Wilf classes of non-symmetric operads

Andrey T. Cherkasov
atcherkasov@hotmail.com
HSE University
Moscow, Russia

Dmitri Piontkovski∗
dpiontkovski@hse.ru
HSE University
Moscow, Russia

ABSTRACT

Two operads are said to belong to the same Wilf class if they have the same generating series. We discuss possible Wilf classifications of non-symmetric operads with monomial relations. As a corollary, this would give the same classification for the operads with a finite Groebner basis.

Generally, there is no algorithm to decide whether two finitely presented operads belong to the same Wilf class. Still, we show that if an operad has a finite Groebner basis, then the monomial basis of the operad forms an unambiguous context-free language. Moreover, we discuss the deterministic grammar which defines the language. The generating series of the operad can be obtained as a result of an algorithmic elimination of variables from the algebraic system of equations defined by the Chomsky–Schützenberger enumeration theorem. We then focus on the case of binary operads with a single relation. The approach is based on the results by Rowland on pattern avoidance in binary trees. We improve and refine Rowland’s calculations and empirically confirm his conjecture. Here we use both the algebraic elimination and the direct calculation of formal power series from algebraic systems of equations. Finally, we discuss the connection of Wilf classes with algorithms for calculation of the Quillen homology of operads.

CCS CONCEPTS

• Mathematics of computing → Generating functions; • Computing methodologies → Algebraic algorithms.

KEYWORDS
	nonsymmetric operads, unambiguous grammar, Wilf class, Groebner bases in operads, tree pattern avoidance, system of algebraic equations, generating function, operad homology

1 INTRODUCTION

An algebraic operad (either symmetric or not) is the union of a sequence of vector spaces, \( P = P_0 \cup P_1 \cup \ldots \) . So, the first invariant of the operad is the sequence of dimensions \( \dim P_0, \dim P_1, \ldots \) either per se or in the form of the generating function, which is called the generating series of the operad.

Suppose that we know a Groebner basis of the operad. Note that the generating series is equal to the one of the associated monomial operad. The generating series of the monomial operad is equal to the generating series of the set of trees avoiding certain patterns, see [12]. These patterns correspond to the leading monomials of the Groebner basis. Two sets of tree patterns are called Wilf equivalent if the corresponding sets of avoiding trees have the same generating series [4, 18]. Similarly, we call two operads (with the same generating sets) Wilf equivalent if their generating series coincide.

We see that two operads belong to the same Wilf equivalence class if the sets of leading monomials of their Groebner bases (with respect to the same generators, for arbitrary orderings) are Wilf equivalent as the sets of tree patterns. Since both operads are defined via the same sequence of vector space dimensions \( \dim P_0, \dim P_1, \ldots \) , in some (weak) sense, they can be considered as flat deformations of each other. Under some additional conditions (such as Koszulity) all operads of the same Wilf class have the same homological invariants. Wilf equivalence can be considered as the weakest version of the isomorphism of operads.

In this paper, we focus on non-symmetric operads. Is there an algorithm to determine if two finitely presented operads (defined by finite lists of generators and relations over a computable1 field) are Wilf equivalent? Generally, there is not. Such a general algorithm does not exist even if we assume that the operads are quadratic (see Proposition 3.1 below).

A number of important operads, however, have finite Groebner bases, so that they belong to the Wilf classes of finitely presented monomial operads. The generating series of the finitely presented monomial operads are algebraic functions [12]. To determine the Wilf class, one needs to (1) construct a system of algebraic equations which defines the generating series and (2) find the generating series from the system. The problem (2) is a standard problem of the algebraic elimination theory; its solution is based on the Groebner bases theory. Three algorithms for the problem (1) are discussed in [12] and one algorithm is given in [11]; the first two are generalized versions of the algorithms by Rowland [18] for enumeration pattern avoiding binary trees.

Here we go further by proving that such a series is \( \mathbb{N} \)-algebraic, that is, it is equal to the generating function of some unambiguous context-free language. It follows from

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ISSAC ’21, July 18–23, 2021, Virtual Event, Russian Federation.
© 2021 Association for Computing Machinery.

∗Corresponding author.

1We call a field computable if there exist finite presentations for all its elements and algorithms for the arithmetic operations.
Theorem 1.1 (Corollary 4.3). Let \( P \) be a finitely generated non-symmetric operad having a finite Groebner basis of relations. Then its natural monomial basis forms a deterministic context-free language.

