}.$$

In particular,

Corollary 2 (Dairyko, Tyner, Pudwell, Wynn, [5]). *Fix $\ell \in \mathbb{Z}^+$. Let t and s be two ℓ-leaf binary tree patterns. Then*

$$F_t(x,0) = F_s(x,0).$$

We obtain a parallel result to Corollary 2 if we focus on

$$\mathrm{toc}_t(n) := \sum_{k=1}^{\infty} k \cdot \mathrm{tr}_t(n,k) = \sum_{T\in\mathbb{T}_n} \mathrm{co}_t(T).$$

We compute $\mathrm{toc}_t(n)$ for any tree t and $n \in \mathbb{Z}^+$ in Section 3. In Section 4 we find a functional equation for $F_t(n,k)$ for any path tree (i.e. any tree avoiding t_6 in Figure 1), and in Section 5 we consider $\mathrm{tr}_t(n,k)$ for any tree pattern t with at most 4 leaves. Finally, in Section 6 we consider implications for pattern containment in permutations.

3 Total number of copies

In this section, we compute $\mathrm{toc}_t(n) = \sum_{k=1}^{\infty} k \cdot \mathrm{tr}_t(n,k) = \sum_{T \in \mathbb{T}_n} \mathrm{co}_t(T)$, i.e. the total number of occurrences of tree pattern t in $\mathbb{T}_n$, for any tree pattern t and any positive integer n. Theorem 3 is parallel to a result of Steyaert and Flajolet [19]. They showed that the total number of occurrences of a (contiguous) ℓ-leaf binary tree pattern in all n-leaf binary trees is independent of the tree pattern and is $\binom{2n-\ell}{n-\ell}$. As it turns out, for non-contiguous tree patterns, we also have the following:

Theorem 3. *Fix $\ell \in \mathbb{Z}^+$. Let t and s be two ℓ-leaf binary tree patterns. Then*

$$\mathrm{toc}_t(n) = \mathrm{toc}_s(n) \textit{ for } n \geq 0.$$

The fact that $\mathrm{toc}_t(n) = \mathrm{toc}_s(n)$ does not guarantee $\mathrm{tr}_t(n,k) = \mathrm{tr}_s(n,k)$ for various choices of k. While Corollary 2 guarantees $\mathrm{tr}_t(n,0) = \mathrm{tr}_s(n,0)$ and Theorem 3 guarantees $\sum_{k=1}^{\infty} k \cdot \mathrm{tr}_t(n,k) = \sum_{k=1}^{\infty} k \cdot \mathrm{tr}_s(n,k)$, it is often the case that $\mathrm{tr}_t(n,k) \neq \mathrm{tr}_s(n,k)$ when $k \geq 1$.

In the following argument we give a bijective proof of Theorem 3. Notice that this is a different approach from the proof of Theorem 1 in [5], which relies on algebraic manipulation of recurrences and generating functions.

Since we are concerned with *pairs* (t,s) of ℓ-leaf binary trees, we make some definitions allowing us to more precisely compare t and s. First, the *intersection* of trees t and s is the largest contiguous rooted tree that is contained in both t and s and includes the root vertex. For example, Figure 3 shows trees t_4 and t_6 along with their intersection.

Two ℓ-leaf trees whose intersection has exactly $\ell - 1$ leaves are called *neighboring trees*. Thus, t_4 and t_6 in Figure 3 are neighboring trees. On the other hand, t_4 and its left-right reflection t_4^r are non-neighboring since their intersection has only 2 leaves. By definition, if t and s are neighboring trees,

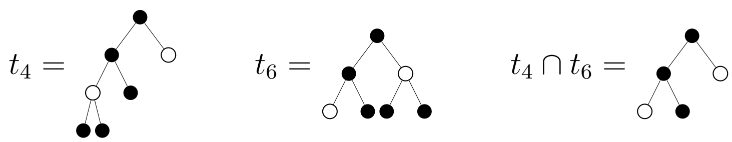

Figure 3: Two neighboring trees and their intersection; white vertices are breaking points

then each of them has exactly two vertices that are not part of the intersection. Call the vertex on each tree that is the parent of the non-intersection vertices the *breaking point*. For example, in Figure 2, the breaking point of t is the left child of the left child of the root. The breaking point of s is the right child of the root. In fact, since both t's breaking point and s's breaking point are part of their intersection, we can identify both breaking points on either of the original trees or on their intersection.

Given neighboring trees t and s, we define a map $\phi_{t,s}$ from the set of copies of t in $\mathbb{T}_n$ to the set of copies of s in $\mathbb{T}_n$. If t appears non-contiguously, we may still identify a (possibly non-contiguous) copy of $t \cap s$ using only edges from that copy of t. The breaking points along this copy of $t \cap s$ are then uniquely determined.

Given a copy c of t in a particular n-leaf tree, find both breaking points on the intersection and swap the subtrees that have the breaking points as their roots. We have now obtained an n-leaf tree with a unique copy of s that has the same intersection with c and the same breaking points. This copy of s is $\hat{c} = \phi_{t,s}(c)$. Figure 4 shows a copy of t_4 being mapped to a copy of t_6 via ϕ_{t_4,t_6}.

