Hilbert series in the category of trees with contractions

Eric Ramos
Department of Mathematics, University of Oregon, Eugene, OR 97403

We consider Hilbert series associated to modules over various categories of trees. Using the technology of Sam and Snowden [SS17], we show that these Hilbert series must be algebraic. We then apply these technical theorems to prove facts about certain natural generating functions associated to trees.

1 Introduction

1.1 The setup

Let $\mathcal{C}$ denote an essentially small category and k a field. Then a representation of $\mathcal{C}$ is a functor from $\mathcal{C}$ to the category of k vector spaces. In their seminal work [SS17], Sam and Snowden established the study of representations of combinatorial categories; categories such as FI, of finite sets and injections. Their framework got at the combinatorial heart of the concurrent development of representation stability, due to Church, Farb, and Ellenberg [CEF15][CF13], while also expanding it in a variety of directions.

The language of Sam and Snowden, very broadly speaking, is useful for proving facts about a category's representations in two related, but distinct, realms. The first of these is related to the presence, or lack thereof, of a *Noetherian property*. Just as with modules over rings, one can make sense of *finite generation* when discussing representations of categories (see Definition 2.2) The Noetherian property asserts that submodules of finitely generated modules are also finitely generated. This is the theoretical backbone of virtually all of representation stability theory, as it allows one to prove finite generation statements about representations appearing in the limits of spectral sequences. The second tool granted by Sam and Snowden's work is a means by which one can understand *Hilbert series* of finitely generated representations of one's category.

To explain what is meant by this, let $M : \mathcal{C} \to \mathrm{Vec}_k$ be a finitely generated $\mathcal{C}$ representation, and assume that you have a function ν from the isomorphism classes of objects of $\mathcal{C}$ (which is guaranteed to be a set by our essential smallness assumption) to $\mathbb{N}$. For example, in the case of FI, one may take ν to be the function which maps each set to its cardinality. Such a function is called a *norm* of the category. Then the *Hilbert series* of M with respect to ν is the formal power series

$$H_{M,\nu}(t) := \sum_x \dim_k(M(x)) t^{\nu(x)},$$

where the sum is over isomorphism classes of objects.

Work of Miyata, Proudfoot, and the author applied Sam and Snowden's theory to a variety of categories of graphs [PRb][PRa][MPR]. In all cases, a Noetherian property was proved for

representations of the categories being considered. This was then applied to prove non-trivial consequences about homology groups of graph configuration spaces, as well as Kazhdan-Lusztig coefficients of graphical matroids [EPW16]. Missing from this prior work, however, is a treatment of the Hilbert series of these representations. The goal of the present work is to bridge this gap in the literature, primarily for the category of planar rooted trees with contractions.

1.2 The main theorem

In this work, a *tree* is a nonempty, at most 1-dimensional, connected, and finite CW-complex that is contractible. A *contraction* is a continuous map between trees that involves contracting one or more edges of the tree while also possibly permuting the vertices (see Definition 2.1). A *planar rooted tree* is a tree with a designated vertex (the *root*) along with total orderings on the sets of edges coming out (e.g. away from the root) of every vertex. There is also a notion of planar contractions (see Definition 2.1), which are contractions that preserve all of various structures of the planar tree. The category of planar rooted trees and planar contractions is denoted $\mathcal{PT}$. We consider representations of the opposite category $\mathcal{PT}^{op}$. These representations were the focal point of the precursor work [PRb].

Before we can discuss the Hilbert series of these representations, we first must decide on a norm. In this paper, we will be working with the norm $\nu(T) = |E_T|$, where E_T is the edge set of T. Therefore, for a finitely generated $\mathcal{PT}^{op}$-module M, one would like to consider

$$H_M(t) = \sum_T \dim_k(M(T))t^{|E_T|}.$$

Before we do this, however, we first take the time to decide upon a "nice" enumeration of the isormophism classes of objects in $\mathcal{PT}^{op}$.

Recall the formal language of *Dyck paths*. That is, the language whose alphabet is the set $\{u, d\}$, made up of words of even length, such that each of the characters u and d occupy exactly half of the word, and up to any i, the sub word of letters up to index i has no more d's than u's. It is a fact that Dyck paths, which happen to be counted by the famous Catalan numbers, are in bijection with planar rooted trees. Given a Dyck path w, we read the word from left to right, letter by letter. Each time a u is read, we take a step upward in the left-most (thusfar untraveled) direction, while every time a d is read we step downward. Therefore, The word ud corresponds to a single edge, while the word $uududd$ corresponds to the planar rooted tree that looks like the letter Y. For a Dyck path w, we will write $T_p(w)$ for the planar tree uniquely associated to w. We therefore write

$$H_M(t) = \sum_w \dim_k(M(T_p(w)))t^{l(w)/2},$$

where $l(w)$ is the length of the path w.

Theorem 1. Let M be a finitely generated $\mathcal{PT}^{op}$-module. Then the Hilbert series

$$H_M(t) = \sum_w \dim_k(M(T(w)))t^{l(w)/2},$$

is algebraic.

The proof philosophy we apply for Theorem 1 follows the lingual category approach of Sam and Snowden [SS17]. In particular, we show that the category of planar rooted trees and contractions is unambiguous and context-free. In the final section of this work, we apply Theorem 1 to prove certain natural generating functions associated to Dyck paths are algebraic. These applications are novel (to the best knowledge of the author), and should be of some independent interest.

Because planar trees are just trees with extra structure, to every Dyck path w we can associate a tree (not planar or rooted), which we call $T(w)$. Note that this association does not uniquely recover the Dyck path, but every tree arises in this way. Writing $\mathcal{T}$ for the category of trees and contractions, and given a finitely generated $\mathcal{T}^{op}$-module M, we define its *Hilbert-Dyck* series as the formal power series

$$HD_M(t) := \sum_w \dim_k(M(T(w)))t^{l(w)/2},$$

where the sum is over all Dyck paths, and $l(w)$ is the length of the path. By how the association $w \mapsto T(w)$ is defined, we observe that $l(w)/2 = |E_T|$. Moreover, because this association is not a bijection, the Hilbert-Dyck series is **not** equal to the usual Hilbert series of modules over this category. Indeed, one may write

$$HD_M(t) = \sum_T p_T \dim_k(M(T))t^{|E_T|},$$

where p_T is the total number of Dyck paths which correspond to the tree T. Unlike the aforementioned Hilbert series of M, the Hilbert-Dyck series will prove to be much more tractable. For instance, assuming M is the module which assigns k to every tree, one has

$$HD_M(t) = \sum_T p_T t^{|E_T|} = \sum_n c_n t^n$$

where c_n is the n-th Catalan Number. This generating function is far better understood than the generating function for the number of isomorphism classes of trees. For instance, it is a celebrated fact that this generating function is algebraic. Our second technical result is that this is the case in general.

Theorem 2. Let M be a finitely generated $\mathcal{T}^{op}$-module. Then the Hilbert-Dyck series

$$HD_M(t) = \sum_w \dim_k(M(T(w)))t^{l(w)/2},$$

is algebraic.

One nice property of algebraic generating functions is their asymptotics are fairly predictable. For instance, one has the following fact.

Fact 1.1 (Theorem 3, [BD15])**.** Let $(f_n)_{n\geq 0}$ be a sequence of natural numbers such that $F(t) := \sum_{n\geq 0} f_n t^n$ is an algebraic function. Further assume that $F(t)$ has a unique singularity at its radius of convergence. Then there exist constants $C, \rho \in \mathbb{R}, \alpha \in \mathbb{Q}$, such that f_n is asymptotically close to $Cn^\alpha \rho^n$. That is to say,

$$\lim_{n\to\infty} \frac{f_n}{Cn^\alpha \rho^n} = 1.$$

Remark 1.2. The requirement that the generating function has a unique singularity on its radius of convergence is not strictly necessary, though the statement is more complicated if we do not assert it. In this more technical case, the ultimate conclusion is only true up to residue classes of n (see [BD15, Theorem 3]).

This fact can be observed explicitly in the case where $f_n = c_n$ is n-th Catalan number. In this case we have, from Stirling's approximation,

$$f_n \cong \frac{1}{\sqrt{\pi}} n^{-3/2} 4^n.$$

Coming back to the context of Theorem 1, we see that if you look at the total dimension

$$\sum_{|E_T|=n} \dim_k M(T)$$

then (possibly up to the residue class of n), it grows at worst like some power of n times an exponential. That is, it grows at worst exponentially. This is consistent with the fact, proven in [PRb], that each individual vector space $M(T)$ is bounded by a polynomial in the number of edges of T, whenever T is sufficiently large. Unfortunately the techniques of the current paper do not immediately recover the constants C, α, and ρ. It would be interesting to see whether this can be done in general, and how they compare to the analogous constants for the generating function of the Catalan numbers.