In Section 4, we discuss the algorithm to construct the unambiguous grammar defining the language. Then the Chomsky–Schützenberger enumeration theorem gives a way to construct the system of algebraic equations from the grammar. We also see that the systems obtained by the methods of [12, Section 2.2] give equivalent systems.

Unfortunately, this algorithm in some cases produces different systems for two operads of the same Wilf class (see example in Section 5). So, we should provide an elimination of variables. We provide a number of computer experiments in the case of operads generated by one binary operation with one monomial relation. Such operads encode the varieties of binary (non-associative) algebras which have been widely studied for decades. If we fix the degree (= the arity) of the relation, then Wilf classes for operads correspond to the Wilf classes of the binary tree patterns with a degree (= the arity) of the relation, then Wilf classes for operads which have been widely studied for decades. If we fix the degree (= the arity) of the relation, then Wilf classes for operads correspond to the Wilf classes of the binary tree patterns with a degree (= the arity) of the relation, then Wilf classes for operads which have been widely studied for decades. If we fix the degree (= the arity) of the relation, then Wilf classes for operads correspond to the Wilf classes of the binary tree patterns with a degree (= the arity) of the relation, then Wilf classes for operads which have been widely studied for decades.

In Section 5, we discuss and particularly improve some of the classes in Section 3. In Section 4 we show that the finitely presented operads have a finite Groebner basis of relations, and satisfy some fixed polynomial identities. We assume that \( \Omega \) is a finite union of finite sets \( \Omega = \Omega_2 \cup \cdots \cup \Omega_k \) where the elements \( \omega \) of \( \Omega_i \) act on each algebra \( A \) in \( W \) as \( t \)-linear operations, \( \omega : A^{\otimes t} \to A \). The variety is defined by two sets, the signature \( \Omega \) and a set of defining identities \( R \). By the linearization process, one can assume that \( R \) consists of multilinear identities.

Consider the free algebra \( P(W) \) on a countable set of indeterminates \( X = \{ x_1, x_2, \ldots \} \). Let \( P_n \subset F \) be the subspace consisting of all multilinear generalizations of homogeneous polynomials on the variables \( \{ x_1, \ldots, x_n \} \), that is, \( P_n \) is the component \( P(W)(1, \ldots, 1, 0, 0, \ldots) \) with respect to the \( \mathbb{Z}^\infty \) grading by the degrees on \( x_i \).

**Definition 2.1.** Given such a variety \( W \), the sequence \( P_W = P := \{ P_1, P_2, \ldots \} \) of the vector subspaces of \( P(W)(x) \) is called an operad.

The \( n \)-th component \( P_n \) may be identified with the set of all derived \( n \)-linear operations on the algebras of \( W \); in particular, \( P_n \) carries the natural structure of a representation of the symmetric group \( S_n \). Such a sequence \( Q = \{ Q(n) \}_{n \in \mathbb{Z}} \) of representations \( Q(n) \) of the symmetric groups \( S_n \) is called an \( S \)-module, so that an operad carries a structure of \( S \)-module with \( P_n = P(n) \). The compositions of operations (that is, a substitution of an argument \( x_i \) by a result of another operation with a subsequent monotone re-numbering of the inputs to avoid repetitions) gives natural maps of \( S \)-modules \( \omega_i : P(m) \otimes P(n) \to P(n+m-1) \). Note that the axiomatization of these operations gives an abstract definition of operads, see [15].

The signature \( \Omega \) can be considered as a sequence of subsets of \( P \) with \( \Omega_n \subset P_n \). Then \( \Omega \) generates the operad \( P \) up to the \( S \)-module structure and the compositions \( \omega_i \) so that it is called the set of generators of the operad.

More generally, the \( S \)-module \( X \) generated by \( \Omega \) is called the (minimal) module of generators of the operad \( P \). It can be also defined independently of \( \Omega \) as \( X = \bigoplus \{ P_n / (P_n \circ P_k) \} \) where \( P_k = P_2 \cup P_3 \cup \cdots \) and \( \circ \) denotes the span of all compositions of two \( S \)-modules. Then one can define a variety \( W \) corresponding to a (formal) operad \( P \) by picking a set \( \Omega \) of generators of \( X \) to be the signature and considering all relations in \( P \) as the defining identities of the variety, so that the variety \( W \) can be recovered by \( P \) “up to a change of variables”. One can also consider the algebras from \( W \) as vector spaces \( V \) with the actions \( \rho(n) : V^{\otimes n} \to V \) compatible with compositions and the \( S \)-module structures, so that the algebras of \( W \) are recovered by \( P \) up to isomorphisms.