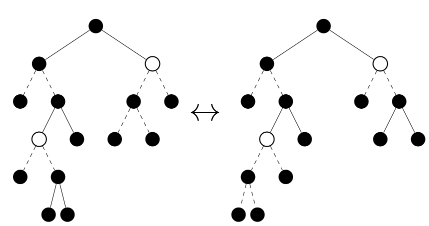

Figure 4: An application of ϕ_{t_4,t_6}

Since t and s are neighboring trees, it is clear that $\phi_{t,s}$ maps a copy of t to a copy of s. Further, $\phi_{t,s}^{-1} = \phi_{s,t}$ since $\phi_{t,s}$ only involves swapping two well-defined subtrees. Thus, $\phi_{t,s}$ is a bijection from the set of copies of t in $\mathbb{T}_n$ to the set of copies of s in $\mathbb{T}_n$.

The fact that $\phi_{t,s}$ is a bijection shows that Theorem 3 is true when t and s are neighboring trees. To show that the theorem holds in general we need the following lemma.

Lemma 4. *Given two ℓ-leaf trees t and s, there is a finite sequence $t = t_1, t_2, \ldots, t_p = s$ of trees such that for any i, ($1 \leq i < p$), t_i and t_{i+1} are neighboring trees.*

Proof. Let t and s be non-neighboring ℓ-leaf binary trees whose intersection is a j-leaf tree. Clearly $j \geq 2$ since both trees share the root vertex and its two children.

To obtain t_{i+1} from t_i, remove a pair of leaves with a common parent from t_i that are not in the intersection of t_i and s, and attach them to a leaf of t_i that is not a leaf of s. The new tree has a larger intersection with s. Repeat until the intersection has $\ell - 1$ leaves. □

Since any two ℓ-leaf tree patterns are a finite sequence of neighboring trees apart, we have that $\phi_{t_{p-1},s} \circ \cdots \circ \phi_{t_2,t_3} \circ \phi_{t,t_2}$ provides a bijection between all copies of t in $\mathbb{T}_n$ and all copies of s in $\mathbb{T}_n$, so Theorem 3 is true.

Theorem 3 also generalizes naturally for copies of m-ary tree patterns within the set of all m-ary trees with n leaves. The definition of intersection and and breaking points remain unchanged, and the swapping action of $\phi_{t,s}$ still applies. To find a sequence of neighboring trees, we need only move collections of m vertices with a common parent instead of pairs, and the rest of the argument goes through as expected.

Now that we know that $\text{toc}_t(n)$ is the same for all ℓ-leaf trees t, we define $\text{toc}_\ell(n) = \text{toc}_t(n)$ where t is an ℓ-leaf tree and compute $\text{toc}_\ell(n)$ in general.

Our first proposition deals with the pathological case of $\ell = 1$.

Proposition 5. $\text{toc}_1(n) = C_{n-1}$ *where C_n is the nth Catalan number.*

Proof. There is exactly one way to contract all edges of a tree to produce the one-leaf tree. Since there is one copy of the one-leaf tree in any given binary tree and there are C_{n-1} binary trees with n leaves, we see that there are C_{n-1} copies of the one-leaf tree in $\mathbb{T}_n$. □

Proposition 6. $\text{toc}_2(n) = (n-1)C_{n-1}$.

Proof. There is only one two-leaf tree and the number of copies of this tree in tree T is equal to the number of internal vertices of T, which is one less than the number of leaves of T. Since there are $n-1$ copies of the two-leaf tree in any n-leaf tree and there are C_{n-1} n-leaf trees, $\text{toc}_2(n) = (n-1)C_{n-1}$. □

More generally, we obtain the following recurrence for $\text{toc}_\ell(n)$:

Proposition 7.

$$\text{toc}_\ell(n) = 2\sum_{k=1}^{n-1} C_{k-1}\text{toc}_\ell(n-k) + \sum_{k=1}^{n-1} C_{k-1}\text{toc}_{\ell-1}(n-k)$$

Proof. Consider ℓ-leaf tree pattern t. A copy of t in T can (a) be fully contained in the left subtree of T's root, (b) be fully contained in the right subtree of T's root, or (c) include T's root.

For the first case, suppose that $\hat{T}$ is an $(n-k)$-leaf tree containing t. $\hat{T}$ appears C_{k-1} times as the left subtree of some n-leaf tree in $\mathbb{T}_n$. Therefore, the number of times t is fully contained in a left subtree of an n-leaf tree in $\mathbb{T}_n$ is $\sum_{k=1}^{n-1} C_{k-1}\text{toc}_\ell(n-k)$. The same sum also counts the number of times t is fully contained in a right subtree of an n-leaf tree in $\mathbb{T}_n$.