1.3 Other categories of graphs

The current work mainly considers the category of trees with contractions. However, Because edge contractions are homotopy equivalences, they preserve the first Betti-number, or *genus*, of the graph. This shows that the category of all graphs and contractions is stratified by this genus invariant, where the tree case is only one stratum. The work [PRa] shows that other strata of the category of all graphs and edge contractions are also of great interest.

The techniques of this paper will generalize to these other strata, although statements of theorems become considerably more difficult. Indeed, by looking at spanning trees, one may think of a higher genus graph as being a tree decorated with the data of how the extra edges are attached to each vertex. This is exemplified in the category $\mathcal{PG}_g$ discussed in [PRa]. Instead of working with

Dyck paths, one instead must work with Dyck paths that are decorated with this finite amount of extra data. In particular, one may define generalized Hilbert-Dyck series and prove they are algebraic. To the author's knowledge, these decorated Dyck paths have not appeared in the literature, and it is therefore unclear whether they are of any particular interest. For this reason, we do not pursue this direction further.

That being said, however, it is certainly possible that these more general categories of graphs have differently defined Hilbert series that admit nice formulas. We leave this as an avenue for possible future research. There is particular interest in understanding Hilbert series of the Graph minor category, as described in [MPR].

Acknowledgements

The author was supported by NSF grant DMS-1704811. He would like to send thanks to Ben Young for various conversations that were useful during the creation of this work. He would also like to send thanks to Nick Proudfoot, whose editorial suggestions vastly improved the quality of the writing. Finally, the author would like to send thanks to the anonymous referee, whose suggestions greatly helped the clarity of certain arguments.

2 Background

2.1 Categories of trees

In this section, we outline the three main categories whose representations will be studied in this work. Most of what follows can be found in [PRb], and [Bar].

Definition 2.1. A **tree** is a one-dimensional contractible CW-complex. A **rooted tree** is a tree paired with a choice of vertex called the **root**. This choice of root implicitly directs the edges of the tree away from the root. A **planar rooted tree**, or just a **planar tree**, is a rooted tree equipped with well-orderings on the sets of edges leaving each vertex. Planar trees have a natural well-ordering on their vertices via a depth-first search from the root.

Given trees T, T', a **contraction** from T to T' is a map of sets

$$\varphi : V_T \sqcup E_T \to V_{T'} \sqcup E_{T'}$$

satisfying:

- $\varphi(V_T) = V_{T'}$;
- for every $e' \in E_{T'}$ there exists a unique edge $e \in E_T$ with $\varphi(e) = e'$;
- for every $e = \{x, y\} \in E_T$, if $\varphi(e) = v' \in V_{T'}$ then $\varphi(x) = \varphi(y) = v'$, while if $\varphi(e) = e' \in E_{T'}$ then $e' = \{\varphi(x), \varphi(y)\}$;

- for every $v' \in V_{T'}$, the preimage $\varphi^{-1}(v') \subseteq V_T \sqcup E_T$ consists of the edges and vertices of some subtree of T.

Given two rooted trees, a **rooted contraction** between them is a contraction of the underlying trees which preserves the root. Finally, a **planar contraction** between planar trees $\varphi : T \to T'$ is a rooted contraction with the property that given two vertices $v_1', v_2' \in V_T$ such that $v_1' < v_2'$ in the depth-first order, one has that the vertex in $\varphi^{-1}(v_1')$ closest to the root is smaller than the vertex in $\varphi(v_2')$ closest to the root, in the depth-first order.

Finally, we will write $\mathcal{T}$ for the category of trees with contractions, $\mathcal{RT}$ for the category of rooted trees with rooted contractions, and $\mathcal{PT}$ for the category of planar trees with planar contractions.

In this paper, we will be largely concerned with the representation theory of the categories $\mathcal{T}$, $\mathcal{RT}$, and $\mathcal{PT}$. The study of such objects was essentially initiated by Barter [Bar], although a different language was used in that work. In the precursers to the current paper [PRb, PRa], Proudfoot and the author prove that the categories presented above are equivalent to those considered by Barter.

Definition 2.2. Let $\mathcal{C}$ denote anyone of the categories $\mathcal{T}$, $\mathcal{RT}$, or $\mathcal{PT}$ and let k be a field. Then a **representation of $\mathcal{C}^{op}$** or a **$\mathcal{C}^{op}$-module** is a contravariant functor

$$M : \mathcal{C} \to \mathrm{Vec}_k$$

where Vec_k is the category of finite dimensional vector spaces over k. Equivalently, a $\mathcal{C}^{op}$-module is a functor

$$M : \mathcal{C}^{op} \to \mathrm{Vec}_k\,.$$

We say that a $\mathcal{C}^{op}$-module M is **finitely generated** if there exists a finite list of trees (or rooted trees, or planar trees) $\{T_i\}_{i\in I}$ such that for any tree $T \notin \{T_i\}_{i\in I}$, the vector space $M(T)$ is spanned by the images

$$M(\varphi) : M(T_i) \to M(T),$$

where $\varphi : T \to T_i$ is a contraction. We call the trees $\{T_i\}$ the **generators** of the module M, and say $\{T_i\}$ **generates** M.

Remark 2.3. The category of $\mathcal{C}^{op}$-modules is abelian, with the standard abelian operations defined point-wise. In particular, we can reuse terms from the language of modules over a ring without ambiguity.

As the field k will not affect the proofs of statements of results, we now fix a field k for the remainder of the paper.

One of the most important properties of finitely generated $\mathcal{C}^{op}$-modules is the Noetherian property.

Theorem 2.4 ([Bar], [PRb]). *If M is a finitely generated $\mathcal{C}^{op}$-module, then all submodules of M are also finitely generated.*

In this paper we consider the types of growth that can appear in the dimensions of the vector spaces $M(T)$. This question was partially considered in the precursor work [PRb], where the following is proved.

Theorem 2.5 ([PRb]). *Let M be a finitely generated $\mathcal{C}^{op}$-module. Then there exists a polynomial $P_M(t) \in \mathbb{Q}[t]$ such that for all trees with $|E_T| \gg 0$, one has*

$$\dim_k(M(T)) \leq P_M(|E_T|).$$

[PRb] also proves results which show how this polynomial behavior is sharp, so long as you vary the trees within certain natural families of trees. In this work we consider growth as it pertains to the module M as a whole, instead of how it pertains to the individual vector spaces which comprise it.

Definition 2.6. It is a well known fact that planar rooted trees with n edges are in bijection with **Dyck paths of length** $2n$. A Dyck path is a word of even length $2n$ in the alphabet $\{u, d\}$ such that each of the characters u and d appear exactly n times and up to any i, the sub word of letters up to index i has no more d's than u's.

Given a Dyck path w of length $2n$, we write $T(w)$ (resp. $T_r(w)$, resp. $T_p(w)$) to denote the tree (resp. rooted tree, resp. planar rooted tree) associated to w. Note that $T(w)$ and $T_r(w)$ do not uniquely determine the original word w, though every tree and rooted tree can be written in this form for some w.

Let M denote a finitely generated $\mathcal{T}^{op}$-module. Then the Hilbert-Dyck series associated to M is the formal power series

$$HD_M(t) = \sum_w \dim_k(M(T(w)))t^{|E(T(w))|},$$

where the sum is over all Dyck paths. Note that $|E(T(w))| = l(w)/2$, where $l(w)$ is the length of the word w. We similarly define Hilbert-Dyck series for modules over the category $\mathcal{RT}^{op}$.

Example 2.7. Consider the $\mathcal{T}^{op}$-module which assigns to every tree the vector space k, and to every contraction the identity map. This is sometimes referred to as the trivial $\mathcal{T}^{op}$-module. Then we have

$$HD_M(t) = \sum_{n \geq 1} c_n t^n,$$

where c_n is the number of Dyck paths of length n, i.e. the n-th Catalan number. In particular, $HD_M(t)$ is precisely the generating function for the Catalan numbers.

Note that, if instead M was the $\mathcal{RT}^{op}$-module (resp. $\mathcal{PT}^{op}$-module) which assigns k to every rooted (resp. planar rooted) tree, then $HD_M(t)$ (resp. $H_M(t)$) is identical to the above.