---

\(^1\)We consider here varieties of algebras without constants, with the identity operator and without other unary operations

\(^2\)More precisely, symmetric connected \( k \)-linear operad with identity
Given an $S$-module $X$, one can also define a free operad $F(X)$
generated by $X$ as the span of all possible compositions of a basis of $X$
modulo the action of symmetric groups. For example, the free operad $F(S\Omega)$
on the free $S$-module $S\Omega$ corresponds to the variety of all algebras of signature $\Omega$.

Suppose that the defining identities $R$ of the variety $W$ can be chosen in such a way that for each
\[ f(x_1, \ldots, x_n) = \sum_i \alpha_i f_i(x_1, \ldots, x_n) \in R, \]
where $\alpha_i \in k$ and $f_i$ are monomials (that is, the compositions of the
operations from $\Omega$), in all monomials $f_i$ the variables $x_1, \ldots, x_n$
 occur in the same relative order. A standard example is the variety of associative algebras, see below. Then one can associate to $W$ a simpler non-symmetric operad.

Generally, a non-symmetric operad is a union $P = P_1 \cup P_2 \cup \ldots$
with the compositions $s_i$ as above but without the actions of the
symmetric groups. To distinguish them, we refer to the operads
defined above as symmetric. Each symmetric operad can be considered
as a non-symmetric one. To each non-symmetric operad $P$ one
 can assign a symmetric operad $P$ where $P_n = S_n P_n$ is a free
$S_n$ module generated by $P_n$. Then $P$ is called a symmetrization
of $P$. In particular, here $\dim P_n = n! \dim P_n$.

An $n$-th codimension of a variety $W$ is just the dimension of the
respective operad component: $c_p(W) = \dim_i P_n$ for $P = P_W$. We
consider both exponential and ordinary generating functions for
this sequence:
\[ E_P(z) := \sum_{n \geq 1} \frac{\dim P(n)}{n!} z^n, G_P(z) := \sum_{n \geq 1} \dim P(n) z^n. \quad (2.1) \]
For example, if $P$ is a symmetrization of a non-symmetric operad $P$
then $E_P(z) = G_P(z)$. By a generating series of a symmetric operad
$P$ we mean the exponential generating function $P(z) = E_P(z)$.
In contrast, for a non-symmetric operad $P$ its generating series is
defined as the ordinary generating function $P(z) = G_P(z)$. In the
variety of both, the ordinary and exponential versions of the
codimension series are studied.

If the set $\Omega$ is finite then the series $P(z)$ defines an analytic function
in the neighborhood of zero. For example, the non-symmetric operad $Ass$
of associative algebras is the operad defined by one binary operation $m$
(multiplication) subject to the relation $m(m(x_1, x_2), x_3) = m(x_1, m(x_2, x_3))$
which is the associativity identity. Its $n$-th component consists of the only equivalence class
of all arity $n$ compositions of $m$ with itself modulo the relation,
so that $Ass(z) = G_{Ass}(z) = \frac{z^2}{(1-z)^3}$. Its symmetrization is the symmetric
operad $\mathcal{A}_{Ass}$ generated by two operations $m(x_1, x_2)$ and
$m'(x_1, x_2) = m(x_2, x_1)$ with the $S_2$ action (12)$m = m$
subject to all the relations of the form $m(m(x_1, x_j), x_k) = m(x_1, m(x_j, x_k))$.
By the above, we have $E_{\mathcal{A}_{Ass}}(z) = \mathcal{A}_z(z) = Ass(z) = G_{Ass}(z)$,
so that $\dim \mathcal{A}_{Ass} = n!$.

### 2.2 Monomial bases and Groebner bases in operads

As the motivation for studying the monomial operads in the operad theory comes often via Groebner bases, we briefly recall here some