If a copy of t includes the root of T, we must count copies of t's left subtree to the left of the root and t's right subtree to the right of the root. By Theorem 3, we may assume that t is the ℓ-leaf right comb, i.e. the unique ℓ-leaf tree where every left child is a leaf. This means that the number of ways for an n-leaf tree to have a copy of t that includes the root is $\sum_{k=1}^{n-1} C_{k-1}\text{toc}_{\ell-1}(n-k)$ where C_{k-1} counts copies of the 1-leaf left subtree addressed in Proposition 5, and $\text{toc}_{\ell-1}(n-k)$ counts copies of the $(\ell-1)$-leaf right comb in the right subtree. □

Fix $\ell \in \mathbb{Z}^+$, and let $t_\ell(x) = \sum_{i=1}^{\infty} \text{toc}_\ell(i)x^i$. Then, using the recurrence of Proposition 7, we have that

$$t_\ell(x) = \frac{t_1(x)t_{\ell-1}(x)}{1-2t_1(x)}.$$

We know from Proposition 5 that $t_1(x) = \dfrac{1-\sqrt{1-4x}}{2}$, the generating function for the Catalan numbers, so by induction, we have

$$t_\ell(x) = \left(-\frac{1}{2}\right)^\ell \frac{\left(-1+\sqrt{1-4x}\right)^\ell}{\left(\sqrt{1-4x}\right)^{\ell-1}}.$$

$n\backslash\ell$	1	2	3	4	5	6	7	8
1	1	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	2	4	1	0	0	0	0	0
4	5	15	7	1	0	0	0	0
5	14	56	37	10	1	0	0	0
6	42	210	176	68	13	1	0	0
7	132	792	794	392	108	16	1	0
8	429	3003	3473	2063	731	157	19	1
9	1430	11440	14893	10254	4395	1220	215	22
10	4862	43758	63004	49024	24465	8249	1886	282

Table 1: $\mathrm{toc}_\ell(n)$ for small n and ℓ

Moreover the following theorem enumerates all copies of a given ℓ-leaf tree in $\mathbb{T}_n$ for any $\ell, n \in \mathbb{Z}^+$.

Theorem 8.

$$\sum_{n\geq 1}\sum_{\ell\geq 1}\mathrm{toc}_\ell(n)x^n y^\ell = \sum_{\ell=1}^{\infty} t_\ell(x)y^\ell = \frac{\sqrt{1-4x}(1-\sqrt{1-4x})y}{(y+2)\sqrt{1-4x}-y}$$

Table 1 gives values of $\mathrm{toc}_\ell(n)$ for $1 \leq n \leq 10$ and $1 \leq \ell \leq 8$. As expected $\mathrm{toc}_\ell(n) = 0$ if $\ell > n$ and $\mathrm{toc}_n(n) = 1$. It also follows that $\mathrm{toc}_{n-a}(n)$ is a polynomial in n of degree a. Further, $\mathrm{toc}_1(n)$ and $\mathrm{toc}_2(n)$ were given above. $\mathrm{toc}_3(n) = 2^{2n-3} - \binom{2n-1}{n} + \binom{2n-3}{n-1}$; this is entry A006419 in the Online Encyclopedia of Integer Sequences [18], which gives several other combinatorial interpretations. $\mathrm{toc}_\ell(n)$ for $\ell \geq 4$ appear to be new sequences to the literature.

4 Pattern containment of path trees

Now that we know $\mathrm{toc}_t(n)$ for any ℓ-leaf tree, we turn our attention to computing $\mathrm{tr}_t(n,k)$ for particular tree patterns. In this section we give a functional equation for

$$F_t(x,y) = \sum_{T\in\mathbb{T}} x^{\ell(T)}y^{\mathrm{co}_t(T)} = \sum_{n\geq 1}\sum_{k\geq 0}\mathrm{tr}_t(n,k)x^n y^k$$

for the case where t is a path tree, that is, t has no vertex which has both left and right grandchildren. Each ℓ-leaf path tree can be encoded uniquely by a word in $\{L, R\}^{\ell-2}$. The two leaf tree is encoded by the empty word. For $\ell > 2$, consider $w = w_1 \cdots w_{\ell-2} \in \{L, R\}^{\ell-2}$. If $w_1 = L$, then w encodes the tree whose root's right child is a leaf, and whose root's left child is the root of the subtree encoded by $w_2 \cdots w_{\ell-2}$. Similarly, if $w_1 = R$, w encodes the tree whose root's left child is a leaf and whose root's right child is the root of the subtree encoded by $w_2 \cdots w_{\ell-2}$. For a path tree t whose encoding is $w_1 \cdots w_\ell$, the deletion $d(t)$ is the tree whose encoding is $w_2 \cdots w_{\ell-2}$. Note that $d^{\ell-2}(t)$ is the 2-leaf tree for any $t \in \mathbb{T}_\ell$. Several iterations of the deletion map on the path tree with word encoding $LRRL$ are shown in Figure 5.

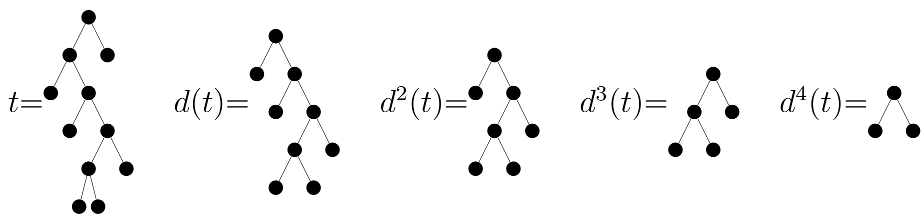

Figure 5: The deletion map for path trees

Theorem 9. *Given ℓ-leaf path tree t, let*

$$G_t(x, y_0, \ldots, y_{\ell-2}) = \sum_{T \in \mathbb{T}} x^{\ell(t)} y_0^{\mathrm{co}_t(T)} y_1^{\mathrm{co}_{d(t)}(T)} y_2^{\mathrm{co}_{d^2(t)}(T)} \cdots y_{\ell-2}^{\mathrm{co}_{d^{\ell-2}(t)}(T)}.$$