It is a well-known fact that the generating function for the Catalan numbers is **algebraic**. That is, it satisfies a polynomial equation with coefficients in $\mathbb{Q}(n)$. Our main result can therefore be seen as a categorification of this fact. See [BM05] for a comprehensive treatment of algebraic generating functions and their applications.

2.2 PDA's and context-free languages

In this section we discuss the theory of Push-down Automata (PDA) and their associated context-free languages. See [ABB97] for a standard reference. Before we dive into the somewhat intimidating formalities of the subject, we take a moment to try to develop the basic intuition for what PDAs are designed to accomplish.

Definition 2.8. Let Σ be a finite set. Then we define the **Kleene star** Σ^* to be the free monoid generated by the set Σ. A **language $\mathcal{L}$ with alphabet Σ** is just any subset of Σ^*. Given a word $w \in \mathcal{L}$, we write $l(w)$ to denote the **length of** w. That is, the number of elements of Σ which appear in w.

In this paper, we follow the standard practice of the field and reserve the symbol ϵ to denote the empty word.

Remark 2.9. Much of what follows will actually work for any **norm** on the language $\mathcal{L}$, not just the length. This level of generality will not be necessary for us.

Complexity in language theory is concerned with two distinct, but essentially equivalent, perspectives. The first perspective is the question of how complicated a grammar needs to be in order to build the language from its alphabet. The second perspective is the question of how sophisticated a machine needs to be to be able to detect whether a given word is in the language. The simplest possible machines are finite state automata. These machines have finitely many states, and a finite list of rules which allow one to move between states given an input element of Σ. The kinds of languages whose inclusion problem can be solved by finite state automata are the so-called **regular languages**. In this paper we will largely be concerned with machines that are one step higher in complexity: finite automata equipped with memory in the form of a stack.

Definition 2.10. A **push-down automaton**, or **PDA**, is a 7-tuple $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, where:

- Q is a finite set called the **states** of P;
- Σ is a finite set, disjoint from Q, called the **alphabet** of P;
- Γ is a finite set, disjoint from Σ and Q, called the **stack symbols** of P;
- $\delta : Q \times (\Sigma \cup \{\epsilon\}) \times (\Gamma \cup \{\epsilon\}) \to \mathcal{P}(Q \times \Gamma^*)$, where $\mathcal{P}$ denotes the power set, is the **transition function** of P;
- $q_0 \in Q$ is the **initial state** of P;

- $Z \in \Gamma$ is the **initial stack symbol** of P;
- $F \subseteq Q$ is the set of **final states** of P.

An **instantaneous description** of P is a triple, $(q, w, S) \in Q \times \Sigma^* \times \Gamma^*$. We interpret an instantaneous description as telling us which state we are currently in, the remainder of the word that is currently being processed, and the contents of the stack, where we understand the left most symbol of S as being the top of the stack. If (q, aw, AS) is an instantaneous description with $a \in \Sigma \cup \{\epsilon\}$ and $A \in \Gamma \cup \{\epsilon\}$, then we write

$$(q, aw, AS) \mapsto (q', w, \alpha S) \tag{1}$$

if $(q', \alpha) \in \delta(q, a, A)$. More generally, if (q, w, S) and (q', w', S') are two instantaneous descriptions of P, then we write

$$(q, w, S) \mapsto (q', w', S')$$

if there is a series of moves of the form (1) transforming (q, w, S) into (q', w', S'). Finally, we say that P **recognizes** a word $w \in \Sigma^*$ if

$$(q_0, w, Z) \mapsto (q_r, \epsilon, S)$$

where q_0 and Z are the initial state and stack symbol, respectively, $S \in \Gamma^*$, and $q_r \in F$ is a final state. The **language $\mathcal{L}(P)$ of** P is the set of all words that are recognized by P. We call P, as well as the language $\mathcal{L}(P)$, **unambiguous** if for any $w \in \mathcal{L}(P)$ there is precisely one sequence of moves of the form (1) that leads to a final state.

We think of a push-down automaton $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$ as being a machine that inputs a word w and outputs either "yes" or "no." It does so in the following way: Writing our word as $w = aw'$, with $a \in \Sigma$, it reads the letter a as well as the top of the stack, Z, and checks its available moves, as prescribed by $\delta(q_0, a, Z)$. These moves may include popping the top of the stack, pushing more symbols onto the stack, or some combination of both, along with a possible jump to a new state. If there are no available moves, then the machine outputs "no." Otherwise, it continues reading the remaining word w' in this way. When the entire word has been read, if the machine is in a final state it outputs "yes," while otherwise it outputs "no."

One should observe that the transition function of a PDA is permitted to read ϵ for either the input letter or the top stack symbol. This does *not* signify that the input word or stack must be empty for this transition to occur. It is more correct to interpret these transitions (sometimes called ϵ-moves in the literature) as saying that this transition can happen regardless of what the next input letter (or top of the stack) is. We will see examples of these kinds of transitions during the proof of the main theorem.

Languages which are of the form $\mathcal{L}(P)$ for some PDA P are called **context-free**. Importantly for us, one has the following foundational result about context-free languages.

Definition 2.11. Let $\mathcal{L}$ be a language over some finite alphabet Σ. Then the **generating function of** $\mathcal{L}$ is the formal power series

$$H_{\mathcal{L}}(t) := \sum_{w \in \mathcal{L}} t^{l(w)},$$

where $l(w)$ is the length of the word $w \in \mathcal{L}$.

Theorem 2.12 (Proposition 3.7, [BM05])**.** *Let $\mathcal{L}$ be a context-free language associated to an unambiguous PDA. Then $H_{\mathcal{L}}(t)$ is algebraic.*

Remark 2.13. We will see that Hilbert-Dyck series are in fact always $\mathbb{Z}$-algebraic (See [BD15, Definition 3]). We do not make use of this distinction in this paper.

Example 2.14. We have already seen that the Catalan numbers have an algebraic generating function. In fact, we can realize the Catalan numbers as the number of words of a given length in an unambiguous context-free language as follows.

Set $P = (Q, \Sigma, \Gamma, \delta, q_0, Z, F)$, where $Q = \{q_0, q_1\}$, $\Sigma = \{u, d\}$, $\Gamma = \{Z, A\}$, and $F = \{q_1\}$. Our transition function will be defined by the assignments:

$$\begin{aligned} \delta(q_0, u, Z) &= (q_0, AZ) \\ \delta(q_0, u, A) &= (q_0, AA) \\ \delta(q_0, d, A) &= (q_0, \epsilon) \\ \delta(q_0, \epsilon, Z) &= (q_1, \epsilon) \end{aligned}$$

Note that we follow the standard practice in the field that when the output of the transition function is a singleton, we suppress the set notation. Moreover, any transition whose output is the empty set is not written.

Also note that in the third transition, the right hand side is of the form (q_0, ϵ). By how instantaneous states were defined, one should always interpret this as saying the previous top of the stack has been popped. In other words, what was once an A at the top of the stack has been replaced with an empty character ϵ.

In words, the first two transitions indicate that when a u is read by the PDA, the symbol A is added to the top of the stack, while the third indicates that if a d is read, the stack is popped. Finally, the last transition indicates that, at any time when the stack only contains the initial symbol, you may move on to the final state. It is clear from this description that P is unambiguous, as the transition function has at most one move for any legal input. Moreover, a quick argument shows that $\mathcal{L}$ is precisely the language of Dyck paths. Our claim then follows from the fact that the number of Dyck paths of a given length agrees with the Catalan numbers.

2.3 Lingual categories

In their seminal work [SS17], Sam and Snowden develop a kind of language theory for categories, which they call lingual categories. Roughly speaking, these are categories whose morphisms can be

encoded as "well-behaved" languages. The upshot to this is one can use well known combinatorial theorems about the Hilbert series of these languages (See Definition 2.11) to conclude non-trivial facts about the dimension growth of modules over the category. In particular, we have the following.