| 2.2 Monomial bases and Groebner bases in operads |
|--------------------------------------------------|
| As the motivation for studying the monomial operads in the operad theory comes often via Groebner bases, we briefly recall here some basic facts of the theory of Groebner bases (essentially, in non-symmetric operads). The reader who is interested in monomial operads only can skip this subsection. |
| The Groebner bases in (shuffle) operads are introduced in [5]; see also [14]. The Groebner bases for non-symmetric operads are discussed in [1]. |
| Fix a discrete set $\Omega$ of generators of a non-symmetric free operad. A nonsymmetric monomial is a multiple composition of operations from $\Omega$. We refer to them simply as monomials. Each monomial is represented by a rooted planar tree with internal vertices labelled by operations. We assume that the edges of the tree lead from the root to the leaves which are free edges. |
| All monomials (including the empty monomial corresponding to the identical operation) form a linear basis of the free non-symmetric operad generated by $\Omega$. Two monomials are called isomorphic if they are isomorphic as labelled trees. A monomial $P$ is divisible by a monomial $Q$ if $Q$ is isomorphic to a submonomial of $P$ where ‘submonomial’ means a labelled subtree. |
| There are families of orderings on the sets of non-symmetric monomials which are compatible with the corresponding compositions. This defines the notion of the leading term of an element of a free operad and leads to the rich Groebner bases theory. The theory includes a version of the Buchberger algorithm [5] and even the triangle lemma [1]. We will call the Groebner basis of the relation ideal of an operad $P$ simply the Groebner basis of $P$. Whereas a general operad could have no finite Groebner basis, a number of important operads (including the operad of associative algebras and their generalized versions) admit such bases. |
| The first known implementation of Groebner base algorithms for an operad is the Haskell package Operads [8]. Its slightly improved version with some bugs fixed by A. Lando can be downloaded at https://github.com/Dronte/Operads . A new Haskell package for operadic Groebner bases (due to Dotsenko and Heijltjes) has been recently published at http://irma.math.unistra.fr/~dot senko/Operads.html . |

#### 2.3 Growth and generating series for operads with finite Groebner bases

The generating series of an operad with a known Groebner basis is equal to the generating series of the corresponding monomial operad, that is, a shuffle operad or a non-symmetric operad whose relations are the leading monomials of the corresponding Groebner basis. The dimension of the $n$-th component of a monomial operad is equal to the number of the monomials of arity $n$ which are not divisible by the monomial relations of the operad. In this section, we consider monomial operads only.

For such an operad, the calculation of the dimensions of its components is a purely combinatorial problem of the enumeration of the labelled trees which do not contain a subtree isomorphic to a relation as a submonomial (a pattern avoidance problem for labelled trees), see [7]. Unfortunately, this problem is too hard to be treated in its full generality. In this section we discuss some partial methods based on the results of [12]. Note that the generating series of general monomial quadratic nonsymmetric operads were first discussed by Parker [16] in other terms.

First, let us discuss a simpler case of non-symmetric operads.
Theorem 2.2 ([12], Th. 2.3.1). The ordinary generating series of a non-symmetric operad with finite Gröbner basis is an algebraic function.

One of the methods for finding the algebraic equation for the generating series of a non-symmetric operad $P$ defined by a finite number of monomial relations $R$ is the following. We consider the monomials (called stamps) of the level less than the maximal level of an element of $R$ which is nonzero in $P$. For each stamp $m = m_i$, we consider the generating function $y_i(z)$ of the set of all nonzero monomials which are left divisible by $m_i$ and are not left divisible by $m_t$ with $t < i$. Then the sum of all $y_i(z)$ is equal to $P(z)$. The divisibility relation on the set of all stamps leads to a system of $N$ equations of the form:

$$y_i = f_i(z, y_1, \ldots, y_N)$$

for each $y_i = y_i(z)$, where $f_i$ is a polynomial and $N$ is the number of all stamps. Note that the degree $d_i$ of the polynomial $f_i$ does not exceed the maximal arity of the generators of the operad $P$. Then the elimination of the variables leads to an algebraic equation of degree at most $d = d_1^2 \ldots d_N^2$ on $P(z)$.

A couple of similar algorithms which in some cases reduce either the number or the degrees of the equations are also discussed in [12]. Knowing an algebraic equation for $P(z)$, one can evaluate the asymptotics for the coefficients $\dim P_n$ by well-known methods [9, Theorem D].

3 THE NON-EXISTENCE OF A GENERAL ALGORITHM

Proposition 3.1. Suppose that the basic field $k$ is computable. Consider the set $H_X$ of non-symmetric quadratic operads $P$ defined by a fixed finite set $X$ of binary generators and some finite set $R$ of quadratic relations on $X$. Then there is a natural $n$ such that $|X| \geq n$ then

(i) the set of Wilf classes of operads from $H_X$ is infinite;
(ii) there does not exist an algorithm which takes as an input two sets $R_1, R_2$ of relations of two operads $P_1, P_2 \in H_X$ which returns TRUE if the operads belong to the same Wilf class and FALSE if not.