Then

$$G_t(x, y_0, \ldots, y_{\ell-2}) = x + y_{\ell-2} G_t(x, y_0, p_1 y_1, \ldots, p_{\ell-2} y_{\ell-2}) G_t(x, y_0, q_1 y_1, \ldots, q_{\ell-2} y_{\ell-2}),$$

where $p_i = y_{i-1}$ and $q_i = 1$ if $d^i(t)$ is the left subtree of $d^{i-1}(t)$ and $p_i = 1$ and $q_i = y_{i-1}$ if $d^i(t)$ is the right subtree of $d^{i-1}(t)$.

Observe that setting $y_1 = y_2 = \cdots = y_{\ell-2} = 1$ causes every catalytic variable to drop out, leaving $F_t(x, y) = G_t(x, y, 1, \ldots, 1)$.

Proof. In this generating function the weight $\mathrm{wt}(T)$ of a given tree T is $\mathrm{wt}(T) = x^{\ell(t)} y_0^{\mathrm{co}_t(T)} y_1^{\mathrm{co}_{d(t)}(T)} y_2^{\mathrm{co}_{d^2(t)}(T)} \cdots y_{\ell-2}^{\mathrm{co}_{d^{\ell-2}(t)}(T)}$. Clearly, for t_1, the one-leaf tree, $\mathrm{wt}(t_1) = x$.

Now, for other trees T we see that each copy of some $d^i(t)$ $(0 \le i \le \ell-2)$ either (a) is contained entirely in the left subtree of T, (b) is contained entirely in the left subtree of T, or (c) includes the root of T. The weight-enumerator for copies of $d^i(t)$ $(0 \le i \le \ell-2)$ covered in cases (a) and (b) is $G_t(x, y_0, y_1, \ldots, y_{\ell-2})G_t(x, y_0, y_1, \ldots, y_{\ell-2})$. If the word representation of $d^i(t)$ begins with L, a copy of $d^i(t)$ including the root consists of the root, the two edges emanating from the root, and a copy of $d^{i+1}(t)$ in the left subtree of t. The p_i contributions keep track of copies of $d^i(t)$ formed in this way. Similarly, the q_i contributions keep track of copies of $d^i(t)$ that include the root of T when $d^i(t)$'s word representation begins with R. The $y_{\ell-2}$ factor keeps track of the copy of the two-leaf tree, t_2, that includes the root of T. □

For example, if t is the tree in Figure 5, we have

$$G_t(x, y_0, y_1, y_2, y_3, y_4) = x + \\ y_4 G_t(x, y_0, y_0y_1, y_2, y_3, y_3y_4) G_t(x, y_0, y_1, y_1y_2, y_2y_3, y_4).$$

For even larger trees, we obtain even more complicated functional equations which are hard to solve in general, but straightforward to extract initial terms from via the computer.

For non-path trees, the interaction of the left and right subtrees of T make this computation more tedious. Analysis of G_t for for small path trees appears in the following section. A parallel argument holds for m-ary path tree containment, although it requires complicated notation for the p_i and q_i terms.

5 Pattern containment of small trees

We have already seen that when t is the one-leaf tree, $\mathrm{tr}_t(n, 1) = C_{n-1}$ and $\mathrm{tr}_t(n, k) = 0$ if $k \neq 1$. Similarly, we know that when t is the two-leaf tree and $n \ge \ell$, then $\mathrm{tr}_t(n, n-1) = C_{n-1}$ and $\mathrm{tr}_t(n, k) = 0$ if $k \neq n-1$. If t^r is the left-right reflection of tree t, then $\mathrm{tr}_{t^r}(n, k) = \mathrm{tr}_{t^r}(n, k)$ for any n and k since if tree T contains k copies of t, then T^r contains k copies of t^r. This means we need only consider one three leaf tree (t_3 in Figure 1) and three different four leaf trees (t_4, t_5, and t_6 in Figure 1) to completely classify tree patterns with at most four leaves.

5.1 Containing a 3-leaf tree

Following the result of Theorem 9, we have that

$$G_{t_3}(x, y_0, y_1) = x + y_1 G_{t_3}(x, y_0, y_0 y_1) G_{t_3}(x, y_0, y_1),$$

and $F_{t_3}(x, y) = G_{t_3}(x, y, 1) = x + x^2 + (y+1)x^3 + (y^3+y^2+2y+1)x^4 + (y^6 + y^5 + 2y^4 + 3y^3 + 3y^2 + 3y + 1)x^5 + (y^{10} + y^9 + 2y^8 + 3y^7 + 5y^6 + 5y^5 + 7y^4 + 7y^3 + 6y^2 + 4y + 1)x^6 + \cdots$.