Definition 2.15. Let $\mathcal{C}$ denote an essentially small category with no non-trivial endomorphisms, and write x for an object of $\mathcal{C}$. Then we write $|\mathcal{C}_x|$ for the set

$$|\mathcal{C}_x| = \{f : x \to y \mid y \text{ is an object of } \mathcal{C}\}/\sim,$$

where $\sim$ is the relation

$$f \sim g \iff h \circ f = g \text{ for some isomorphism } h.$$

The set $|\mathcal{C}_x|$ can be enhanced with the structure of a poset, with order relation given by

$$f \leq g \iff g = h \circ f \text{ for some morphism } h.$$

We say that the category $\mathcal{C}$ is an **unambiguous and context-free** if the following four conditions hold:

- the category $\mathcal{C}$ is **Gröbner** in the sense of Sam and Snowden [SS17, Definition 4.3.1];
- for every object x, there exists a set theoretic bijection

  $$\iota_x : |\mathcal{C}_x| \cong \mathcal{L}_x,$$

  where $\mathcal{L}_x$ is an unambiguous context-free language;
- for every object x, and every order ideal I of the poset $|\mathcal{C}_x|$, the image $\iota_x(I) \subseteq \mathcal{L}_x$ is also an unambiguous context-free language;
- there exists a function ν, called the **norm** of $\mathcal{C}$, from the set of isomorphism classes of objects of $\mathcal{C}$ to $\mathbb{N}$ such that for any object x and any morphism $f : x \to y$,

  $$\nu(y) = l(\iota_x(f)).$$

The theory of Gröbner categories was developed by Sam and Snowden. One can think of this condition as saying that the representation theory of the category admits a theory of Gröbner bases. This notation was extended by Miyata, Proudfoot, and the author in [MPR] to modules over categorical algebras. For the purposes of the present work, just note that the category $\mathcal{PT}^{op}$ was proved to be Gröbner by Barter in [Bar]. We therefore do not need to worry too much about this condition going forward.

Theorem 2.16 (Theorem 6.3.2, [SS17]). *Let $\mathcal{C}$ be an unambiguous context-free category with norm ν, and let M be a $\mathcal{C}$-module. If M is finitely generated, then the formal power series*

$$H_{M,\nu}(t) := \sum_x \dim_k(M(x))t^{\nu(x)}$$

is algebraic.

In view of Theorem 2.16, and Definition 2.6, our path forward has now become clear. Our first step will be to prove that the category $\mathcal{PT}^{op}$ is unambiguous and context-free, thereby generalizing the computation in Example 2.14. This will imply that the Hilbert series for finitely generated modules over $\mathcal{PT}^{op}$ are algebraic by Theorem 2.16. Following this, we leverage the fact that the forgetful functors $\mathcal{PT} \to \mathcal{RT}$ and $\mathcal{PT} \to \mathcal{T}$ have the so-called property (F) (see [PRb] and [SS17]). In particular, pulling back any finitely generated $\mathcal{RT}^{op}$ or $\mathcal{T}^{op}$-module to a module over $\mathcal{PT}^{op}$ preserves finite generation. This will imply that Hilbert-Dyck series of finitely generated modules over the categories $\mathcal{RT}^{op}$ and $\mathcal{T}^{op}$ will be algebraic, as desired.

3 The proof of the main theorem

In this section, we prove our main theorem via the strategy just outlined. In particular, our ultimate goal is the prove the following.

Theorem 3.1. *The category $\mathcal{PT}^{op}$ is unambiguous and context-free, with norm given by*

$$\nu(T) = 2 \cdot (\#\ \textit{of edges of}\ T).$$

Remark 3.2. We note that, with the norm defined as it is above, the associated Hilbert series are not exactly the previously defined Hilbert series (Definition 2.6). However, they are related by substituting t for $\sqrt{t}$. This operation clearly preserves the ultimate conclusion that the Hilbert series are algebraic, and we therefore stick with the aforementioned norm so that Lemma 3.10 remains true.

Proving this theorem happens in three steps. To begin, we must first decide on a means of encoding the morphisms of $\mathcal{PT}^{op}$ as words in a language.

For the remainder of this section, we fix a planar rooted tree T with n vertices.

Definition 3.3. We may assume that the vertices of T have been identified with $\{0, \ldots, n-1\}$. Then we define the alphabet Σ_T to be the finite set of symbols

$$\Sigma_T := \{u_i, d_i \mid i \in \{0, \ldots, n-1\}\}.$$

Thus, $|\Sigma_T| = 2n$. We will encode $|\mathcal{PT}^{op}_T|$ as a language over the alphabet Σ_T.

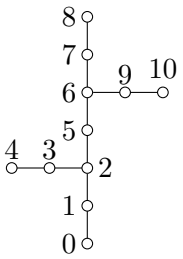

Figure 1: A planar tree T'

Let T' be a planar rooted tree, and let $\phi : T \to T'$ be an opposite planar contraction, with associated contraction $\psi : T' \to T$. Then we encode ϕ as a word in Σ_T as follows. Let w be the Dyck path associated to the tree T'. We add subscripts to the u's and d's in this path by looking at ψ applied to the head (directed, as always, away from the root) of the associated directed edge when a u is read, and the tail of the associated edge when a d is read. We will write w_ϕ to denote this word.

Finally, we write $\mathcal{L}_T$ for the language

$$\mathcal{L}_T := \{w_\phi \mid \phi : T \to T'\}.$$

Example 3.4. To see an example of the above encoding, let T' be the planar tree pictured in Figure 1. Then the Dyck path associated to T' is given by

$$uuuuddduuuudduuddddddd.$$

Assume now that T is a single edge, with root and head labeled by 0 and 1, respectively, and let $\psi : T' \to T$ be the planar contraction which sends the vertices labeled 9 and 10 to the head of T, and all other vertices to the root of T. Then,

$$w_\phi = u_0u_0u_0u_0d_0d_0u_0u_0u_0u_0d_0d_0u_1u_1d_1d_1d_0d_0d_0d_0.$$

Remark 3.5. The idea to encode these morphisms as modified Dyck paths was first accomplished by Barter in [Bar], where they were called Catalan words. Our encoding is different from his, but the basic premise is the same.

Also note that if $\zeta : V(T') \to V(T)$ is any function of sets (not necessarily a planar contraction), then one can similarly make sense of a word on the alphabet $\{u_i, d_i\}$ corresponding to ζ. We will use this observation during the proof of the main theorem.

It was already proven in [Bar] that $\mathcal{PT}^{op}$ is Gröbner. This resolves the first condition in Definition 2.15. We will now prove that $\mathcal{L}_T$ is always an unambiguous context-free language, thus verifying the second condition of Definition 2.15.

Proposition 3.6. *The language $\mathcal{L}_T$ is an unambiguous context-free language.*

Proof. Our goal will be to produce an unambiguous PDA, P_T, whose associated language is $\mathcal{L}_T$. We define the components of this PDA in turn as follows:

- The states of the PDA Q_T are given by the initial state q_0, the final state q_f, as well as a pair of states $q_{(e,u)}$ and $q_{(e,d)}$ for every edge e of T.
- The alphabet of the PDA is Σ_T, while the stack alphabet Γ_T contains the initial symbol Z, as well as symbols A_v for every vertex v of T.

To finish the construction of P_T, we need to detail our transition relations. We accomplish this by examining a handful of cases, which condition on the state we are currently situated at.

CASE: Transitions originating from the initial state q_0.

In this case we have

$$\begin{aligned}
\delta(q_0, u_0, \epsilon) &= (q_0, A_0A_0) \\
\delta(q_0, d_0, A_0) &= (q_0, \epsilon) \\
\delta(q_0, u_1, \epsilon) &= (q_{(e_1,u)}, A_1), \text{ where } e_1 \text{ is the first edge leaving the root.}
\end{aligned}$$

In other words, in this state the PDA can read either u_0, d_0, or u_1. while it is reading the symbols u_0 and d_0, it essentially acts as the PDA which recognizes the language of Dyck paths (see Example 2.14). If it reads the symbol u_1, however, it moves to the first non-initial state, while adding an A_1 to the top of the stack.

CASE: Transitions originating from the state $q_{(e,u)}$, where e is some edge of T whose head (directed away from the root) is the vertex v.

This case has two subcases. Firstly, assume that the vertex v is not a leaf, and that the smallest edge leaving v is e', with head v'. In this subcase we see,

$$\begin{aligned}
\delta(q_{(e,u)}, u_v, A_v) &= (q_{(e,u)}, A_vA_v) \\
\delta(q_{(e,u)}, d_v, A_v) &= (q_{(e,u)}, \epsilon) \\
\delta(q_{(e,u)}, u_{v'}, A_v) &= (q_{(e',u)}, A_{v'}A_v).
\end{aligned}$$

Note that these transitions are essentially the same as in the previous case, with one somewhat subtle difference. While this state can accept the letters u_v and $u_{v'}$, it can only do so if the top of

the stack displays the symbol A_v. The reason for this is that, based on the first two transitions, it is technically possible for a sufficient number of d_v symbols to be read so as to completely pop A_v off the stack. If one were to then try to add either an A_v or an $A_{v'}$ to the stack (e.g. by reading a u_v or $u_{v'}$-respectively), the input word could not possible be coming from a contraction. Indeed, if a u_v is read at this point, then the vertex map associated to the input word (see Remark 3.5) would have a disconnected preimage at v. If a $u_{v'}$ is read, then the vertex map does not preserve edge adjacency, and is therefore not a contraction either. These transitions are therefore modified to save us from accepting such a word.