Proof. We use Part (ii) of [17, Theorem 3.1]. It states that, under the conditions of Proposition, for some rational function $Q(z)$ there does not exist an algorithm which takes as an input the list $R$ of relations of the operad $P$ such that there is a coefficient-wise inequality $G_p(z) \leq Q(z)$ and returns TRUE if the equality $G_p(z) = Q(z)$ holds and FALSE if not. It follows that the set $H_X$ contains both operads with $G_p(z) = Q(z)$ and operads with $G_p(z) \neq Q(z)$, and there is no algorithm to separate these two subsets. If $R_1$ is an operad of the first kind and $R_2$ is an operad of the second kind, then there is no general algorithm to check whether they belong to the same Wilf class. This proves (ii). Part (i) obliviously follows from (ii).

The rest of our results are positive.

4 OPERADS, TREE PATTERN AVOIDANCE, AND UNAMBIGUOUS CONTEXT-FREE LANGUAGES

In this section, we prove Theorem 1.1. First, let us recall the notation concerning operads and trees.

We consider planar rooted trees with finite possible type of vertices. These types are the following: the root (as a vertex of a special type) and a finite set $X$ of types for the internal vertices and the leaves such that the vertices of the same type have the same number of children. Then the set $X$ is decomposed into the disjoint union $X = X_0 \cup \ldots \cup X_d$ for some $d > 0$, where $X_i$ is the set of the types of vertices with $i$ children (the leaves are assumed to have zero children). We fix some type of leaves $x \in X_0$ and refer to the leaves of type $x$ as free ends. Below we consider trees with no leaves but with free ends, so that we assume that $X_0 = \{x\}$ is a singleton. We call such trees labelled trees or simply trees (with the set of labels $X = X_0 \cup \ldots \cup X_d$).

One can graft (compose) trees by attaching the root of one tree to a free end of another one or replacing free ends with variables. Let us fix a (finite) set $Y$ of labelled trees called patterns. We say that a tree $T$ avoids the pattern set $Y$ if there is no way to obtain $T$ by a subsequent grafting of several trees to each other and at least one of them is a pattern. In other words, a tree does not avoid the patterns if it contains a subtree isomorphic to some element of $Y$. The problem is to enumerate all the trees avoiding the patterns.

Some cases of this problem have been discussed in a number of papers. The case $X = X_0 \cup X_2$, $X_0 = \{x\}$ of binary trees has been considered by Loday [13] and [18] (with $X_2 = \{m\}$). The ternary tree case has been discussed in [10]. The case of quadratic patterns in binary trees has been under consideration in [16]. The general labelled trees case has been considered in [12, Section 2], see also [11].

In Polish notation, each such tree can be encoded by a word on the alphabet $X$. So, all labelled trees avoiding the pattern set $Y$ are in a one-to-one correspondence to some formal language on the alphabet $X$. We denote this language as $L(X|Y)$. The language $L(X|\emptyset)$ is referred to as free and is denoted by $F_X$.

For example, let $X = X_0 \cup X_2$, where $X_2 = \{m_1, \ldots, m_t\}$ and $X_0 = \{x\}$ (where $x$ is a mark for free end). Suppose that $Y = Y_1 \cup Y_2$, where $Y_1 = \{m_i: m_j \ldots m_k | i = 1 \ldots k\}$ and $Y_2$ is some set of trees which are not divisible by the elements of $Y_1$. The last condition means that any right-sided branch cannot have length 2 or more. Then the elements of $Y_2$ should have the form $wx\ldots x$, where $w \in X_2$ is a word on the alphabet $X_2$ and the number of $x$ is length ($w$) + 1. So,

$$L(X|Y) = \{wx^{\text{length}(w)+1} | w \in X_2 \text{ and no subword of } w \text{ belongs to } Y_2 \}.$$

(Note that a word $v$ is called a subword of a word $u$ if $u = avb$ for some words $a$ and $b$.) This means that the words of the language $L(X|Y)$ are in a one-to-one correspondence with the words on $X_2$ which have no subwords lying in $Y_2$, that is, with the monomial basis of the monomial associative algebra $k\langle X_2 \rangle/(Y_2)$.

On the other hand, the words of the free language $L(X|\emptyset)$ with $X$ as above are in a one-to-one correspondence with the generalized Dyck language with $s$ pairs of parentheses.

We will prove the following.

Theorem 4.1. Suppose that the sets $X$ and $Y$ as above are finite. Then the language $L(X|Y)$ is deterministic context-free.