Several nice sequences quickly appear. In particular:

- As expected, $\text{tr}_{t_3}(n, 0) = 1$.
- $\text{tr}_{t_3}(n, 1) = n - 2$ for $n \geq 2$. (ogf $\frac{x^2}{(1-x)^2}$)
- $\text{tr}_{t_3}(n, 2) = \binom{i-2}{2}$ for $n \geq 3$ (OEIS A000217, ogf $\frac{x^3}{(1-x)^3}$)
- $\text{tr}_{t_3}(n, 3) = \binom{n-3}{1} + \binom{n-3}{2} + \binom{n-3}{3}$ for $n \geq 3$ (OEIS A004006, ogf $\frac{x^3(1-x+x^2)}{(1-x)^4}$)

Each of these formulas can be proved directly by case analysis. In general, $\text{tr}_{t_3}(n, k)$ has a rational ordinary generating function with denominator $(x-1)^{k+1}$, but the numerator has increasingly many terms as k increases.

5.2 Containing a 4-leaf tree

4-leaf trees provide the first opportunity to consider trees with an equal number of leaves that are not reflections of one another. We must consider three different tree patterns for a complete analysis. Two of these three trees fall under the scope of Theorem 9.

For t_4, we have the functional equation

$$G_{t_4}(x, y_0, y_1, y_2) = x + y_2 G_{t_4}(x, y_0, y_0 y_1, y_1 y_2) G_{t_4}(x, y_0, y_1, y_2).$$

Particular sequences:

- As expected, $\text{tr}_{t_4}(n, 0) = 2^{n-2}$ for $n \geq 2$.

- $\mathrm{tr}_{t_4}(n,1) = (n-3)2^{n-4}$ for $n \geq 4$. (OEIS A001787)
- $\mathrm{tr}_{t_4}(n,2) = (n-4)(n-1)2^{n-7}$ for $n \geq 5$ (OEIS A001793)
- $\mathrm{tr}_{t_4}(n,3) = \frac{(n-5)(n-3)(n+2)}{3}2^{n-9}$ for $n \geq 6$ (OEIS A055585)

$\mathrm{tr}_{t_4}(n,k)$ for $k \geq 4$ is new to the OEIS, but each of the sequences above is referenced as the number of 132-avoiding permutations of a given length containing a particular number of copies of the pattern 123. We will see more about this connection to pattern-avoiding permutations in Section 6.

For t_5, we have the functional equation

$$G_{t_5}(x, y_0, y_1, y_2) = x + y_2 G_{t_5}(x, y_0, y_0 y_1, y_2) G_{t_5}(x, y_0, y_1, y_1 y_2).$$

Particular sequences:

- As expected, $\mathrm{tr}_{t_5}(n,0) = 2^{n-2}$ for $n \geq 2$.
- $\mathrm{tr}_{t_5}(n,1) = (n-2)2^{n-5}$ for $n \geq 4$. (OEIS A001792)

$\mathrm{tr}_{t_5}(n,k)$ for $k \geq 2$ is new to the OEIS. $\mathrm{tr}_{t_5}(n,1)$ shows up in a number of combinatorial contexts from compositions to the game of Hex. Also, notice that $\mathrm{tr}_{t_4}(n,k) \neq \mathrm{tr}_{t_5}(n,k)$ when $k > 0$.

t_6 is not a path tree, and thus requires other techniques. If we consider the polynomial $g_{t_6,n}(y) = \sum_{T \in \mathbb{T}_n} y^{\mathrm{co}_{t_6}(T)}$, we obtain the recursion below.

$$g_{t_6,n}(y) = \begin{cases} 1 & n = 1 \\ \sum_{i=1}^{n-1} y^{(i-1)(n-i-1)} g_{t_6,i}(y) g_{t_6,n-i}(y) & \textit{otherwise} \end{cases}$$

Here, i counts the number of leaves to the left of the root, $y^{(i-1)(n-i-1)}$ accounts for copies of t_6 including the root of T, and $g_{t_6,i}(y)$ (resp. $g_{t_6,n-i}(y)$) accounts for copies of t_6 entirely contained in the left (resp. right) subtree of T.

Particular sequences:

- As expected, $\mathrm{tr}_{t_6}(n,0) = 2^{n-2}$ for $n \geq 2$.
- $\mathrm{tr}_{t_6}(n,1) = 2^{n-4}$ for $n \geq 4$. (ogf $\frac{x^3}{1-2x}$)
- $\mathrm{tr}_{t_6}(n,2) = 2^{n-3}$ for $n \geq 5$. (ogf $\frac{4x^4}{1-2x}$)

- $\mathrm{tr}_{t_6}(n,3) = 2^{n-3}$ for $n \geq 6$. (ogf $\frac{8x^5}{1-2x}$)
- $\mathrm{tr}_{t_6}(n,4)$ has ogf $\frac{16x^6+6x^4}{1-2x}$

In fact, it is clear that for any fixed k and sufficiently large n, $\frac{\mathrm{tr}_{t_6}(n,k)}{\mathrm{tr}_{t_6}(n-1,k)} = 2$. This is because there are a finite number of ways to arrange exactly k copies of t_6 before the only option is to take an $(n-1)$-leaf tree with k copies of t_6 and make it to be either the left subtree or the right subtree of a new n-leaf tree. The numerators of the ordinary generating functions for $\mathrm{tr}_{t_6}(n,k)$ for fixed k have increasingly many terms as k grows larger.

Larger non-path trees introduce additional difficulties. Counting copies of the left (resp. right) subtree of t_6 is equivalent to counting single vertices. Counting copies of a non-path tree that includes the root is more complicated when either subtree is larger.