In the second subcase, we assume that e is a leaf, and that the tail of this leaf is v'. We have

$$\begin{aligned}
\delta(q_{(e,u)}, u_v, A_v) &= (q_{(e,u)}, A_v A_v) \\
\delta(q_{(e,u)}, d_v, A_v) &= (q_{(e,u)}, \epsilon) \\
\delta(q_{(e,u)}, \epsilon, A_{v'}) &= (q_{(e,d)}, A_{v'}).
\end{aligned}$$

This subcase is similar to the previous. Because we have assumed that e is a leaf, there is nowhere to go but back down to v'. If at any point the top of the stack displays the symbol $A_{v'}$, then we have closed off all of the u_v letters in our word, and must now move back down the tree T to proceed with our mapping.

CASE: Transitions originating from the state $q_{(e,d)}$, where e is some edge of T whose tail (directed away from the root) is the vertex v.

Once again we have a few subcases. In the first subcase, we assume that v is not the root, and that there is some edge e', with head v', which is the smallest edge outgoing from v for which $A_{v'}$ has never appeared on the stack. We have

$$\begin{aligned}
\delta(q_{(e,d)}, u_v, A_v) &= (q_{(e,d)}, A_v A_v) \\
\delta(q_{(e,d)}, d_v, A_v) &= (q_{(e,d)}, \epsilon) \\
\delta(q_{(e,d)}, u_{v'}, A_v) &= (q_{(e',u)}, A_{v'} A_v).
\end{aligned}$$

In our second subcase, we assume that v is still not the root, no such edge e' exists, that e'' is the incoming edge of v, and that v'' is the other endpoint of e''. Further assume that v'' is also not the root. In the context of planar contractions, we will be in this case when we have already resolved how we are going to map the vertices of T' to the vertices of T above v. All that remains is to finish mapping vertices to v, and move back down the tree T. We have

$$\begin{aligned}
\delta(q_{(e,d)}, u_v, A_v) &= (q_{(e,d)}, A_v A_v) \\
\delta(q_{(e,d)}, d_v, A_v) &= (q_{(e,d)}, \epsilon) \\
\delta(q_{(e,d)}, \epsilon, A_{v''}) &= (q_{(e'',d)}, A_{v''}).
\end{aligned}$$

Repeating the previous subcase, but assuming that v'' is the root we have,

$$\begin{aligned}
\delta(q_{(e,d)}, u_v, A_v) &= (q_{(e,d)}, A_v A_v) \\
\delta(q_{(e,d)}, d_v, A_v) &= (q_{(e,d)}, \epsilon) \\
\delta(q_{(e,d)}, \epsilon, A_0) &= (q_{(e'',d)}, A_0) \\
\delta(q_{(e,d)}, \epsilon, Z) &= (q_{(e'',d)}, Z).
\end{aligned}$$

This subcase is largely the same as the previous, with the extra caveat that A_0 is the only stack symbol that may never be pushed to the stack. This will happened, from the perspective of contractions, if the only vertex of T' mapping to the root of T is the root of T'. We therefore have to be a bit careful to make sure the last two cases above are written separately.

For our penultimate subcase, we assume that $v = 0$ is the root, and that e' is the smallest unvisited outgoing edge with head v'.

$$\begin{aligned}
\delta(q_{(e,d)}, u_0, A_0) &= (q_{(e,d)}, A_0 A_0) \\
\delta(q_{(e,d)}, u_0, Z) &= (q_{(e,d)}, A_0 Z) \\
\delta(q_{(e,d)}, d_0, A_0) &= (q_{(e,d)}, \epsilon) \\
\delta(q_{(e,d)}, u_{v'}, \epsilon) &= (q_{(e',u)}, A_{v'})
\end{aligned}$$

Finally, assume that v is the root, and that all outgoing edges of v have been visited. Then there is nothing left to be done but resolve the symbols A_0 and move on to the final state.

$$\begin{aligned}
\delta(q_{(e,d)}, u_0, A_0) &= (q_{(e,d)}, A_0 A_0) \\
\delta(q_{(e,d)}, u_0, Z) &= (q_{(e,d)}, A_0 Z) \\
\delta(q_{(e,d)}, d_0, A_0) &= (q_{(e,d)}, \epsilon) \\
\delta(q_{(e,d)}, \epsilon, Z) &= (q_f, \epsilon)
\end{aligned}$$

We observe that, given any partial input, the transition function has at most one possible move. In particular, this PDA is unambiguous. It therefore remains to prove that the language of this PDA is $\mathcal{L}_T$. We proceed by induction on the number of edges of T.

In the case wherein T is a single point, the language $\mathcal{L}_T$ is clearly seen to be the language of Dyck paths, whereas the PDA P_T is easily seen to precisely agree with the PDA of Example 2.14. Assume then that $P_{T'}$ has associated language $\mathcal{L}_{T'}$ for all trees T' with $< n$ edges for some $n \geq 1$, and let T be a planar rooted tree with n edges. Write the planar rooted subtrees attached to the root of T as $T_1, \ldots, T_r$, ordered in the natural way. Then by induction, as well as the definition of P_T, we see that $\mathcal{L}(P_T)$ is the language of words w such that there exists some Dyck path $e = e_1 e_2 \cdots e_m$ on the alphabet $\{u_0, d_0\}$, as well as words $w_{T_i} \in \mathcal{L}_{T_i}$ with

$$w = e_1 \cdots e_{i_1} w_{T_1} e_{i_1+1} \cdots e_{i_2} w_{T_2} e_{i_2+1} \cdots e_m.$$

Note that the respective alphabets we are using for the words w_{T_i} are on the symbols $\{u_j, d_j\}$ where the permitted j are determined by the vertices appearing in the respective subtrees T_i. It is obvious that this decomposition describes the words of $\mathcal{L}_T$, as desired. □

Example 3.7. To make things a bit more concrete, we fully describe the PDA P_T in the case wherein T is the tree that looks like the letter Y, with root on the bottom leaf. In this case vertices are numbered $0, 1, 2$ and 3, in depth-first fashion, while we write our edges as e_1, e_2, e_3. Here, the index of the edge indicates the endpoint of the edge further from the root. Then we have

- $Q_T = \{q_0, q_{(e_1,u)}, q_{(e_2,u)}, q_{(e_2,d)}, q_{(e_3,u)}, q_{(e_3,d)}, q_{(e_1,d)}, q_f\}$
- $\Sigma_T = \{u_0, u_1, u_2, u_3, d_0, d_1, d_2, d_3\}, \Gamma_T = \{Z, A_0, A_1, A_2, A_3\}$

The complete list of our transition rules are given as follows:

$$\begin{aligned}
\delta(q_0, u_0, \epsilon) &= (q_0, A_0) \\
\delta(q_0, d_0, A_0) &= (q_0, \epsilon) \\
\delta(q_0, u_1, \epsilon) &= (q_{(e_1,u)}, A_1) \\
\delta(q_{(e_1,u)}, u_1, A_1) &= (q_{(e_1,u)}, A_1 A_1) \\
\delta(q_{(e_1,u)}, d_1, A_1) &= (q_{(e_1,u)}, \epsilon) \\
\delta(q_{(e_1,u)}, u_2, A_1) &= (q_{(e_2,u)}, A_2 A_1) \\
\delta(q_{(e_2,u)}, u_2, A_2) &= (q_{(e_2,u)}, A_2 A_2) \\
\delta(q_{(e_2,u)}, d_2, A_2) &= (q_{(e_2,u)}, \epsilon) \\
\delta(q_{(e_2,u)}, \epsilon, A_1) &= (q_{(e_2,d)}, A_1) \\
\delta(q_{(e_2,d)}, u_1, A_1) &= (q_{(e_2,d)}, A_1 A_1) \\
\delta(q_{(e_2,d)}, d_1, A_1) &= (q_{(e_2,d)}, \epsilon) \\
\delta(q_{(e_2,d)}, u_3, A_1) &= (q_{(e_3,u)}, A_3 A_1) \\
\delta(q_{(e_3,u)}, u_3, A_3) &= (q_{(e_3,u)}, A_3 A_3) \\
\delta(q_{(e_3,u)}, d_3, A_3) &= (q_{(e_3,u)}, \epsilon) \\
\delta(q_{(e_3,u)}, \epsilon, A_1) &= (q_{(e_3,d)}, A_1)
\end{aligned}$$

$$\begin{aligned}
\delta(q_{(e_3,d)}, u_1, A_1) &= (q_{(e_3,d)}, A_1A_1) \\
\delta(q_{(e_3,d)}, d_1, A_1) &= (q_{(e_3,d)}, \epsilon) \\
\delta(q_{(e_3,d)}, \epsilon, A_0) &= (q_{(e_1,d)}, A_0) \\
\delta(q_{(e_3,d)}, \epsilon, Z) &= (q_{(e_1,d)}, Z) \\
\delta(q_{(e_1,d)}, u_0, A_0) &= (q_{(e_1,d)}, A_0A_0) \\
\delta(q_{(e_1,d)}, u_0, Z) &= (q_{(e_1,d)}, A_0Z) \\
\delta(q_{(e_1,d)}, \epsilon, Z) &= (q_f, \epsilon).
\end{aligned}$$