We end this section with a conjecture. Further computational data suggests this is the case, but settling this question in general remains an open problem.

Conjecture 10. $\mathrm{tr}_t(n,k) = \mathrm{tr}_s(n,k)$ *for all* $k \geq 0$ *and* $n \geq 1$ *if and only if* $s = t^r$.

6 Connections to pattern-avoiding permutations

Several sequences obtained by counting trees that contain non-contiguous binary tree patterns are already known in the literature for pattern-containing permutations. In this section we make the relationship between trees and permutations explicit.

To this end, let $\mathcal{S}_n$ denote the set of permutations of length n. As in the introduction, given $\pi \in \mathcal{S}_n$ and $\rho \in \mathcal{S}_k$ we say that π contains ρ as a pattern if there exist indices $1 \leq i_1 < \cdots < i_k \leq n$ such that $\pi_{i_a} < \pi_{i_b}$ if and only if $\rho_a < \rho_b$. Let $\mathcal{S}_n(Q) = \{\pi \in \mathcal{S}_n \mid \forall \rho \in Q, \pi \text{ avoids } \rho\}$, and $\mathrm{s}_n(Q) = |\mathcal{S}_n(Q)|$. For example, $\mathrm{s}_n(\{12\}) = 1$ for $n \geq 1$ since the only way to avoid the pattern 12 is to be the decreasing permutation of length n. It is also well-known that if $\rho \in \mathcal{S}_3$, then $\mathrm{s}_n(\{\rho\}) = C_n$ where C_n is the nth Catalan number.

The following theorem provides an initial relationship between pattern-avoiding trees and pattern-avoiding permutations that we seek to expand.

Theorem 11 (Dairyko, Tyner, Pudwell, Wynn, [5])**.** *Let t be any binary tree pattern with $k \geq 2$ leaves. Then*

$$\mathrm{tr}_t(n, 0) = \mathrm{s}_{n-1}(\{132, 12 \cdots (k-2)(k-1)\}).$$

In fact, a stronger statement is true. It is well known that the set of binary trees with n leaves is in bijection with the set of permutations of length $n-1$ which avoid the pattern 132.

To see this, label the root of tree t with the label $n-1$. Now, suppose there are i internal vertices to the right of the root and $(n-i-2)$ internal vertices to the left of the root. The i vertices on the right will receive labels from the set $\{1, \ldots, i\}$ and the vertices on the left will receive labels from the set $\{i+1, \ldots n-2\}$. For each subtree, give the root the largest available label and continue recursively until each internal vertex has been labeled.

Now, there is a natural left-to-right ordering of the vertices of t; in particular for each vertex v, all vertices in v's left subtree are to the left of v and all vertices in v's right subtree are to the right of v. Read the labels of the vertices from left to right to obtain a permutation $\pi \in \mathcal{S}_{n-1}$. Necessarily, π avoids 132 because all labels to the left of a given vertex have larger labels than all labels to the right.

This correspondence between 132-avoiding permutations and binary trees is not new. If one ignores the leaves in our trees, the bijection given above is a symmetry of the correspondence between postorder-labeled trees with inorder-read permutations found in [8]. Further work connecting permutations to binary trees in the context of sorting can be found in [3], [7], [13], [14], [15], and [17].

To make this result even stronger we turn to *mesh patterns*. Mesh patterns were introduced by Brändén and Claesson [4] in a search for more compact expressions for various permutation statistics. They were later generalized by Úlfarsson [20] to unify the results for permutation patterns used in characterizing Schubert varieties and in analyzing stack-sortability.

The graph of permutation $\pi = \pi_1 \cdots \pi_n$ is obtained by plotting the points $\{(i, \pi_i) \mid 1 \leq i \leq n\}$ in the Cartesian plane. If π contains $\rho \in \mathcal{S}_m$ as a classical pattern, as defined above, then the graph of π has m rows and m columns whose points appear in the same arrangement as the points in the graph of ρ.

The graph of a mesh pattern is the graph of a classical permutation with some squares in the graph shaded. For example, the graph of 132 and the

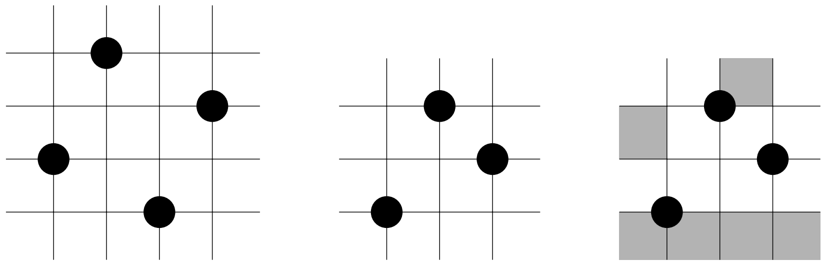

Figure 6: 2413, 132, and a mesh pattern

graph of a mesh pattern with underlying permutation pattern 132 are shown in Figure 6. A copy of a mesh pattern is a copy of the underlying classical pattern but where no points appear in the shaded regions. A permutation is said to avoid a mesh pattern if it contains no copies of the mesh pattern. For example, the permutation 2413, whose graph is also shown in Figure 6 contains 132 as evidenced by the subsequence 243. However 2413 avoids the mesh pattern shown because there is no copy of 132 where all gray regions are empty; in particular, the only copy of 132 is given by the subsequence 243, but the point for the digit 1 appears in the gray strip at the bottom of the mesh pattern.

The bijection above between binary trees and 132-avoiding permutations associates each tree to a classical permutation pattern in a natural way. However sometimes the pattern corresponding to a particular tree may embed in a larger permutation without the tree pattern being embedded in the corresponding larger tree. For example, the permutations 3241, 3421, and 321 and their corresponding trees are shown in Figure 7. Notice that while 3241 contains a copy of 321, the corresponding tree does not contain a copy of the 4-leaf right comb. Also, while 3421 contains precisely 2 copies of the permutation pattern 321, the corresponding tree only contains one copy of the 4-leaf right comb. We repair this discrepancy by associating trees with mesh patterns.

The discrepancy between tree patterns and permutation patterns occurs precisely when pattern ρ has a descent, i.e. a pair of adjacent elements such that $\rho_i > \rho_{i+1}$. A descent in a permutation pattern can be embedded in a tree either as one vertex being the right child (or right descendant) of another vertex or as one vertex being in the left subtree and the other in the right subtree of a third vertex. For example, in Figure 7, when 3241 contains the

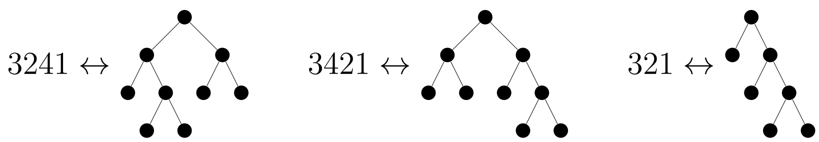

Figure 7: The permutations 3241, 3421, and 321 with their corresponding trees

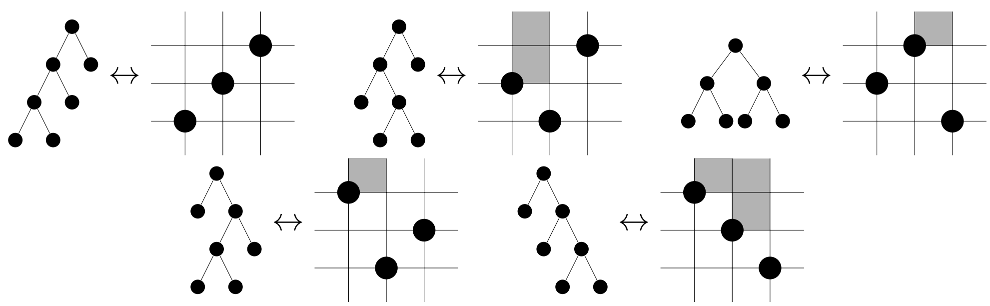

Figure 8: 4-leaf tree patterns and their corresponding mesh patterns

permutation pattern 321, the descent 32 embeds with the 2 vertex as right child of the 3 vertex, while the descent 21 embeds in two separate subtrees of the 4 vertex.

To prevent the split of a descent between two subtrees, we associate each tree pattern with a mesh permutation pattern in the following way:

1. Given tree pattern t, compute π^t, the permutation given by the vertex-labeling bijection above.
2. Construct the permutation graph of π^t.
3. For each descent in π^t, shade all squares between and above the two points involved in the descent. Call the resulting mesh pattern $\hat{\pi^t}$.

Now, copies of $\hat{\pi^t}$ in permutation π^T correspond precisely to copies of tree t in T since the possibility of splitting t between left and right subtrees, without using the root, is removed. Figure 8 shows this correspondence for the 4-leaf tree patterns and Figure 9 shows the correspondence for an even larger tree pattern.

Now, using this map from tree patterns t to mesh patterns $\hat{\pi^t}$ we obtain the following stronger version of Theorem 11.

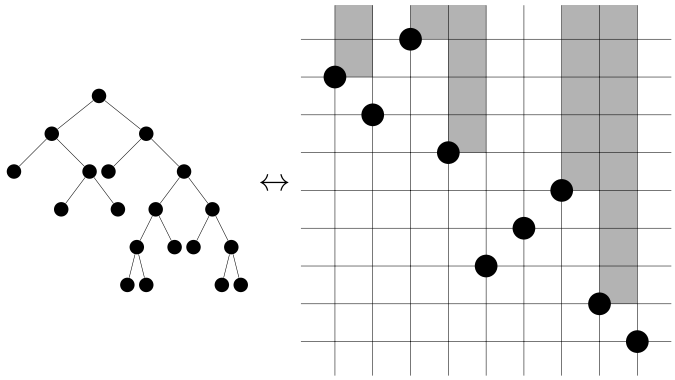

Figure 9: A 9-leaf tree pattern and its corresponding mesh pattern

Theorem 12. *Let t be any binary tree pattern with $\ell \geq 2$ leaves. Then*

$$\mathrm{tr}_t(n,0) = \mathrm{s}_{n-1}(\{132, \hat{\pi^t}\}) = \mathrm{s}_{n-1}(\{132, 12\cdots(\ell-2)(\ell-1)\}).$$

In particular, this restatement gives a set of C_ℓ Wilf-equivalent pattern sets of the form $\{132, \hat{\pi^t}\}$ for any integer ℓ, and furthermore, since the increasing pattern $12\cdots(\ell-1)$, corresponding to the ℓ-leaf left comb has no descents, each of these is pattern pairs equivalent to the classical pattern pair $\{132, 12\cdots(\ell-1)\}$.