In words: in the initial state, the PDA can process three letters: u_0, d_0 and u_1. In the first case, the symbol A_0 is pushed onto the stack, while in the second case A_0 is popped from the stack. In the third case, we jump to the first non-initial state, and push the symbol A_1 onto the stack. To relate this to the context of planar contractions, we known that the root of T' must map to the root of T. At this point we traverse T' using the usual Dyck path method, at each step keeping track of what vertex of T the contraction is mapping our current vertex of T' to. In particular, at the beginning we map everything to the root of T', until we step to the first vertex of T' which maps to the first non-root vertex of T (in the depth-first order). At this point, we have entered the regime of the word w_ϕ where the symbols u_1 and d_1 become active, while u_0 and d_0 become inactive. This will remain the case until we close off all of the symbols u_1 (as well as any intermediate symbols corresponding to the vertices of T accessible from the vertex 1 without passing through the root). That is to say, until the top of the stack is either the symbol A_0 or Z. Here our PDA will have the option to move into the section of the word corresponding to a region of T' which is once more being sent to the root. In this region, while we are free to use the symbols u_0 and d_0, we have to be careful not to suddenly begin reusing the symbol u_1. Indeed, once we have stepped back to the root in T, it would be a violation of the definition of contraction to return to the first vertex. This is why our states not only record which vertex of T we are currently mapping to, but also whether we have just entered this regime from below, or above.

The construction of the PDA in Proposition 3.6 inspires the following definition.

Definition 3.8. Let w_ϕ be a word in $\mathcal{L}_T$, and let (e, u) (resp. (e, d)) be a pair of an edge and a direction corresponding to a state in the PDA of Proposition 3.6. Then the letters appearing in the word w_ϕ which are processed by this PDA whilst in the state corresponding to (e, u) (resp. (e, d)) comprise what we call the (e, u) **(resp.** (e, d)**) section of the word** w_ϕ. The portion of w_ϕ which is parsed in the q_0-state of the PDA will be called the **initial section**. When the specific state of the PDA is not relevant to what is being discussed, we will often times just refer to the **sections of the word** w_ϕ. Very importantly, by how the PDA of Proposition 3.6 was defined, the only section that can possibly be empty is the final section. In other words, the PDA can not, in all but one case, skip states.

Example 3.9. If we take $w_\phi = u_0u_0d_0u_1u_1d_1d_1d_0$, then the initial section of w_ϕ is the subword $u_0u_0d_0u_1$, while the (e, u)-section is $u_1d_1d_1$, and the (e, d)-section is d_0.

In accordance with Definition 2.15, we have to verify that the order ideals of the poset $|\mathcal{PT}_T^{op}|$ are also unambiguous context-free languages, as well as the condition that our norm agrees with the length function on the language. The latter of these two goals is immediate from the relevant definitions.

Lemma 3.10. *The norm of ν defined in the statement of Theorem 3.1 respects the length on $\mathcal{L}_T$.*

Before we can begin the proof of our final required statement,we introduce some notation that will be useful.

Definition 3.11. Let $w_\phi, w_{\phi'} \in \mathcal{L}_T$ be two words. We say that w_ϕ is **strongly contained in** $w_{\phi'}$ if w_ϕ is a subword of $w_{\phi'}$, and whenever u_i, d_i are a pair in w_ϕ - that is this d_i is the alphabet symbol whose reading pops the original contribution of u_i from the stack of P_T - they are also a pair in $w_{\phi'}$. For instance, while *uudd* appears as a subword of *udududud*, it is not strongly contained in this word.

Remark 3.12. Counting patterns in Dyck paths is a relatively new field which seems to have many results analogous to the much more classical setting of counting patterns in permutations. See [BBFGPW14] for a treatment of these results. In this paper, we will be concerned with patterns that strongly appear in the word, as in the above definition. Our goal will be to show that the language of Dyck paths strongly containing any fixed pattern is actually unambiguous and context-free.

Proposition 3.13. *Let T be a fixed planar tree, and let I be an order ideal of $|\mathcal{PT}_T^{op}|$. Then the language associated to I is unambiguous and context-free.*

Proof. We first prove the proposition in the case where the order ideal is principal. In particular, we assume that

$$I = (\phi) = \{\phi' \mid \phi' = \phi'' \circ \phi \text{ for some } \phi''.\}$$

Translating everything through various definitions and equivalences, our goal in this proposition is to prove the following. We must show that the language

$$\{w_{\phi'} \mid w_{\phi'} \text{ strongly contains } w_\phi\} \subseteq \mathcal{L}_T,$$

is unambiguous and context-free.

Consider the PDA with component parts written as:

- $Q = \{q_0^i, q_{(e,u)}^{i_{(e,u)}}, q_{(e,d)}^{i_{(e,d)}}, q_{found}, q_f\}$
- $\Gamma = \{Z, (w_{\phi,j})\}$,

where i ranges from 0 to the size of the initial section of w_ϕ, and each $i_{(e,u)}$ or $i_{(e,d)}$ range from 0 to the length of the (e,u) or (e,d) section of the word w_ϕ, respectively.

In words, the states of the PDA will encode both the section of the word we are currently parsing, as well as how much of w_ϕ we have observed thus-far. Thus, this PDA can be viewed as a

kind of refinement of the PDA of Proposition 3.6. we split each state of the PDA of Proposition 3.6 into a collection of states corresponding to the length of the corresponding section of the word w_ϕ. Our stack symbols include the initial symbol Z as well as symbols corresponding to the subwords of w_ϕ comprised of the first j letters for each $0 \leq j \leq l(w_\phi)$. This will be done so that when a letter of our word is read, the stack can be made to remember what part of the word w_ϕ it may correspond to. Using the stack in this way also assures that we will have found a strongly included copy of w_ϕ, and not just a normal copy.

We describe the transitions of this PDA in a particular example, and then discuss how they generalize. Let T be a single edge, and $w_\phi = u_0 d_0 u_1 u_1 d_1 d_1$. In this case our PDA has states given by

$$Q = \{q_0^0, q_0^1, q_0^2, q_{(e,u)}^0, q_{(e,u)}^1, q_{(e,u)}^2, q_{(e,u)}^3, q_{(e,d)}^0, q_{found}, q_f\},$$

while our stack alphabet is given by

$$\Gamma = \{Z, (U_0), (U_0 D_0), (U_0 D_0 U_1), (U_0 D_0 U_1 U_1), (U_0 D_0 U_1 U_1 D_1), (U_0 D_0 U_1 U_1 D_1 D_1))\}$$

Our transition function is given as follows

$$\begin{aligned}
\delta(q_0^0, u_0, Z) &= (q_0^1, (U_0)Z) \\
\delta(q_0^1, u_0, \epsilon) &= (q_0^1, (U_0)) \\
\delta(q_0^1, d_0, (U_0)) &= (q_0^2, \epsilon) \\
\delta(q_0^2, u_0, \epsilon) &= (q_0^2, (U_0 D_0)) \\
\delta(q_0^2, d_0, (U_0)) &= (q_0^2, \epsilon) \\
\delta(q_0^2, d_0, (U_0 D_0)) &= (q_0^2, \epsilon) \\
\delta(q_0^2, u_1, \epsilon) &= (q_{(e,u)}^0, (U_0 D_0 U_1)) \\
\delta(q_{(e,u)}^0, u_1, \epsilon) &= (q_{(e,u)}^1, (U_0 D_0 U_1 U_1)) \\
\delta(q_{(e,u)}^1, u_1, \epsilon) &= (q_{(e,u)}^1, (U_0 D_0 U_1 U_1)) \\
\delta(q_{(e,u)}^1, d_1, (U_0 D_0 U_1 U_1)) &= (q_{(e,u)}^2, \epsilon) \\
\delta(q_{(e,u)}^2, u_1, \epsilon) &= (q_{(e,u)}^2, (U_0 D_0 U_1 U_1 D_1)) \qquad (2) \\
\delta(q_{(e,u)}^0, d_1, (U_0 D_0 U_1 U_1 D_1)) &= (q_{(e,u)}^2, \epsilon) \qquad (3) \\
\delta(q_{(e,u)}^2, d_1, (U_0 D_0 U_1 U_1)) &= (q_{(e,u)}^3, \epsilon) \\
\delta(q_{(e,u)}^2, d_1, (U_0 D_0 U_1)) &= (q_{(e,d)}^0, \epsilon) \\
\delta(q_{(e,u)}^3, u_1, \epsilon)) &= (q_{(e,u)}^3, (U_0 D_0 U_1 U_1 D_1 D_1)) \\
\delta(q_{(e,u)}^3, d_1, (U_0 D_0 U_1 U_1 D_1 D_1)) &= (q_{(e,u)}^3, \epsilon) \\
\delta(q_{(e,u)}^3, d_1, (U_0 D_0 U_1 U_1)) &= (q_{(e,u)}^3, \epsilon) \\
\delta(q_{(e,u)}^3, d_1, (U_0 D_0 U_1 U_1 D_1)) &= (q_{(e,u)}^3, \epsilon) \\
\delta(q_{(e,u)}^3, d_1, (U_0 D_0 U_1)) &= (q_{(e,u)}^3, \epsilon)
\end{aligned}$$

$$
\begin{aligned}
\delta(q^0_{(e,d)}, \epsilon, \epsilon) &= \delta(q_{found}, \epsilon, \epsilon) \\
\delta(q_{found}, u_0, \epsilon) &= (q_{found}, (U_0 D_0 U_1 U_1 D_1 D_1)) \\
\delta(q_{found}, d_0, (U_0 D_0 U_1 U_1 D_1 D_1)) &= (q_{found}, \epsilon) \\
\delta(q_{found}, d_0, (U_0 D_0)) &= (q_{found}, \epsilon) \\
\delta(q_{found}, d_0, (U_0)) &= (q_{found}, \epsilon) \\
\delta(q_{found}, \epsilon, Z) &= (q_f, \epsilon).
\end{aligned}
$$

To summarize, the states of the PDA indicate both the section of w_ϕ which has been thus far detected, as well as the section of the input word currently being processed. We note that sections of w_ϕ must appear in the corresponding sections of the word being processed, and so there is nothing lost by partitioning our states this way. The stack symbols are meant to indicate how much of the word w_ϕ has been processed at the time the current letter is being read. This way, for instance, when certain symbols are popped from the stack, the PDA can determine when it needs to leave a certain state as the currently observed partial pattern can no longer be completed. On the other hand, this also allows for us to know whether or not a currently being read down symbol corresponds to the correct up symbol in so far as the pattern is concerned. In particular, it guarantees the copy of w_ϕ being detected is strongly included in the word, and not just a subword. The state q_{found} is entered when the word w_ϕ has been fully detected, and we no longer have to worry about tracking exactly what is being read.

In so far as the process of tracking the word w_ϕ, there are a few things that can happen that are out of the expectation of w_ϕ. For instance, after having read a u_0 at the beginning of the word, we have entered the state q^1_0, indicating we are in the initial section of our word, and have already found the first desired up letter of w_ϕ. In-so-far as w_ϕ is concerned, the expectation is to next read a down letter. As one can see from the above PDA description, if, another up letter is read instead, then we do not change states. More generally, if we are in a state that was entered after successfully reading an expected up letter of w_ϕ, and unexpectedly read another up letter, then the PDA stays in this state. We think of this unexpected up letter as a kind of buffer for the next kind of unexpected move.

If we have entered a state having correctly read an up letter, but then unexpectedly read a down letter, then it is impossible for the most recently read up letter to be part of a strong copy of w_ϕ. If we still have some of the aforementioned “buffer" up letters (which will be indicated by the top of the stack having the correct symbol), then we may stay in this state and wait for the next letter. Otherwise, we must abandon this state and fall back to a state that indicates we haven't seen as much of the pattern as originally thought. A similar phenomenon happens if we just read an expected down letter, and then read an unexpected down letter.

The final kind of unexpected letter is seen in transitions (2) and (3). At this point in the PDA, the partial pattern $u_0 d_0 u_1 u_1 d_1$ has been detected, and the next letter read is a u_1. Whenever one follows a down move with an unexpected up move, you have entered a section of your word which can no longer contribute to completing the partial pattern originally being observed. Therefore, the

PDA must fall back to an earlier state and begin looking for a new pattern. If it finds this new pattern, then it enters the "found" state. Otherwise the symbol $(U_0D_0U_1U_1D_1)$ will eventually be popped from the stack, indicating to the PDA it must return to the original partial pattern and attempt to complete it.

Note that the above four cases illustrate why it is not necessary for the PDA to backtrack. It is essentially keeping track at all times of whether strong inclusion of w_ϕ is possible, not exactly where that copy is located.

Observe many states and transitions are not strictly necessary in this example. For instance, once two u_1 have been observed, it is impossible for us to not find our pattern. We present this example in this overly long way just to make it more clear how it generalizes. Finally, Observe that our PDA is unambiguous as for every input letter and top of the stack, there is at most one transition available.

Now that we have treated the case of a principal order ideal, we must treat the general case of an order ideal I. To begin, recall that Barter [Bar] has already shown that the poset $|\mathcal{PT}_T^{op}|$ is Noetherian. In particular, all order ideals can be expressed as a finite union $I = \cup_{i=1}^{N}(\phi_i)$. We therefore must prove

$$\{w_\phi \mid w_\phi \text{ strongly contains at least one of the words } w_{\phi_i}\}$$

is unambiguous and context-free. Because the list of desired patterns is finite, it is clear that one may modify the above so that our states record how much we have seen of each of the patterns independently. Our stack symbols will encode the currently observed partial patterns for each of the finitely many target patterns. □

4 Applications of the main theorem

In this section, we see certain concrete applications of the main Theorem 1. Our first application involves counting a certain recursive invariant of Dyck words.

Definition 4.1. Given a Dyck path w, we define its **degree sequence** as the $(\frac{l(w)}{2}+1)-tuple$ $(\alpha_w^{(v)})_{v\in T(w)}$ encoding the degree sequence of the associated planar rooted tree $T_p(w)$. If α_w is the degree sequence of some Dyck path, then we write

$$||\alpha_w||_* := \sum_{v\in T(w)} \binom{\alpha_w^{(v)}}{2}$$

for the **star-norm** of α_w.

Our first application involves the generating function of the star-norm.

Theorem 4.2. *The generating function*

$$H_{star}(t) := \sum_w ||\alpha_w||_* t^{l(w)/2}$$

is algebraic.

Proof. We will encode the series $H_{star}(t)$ as the Hilbert series of some finitely generated $\mathcal{PT}^{op}$-module. Let T be a tree, and let $\text{cone}(T)$ denote the graph obtained by adding a single vertex and connecting it to every vertex of T. For instance, the cone of a single edge is a triangle, while the cone of a path with three vertices is two triangles glued along an edge. In [PRb, Theorem 1.6], a finitely generated $\mathcal{PT}^{op}$-module M is constructed with the property that for any Dyck path w,

$$M(T_p(w)) = H_1(\text{UConf}_2(\text{cone}(T_p(w)); \mathbb{Q})),$$

where $\text{UConf}_2(\text{cone}(T_p(w)))$ is the two particle unordered configuration space of the graph $\text{cone}(T_p(w))$ (see [ADCK19][Ram18][Far06], for instance). It is also shown in [PRb, Example 3.11] that

$$\dim_{\mathbb{Q}}(H_1(\text{UConf}_2(\text{cone}(T_p(w)); \mathbb{Q}))) = l(w)/2 + ||\alpha_w||_*.$$

Therefore, the Hilbert series of the module M is given by

$$H_M(t) = \sum_w (l(w)/2 + ||\alpha_w||_*) t^{l(w)/2} = \sum_w (l(w)/2) t^{l(w)/2} + H_{star}(t).$$

On the other hand, one can see that

$$\sum_w (l(w)/2) t^{l(w)/2} = t \cdot \frac{\partial}{\partial t} (\sum_w t^{l(w)/2}).$$

It is classically known that

$$\sum_w t^{l(w)/2} = \frac{1 - \sqrt{1-4t}}{2t},$$

whence

$$\frac{\partial}{\partial t} (\sum_w t^{l(w)/2}) = \frac{(-2t - \sqrt{1-4t} + 1)}{(2t^2 \sqrt{1-4t})}$$

is an algebraic function. It follows that $\sum_w (l(w)/2) t^{l(w)/2}$ is algebraic, and the same must be true of $H_{star}(t)$. □

Remark 4.3. The observation that the derivative of $\sum_w (l(w)/2) t^{l(w)/2}$ is once again algebraic is really a specific case of a much more general phenomenon related to D-finite series. See [BD15, Proposition 4] for more on this.