We also obtain a stronger statement for pattern *containment* once we augment our current notation for permutation patterns. Because of the bijection between trees and 132-avoiding permutations, we are concerned with permutations in $\mathcal{S}_n(\{132\})$. Now, let

$$a_{n,k}(q) = \{\pi \in \mathcal{S}_n(\{132\}) \mid \pi \text{ contains exactly } k \text{ copies of pattern } q\}.$$

We saw above that $a_{n-1,0}(\hat{\pi^t}) = \mathrm{tr}_t(n,0)$. In fact, the correspondence given above yields the following result of which Theorem 11 is a special case.

Theorem 13. *Let t be any binary tree pattern with $\ell \geq 2$ leaves. Then* $\mathrm{tr}_t(n,k) = a_{n-1,k}(\hat{\pi^t})$.

Because of this correspondence $\mathrm{tr}_{t_3}(n,k)$ counts 132-avoiding permutations with k copies of 12, $\mathrm{tr}_{t_4}(n,k)$ counts 132-avoiding permutations with k copies of 123, and so on.

Further, Theorem 3 causes us to revisit the question of the total number of copies of a given pattern within the set of all length n permutations. To this end, let $A_n(q)$ be the number of copies of pattern q in $\mathcal{S}_n(\{132\})$.

Bóna [1, 2] shows that $A_n(213) = A_n(231) = A_n(312)$ and for sufficiently large n, $A_n(123) < A_n(213) = A_n(231) = A_n(312) < A_n(321)$. In the context of tree patterns, we obtain the following corollary to Theorem 3

Corollary 14. *Given an integer $\ell \in \mathbb{Z}^+$, there exist C_ℓ mesh patterns $\hat{\pi^t}$ for which $A_n(\hat{\pi^t}) = A_n(12\cdots(\ell-1)) = \text{toc}_t(n)$.*

This corollary provides a hidden symmetry to Bóna's result in that there is a mesh pattern $\hat{\pi^t}$ associated to each 4-leaf tree t for which $A_n(\hat{\pi^t}) = A_n(123) = \text{toc}_t(n)$.

References

[1] M. Bóna. The absence of a pattern and the occurrences of another. *Discrete Math. Theor. Comput. Sci.* 12(2):89–102, 2010.

[2] M. Bóna. Surprising symmetries in objects counted by Catalan numbers. *Electron. J. Combin.*, 12(1), P62, 2012.

[3] M. Bousquet-Mèlou. Sorted and/or Sortable Permutations. *Discrete Math.*, 223:23–30, 2000.

[4] P. Brändén, A. Claesson. Mesh patterns and the expansion of permutation statistics as sums of permutation patterns. *Electron. J. Combin.*, 18(2), P5, 2011.

[5] M. Dairyko, L. Pudwell, S. Tyner, and C. Wynn. Non-contiguous pattern avoidance in binary trees. *Electron. J. Combin.* 19(3), P22, 2012.

[6] V. Dotsenko. Pattern avoidance in labeled trees. *Sém. Lothar. Combin.*, B67b: 27 pages, 2012.

[7] S. Even. *Graph Algorithms.* Computer Science Press, 1979.

[8] T. Feil, K. Hutson, R.M. Kretchmar. Tree Traversals and Permutations. *Congr. Numer.*, 172:201–221, 2005.

[9] P. Flajolet and R. Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[10] P. Flajolet, P. Sipala, and J. M. Steyaert. Analytic variations on the common subexpression problem. *Automata, Languages, and Programming: Proc. of ICALP 1990*. volume 443 of *Lecture Notes in Computer Science*, pages 220–234. Springer, 1990.

[11] N. Gabriel, K. Peske, L. Pudwell, and S. Tay. Pattern Avoidance in Ternary Trees. *J. Integer Seq.* 15:12.1.5, 2012.

[12] A. Khoroshkin and D. Piontkovski. On generating series of finitely presented operads. preprint, arXiv:1202.5170.

[13] D.E. Knuth. The art of computer programming. Volume 3. Sorting and Searching. Addison- Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1973.

[14] D. Rotem. Stack Sortable Permutations. *Discrete Math.*, 33:185–196, 1981.

[15] D. Rotem and Y. Varol. Generating Binary Trees from Ballot Sequences. *J. ACM*, 25:396–404, 1978.

[16] E. S. Rowland. Pattern avoidance in binary trees. *J. Combin. Theory, Ser. A*, 117:741–758, 2010. 741–758.

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

[18] N. Sloane. The Encyclopedia of Integer Sequences. Available at http://oeis.org, 2013.

[19] J. M. Steyaert and P. Flajolet. Patterns and pattern-matching in trees: an analysis. *Info. Control*, 58:19–58, 1983.

[20] H. Úlfarsson. A unification of permutation patterns related to Schubert varieties. Pure Math. Appl. (PU.M.A.). 22(2): 273–296, 2011.