Note that the above Theorem is actually just applying the techniques of this paper to the case of the first homology of tree configuration spaces. In fact, the results of [PRb] tell us that all of the homology groups are finitely generated as $\mathcal{PT}^{op}$-modules. In particular, the same proof technique as the above can be used to prove a variety of more complicated numerical invariants of degree sequences of Dyck paths have algebraic generating functions. See [Ram18] for what these formulas look like.

For our second application, we look to counting subtrees of a given tree. The generating function for counting subtrees is of considerable interest in computer science (see [Rus81], For instance). In this work we consider the generating function of the following collections of numbers.

Definition 4.4. Let w be a Dyck path with associated planar rooted tree $T_p(w)$, and let $l \geq 2$ be fixed. Then we set $s_l(w)$ to be the invariant

$$s_l(w) = \begin{cases} \#(\text{planar rooted subtrees of } T_p(w)) & \text{if } T_p(w) \text{ has no more than } l \text{ leaves} \\ 0 & \text{otherwise.} \end{cases}$$

Example 4.5. For instance, if $l = 2$, then those w for which $s_l(w) \neq 0$ precisely correspond to planar rooted paths. If $T_p(w)$ is a planar rooted path, then each subtree that is not a single vertex is in bijection with unordered pairs of vertices of $T_p(w)$. In particular,

$$s_2(w) = \begin{cases} (l(w)/2+1) + \binom{l(w)/2+1}{2} & \text{if } T_p(w) \text{ is a path} \\ 0 & \text{otherwise} \end{cases} = \begin{cases} \binom{l(w)/2+2}{2} & \text{if } T_p(w) \text{ is a path} \\ 0 & \text{otherwise.} \end{cases}$$

Thus,

$$\sum_w s_2(w) t^{l(w)/2} = 1 + \sum_{n \geq 1} n \binom{n+2}{2} t^n$$

where the factor of n is the number of Dyck paths corresponding to paths with $n \geq 1$ edges, and the plus one comes from the case of the empty path corresponding to $T_p(w)$ being a single point. This implies that the generating function for $s_2(w)$ is rational. In general, we will see that $s_l(w)$ has an algebraic generating function.

Theorem 4.6. *Let $l \geq 2$ be fixed. Then the generating function*

$$H_l(t) := \sum_w s_l(w) t^{l(w)/2}$$

is algebraic.

Proof. We encode $H_l(t)$ as the Hilbert series of some $\mathcal{PT}^{op}$-module. In [PRb, Theorem 1.9], Proudfoot and the author construct a finitely generated $\mathcal{PT}^{op}$-module M such that

$$\dim_{\mathbb{Q}}(M(T)) = \begin{cases} KL_1(\text{cone}(T)) & \text{if } T \text{ has} \leq l \text{ leaves} \\ 0 & \text{otherwise,} \end{cases}$$

where $KL_1(\text{cone}(T))$ is the first Kazhdan-Lusztig coefficient of the graphical matroid associated to cone(T) (see [EPW16, Proposition 2.12]). For our purposes, what is important is the fact that for any matroid $\mathcal{M}$,

$$KL_1(\mathcal{M}) = \#(\text{codimension 1 flats of } \mathcal{M}) - \#(\text{dimension 1 flats of } \mathcal{M}).$$

The graphical matroid of cone(T) has a 1 dimensional flat for every edge, and therefore

$$\#(\text{dimension 1 flats of cone}(T)) = |E_T| + |V_T| = 2|E_T| + 1.$$

On the other hand, the number of codimension 1 flats of cone(T) are easily seen to be in bijection with subtrees of T. Thus,

$$H_M(t) = \sum_w \dim_{\mathbb{Q}}(M(T_p(w)))t^{l(w)/2} = \sum_{T_p(w) \text{ has} \leq l \text{ leaves}} (s_l(w) - (l(w)+1))t^{l(w)/2}.$$

On the other hand, we can define a $\mathcal{PT}^{op}$-module N such that,

$$N(T) := \mathbb{Q}E_{\text{cone}(T)},$$

the vector space with basis indexed by the edges of cone(T). N is finitely generated by a single edge and a single vertex. Moreover, it contains a submodule N' which is generated by the pieces $N(T)$, where T has strictly more than l leaves. The quotient N/N' has the property that

$$N/N'(T) = \begin{cases} \mathbb{Q}E_{\text{cone}(T)} & \text{if } T \text{ has} \leq l \text{ leaves} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the Hilbert series of N/N' is precisely,

$$\sum_{T_p(w) \text{ has} \leq l \text{ leaves}} (l(w)+1)t^{l(w)/2}.$$

Because N is finitely generated, N/N' is as well, and therefore this Hilbert series must be algebraic. This implies that our generating function is also algebraic, as desired. □

References

[ABB97] Jean-Michel Autebert, Jean Berstel, and Luc Boasson, *Context-free languages and push-down automata*, Handbook of formal languages, Springer, 1997, pp. 111–174.

[ADCK19] Byung Hee An, Gabriel C. Drummond-Cole, and Ben Knudsen, *Subdivisional spaces and graph braid groups*, Doc. Math. **24** (2019), 1513–1583.

[Bar] Daniel Barter, *Noetherianity and rooted trees*, arXiv:1509.04228.

[BBFGPW14] Axel Bacher, Antonio Bernini, Luca Ferrari, Benjamin Gunby, Renzo Pinzani, and Julian West, *The dyck pattern poset*, Discrete Mathematics **321** (2014), 12–23.

[BD15] Cyril Banderier and Michael Drmota, *Formulae and asymptotics for coefficients of algebraic functions*, Combinatorics, Probability and Computing **24** (2015), no. 1, 1–53.

[BM05] Mireille Bousquet-Mélou, *Algebraic generating functions in enumerative combinatorics and context-free languages*, Annual Symposium on Theoretical Aspects of Computer Science, Springer, 2005, pp. 18–35.

[CEF15] Thomas Church, Jordan S. Ellenberg, and Benson Farb, *FI-modules and stability for representations of symmetric groups*, Duke Math. J. **164** (2015), no. 9, 1833–1910.

[CF13] Thomas Church and Benson Farb, *Representation theory and homological stability*, Adv. Math. **245** (2013), 250–314.

[Drm04] M. Drmota,*Combinatorics and asymptotics on trees*, Cubo Journal (2004).

[EPW16] Ben Elias, Nicholas Proudfoot, and Max Wakefield, *The Kazhdan-Lusztig polynomial of a matroid*, Adv. Math. **299** (2016), 36–70.

[Far06] Daniel Farley, *Homology of tree braid groups*, Topological and asymptotic aspects of group theory, Contemp. Math., vol. 394, Amer. Math. Soc., Providence, RI, 2006, pp. 101–112.

[MPR] Dane Miyata, Nicholas Proudfoot, and Eric Ramos, *The categorical graph minor theorem*, arXiv:2004.05544.

[Ott48] Richard Otter, *The number of trees*, Annals of Mathematics (1948), 583–599.

[PRa] Nicholas Proudfoot and Eric Ramos, *The contraction category of graphs*, arXiv:1907.11234.

[PRb] Nicholas Proudfoot and Eric Ramos, *Functorial invariants of trees and their cones*, arXiv:1903.10592.

[Ram18] Eric Ramos, *Stability phenomena in the homology of tree braid groups*, Algebraic & Geometric Topology **18** (2018), 2305–2337.

[Rus81] Frank Ruskey, *Listing and counting subtrees of a tree*, SIAM Journal on Computing **10** (1981), 141–150.

[SS17] Steven V. Sam and Andrew Snowden, *Gröbner methods for representations of combinatorial categories*, J. Amer. Math. Soc. **30** (2017), no. 1, 159–203.