Given an alphabet $X$, one can associate to it a weight function $w : X^* \rightarrow \mathbb{Z}_+^s$ by assigning nonzero weights $w(a) \in \mathbb{Z}_+^s$ to
each letter $a \in X$ and expanding the weight to $X^*$ by the rule $w(\omega) = w(\omega_a)w(o)$. For a language $L$ on $X$, one can consider the generating function
\[
H_L(z) = \sum_{\alpha \in \mathbb{Z}_+} z^{|w(\alpha)|},
\]
where $z = (z_1, \ldots, z_d)$ and $z^{(n_1 \ldots n_d)} = z_1^{n_1} \cdots z_d^{n_d}$. For example, in the case $w(x_1) = \cdots = w(x_d) = 1 \in \mathbb{Z}_+$ the formal power series $H_L(z_1)$ is the generating function for the growth function $g_L(n) = \#(u \in L | \text{length}(u) = n)$ of the language $L$.

The famous enumeration theorem by Chomsky and Schützenberger [2] describes growth functions of unambiguous context-free languages. Using this and the theorem by D’Alessandro, Intrigila, and Varriacchio about generating function of sparse context free languages [3], we get

**Corollary 4.2.** Let $H(z)$ be the generating function of the language $L(X|Y)$ above, where the sets $X$ and $Y$ are finite and $z = (z_1, \ldots, z_d)$ is a vector of variables. Then the formal power series $H(z)$ satisfies a non-trivial algebraic equation with coefficients in $\mathbb{Z}[z]$. If, moreover, the growth of the language is sub-exponential, then the function $H(z)$ is rational.

The Chomsky–Schützenberger theorem gives a way to construct a system of algebraic equations for $H(z)$. Its variables are the generating functions of the sublanguages of $L(X|Y)$ which can be derived from the non-terminals of the unambiguous context-free grammar. If we use the grammar $G$ for the language $L(X|Y)$ (see Lemma 4.5), we get a system equivalent to the one described in Subsection 2.3 and [12, 2.2.1]. Moreover, after a triangular linear change of variables, it is also equivalent to another system of equations described in [12, 2.2.2]. In the case of binary one-relator operads, these two kinds of systems were created earlier by Rowland (see [18]; we discuss this case in Section 5 below).

**Corollary 4.3.** Let $P$ be a finitely generated non-symmetric operad having a finite Groebner basis of relations. Then its natural monomial basis forms a deterministic context-free language $L(X|Y)$ for some finite $X$ and $Y$. In particular, the generating series of the operad satisfy the conclusion of Corollary 4.2.

Given a tree $t \in P_X$, its height $ht(t)$ is the maximal number of internal nodes lying on the same branch. Let $d = \max \{ht(t) | t \in Y\}$ be the maximal pattern height. In the case of empty $Y$, we put $d = 0$.

We say that a tree $t$ is a rooted subtree of a tree $\nu$ (notation: $t \leq \nu$) if $t$ can be obtained from $\nu$ by grafting some other trees onto it. If, in addition, $t \neq \nu$, we write $r < t$.

In the notation of Theorem 4.1, let us denote $L = L(X|Y)$ and $L' = L'(X|Y)$. For $n \geq 0$, let $L_n$ (resp., $L'_n$) denote the set of all trees in $L$ (resp., $L'$) having a height of at most $n$.

Let $t \in L$. Let $\bar{M}_t = \{v \in L | t \leq v\}$ denote the set of all the trees of $L$ obtained from $t$ by grafting other trees onto it. Put
\[
M_t = \bar{M}_t \setminus \bigcup_{s \in L, t < s} \bar{M}_s.
\]

**Lemma 4.4.** The language $L = L(X|Y)$ is the disjoint union of the subsets $M_t$ with $t \in L_d$.

**Proof.** Obviously, the union of all such sets $M_t$ is the same as the union of all sets $\bar{M}_t$, where $t$ runs $L_d$. Since $\bar{M}_1 = L$ (where 1 is the tree consisting of the root and single free end), this union is equal to $L$. Let us prove that the union is disjoint.

Ad absurdum, suppose that for some different $s, t \in L_d$ there exists a tree $p \in M_s \cap M_t$. Since $p \in \bar{M}_s \cap \bar{M}_t$, the both trees $s$ and $t$ are rooted subtrees of $p$. So, there is the minimal (w. r. t. the relation $<$) rooted subtree $r$ of $p$ such that both $s$ and $t$ are rooted subtrees of $r$. The set of internal nodes of $r$ (as a subgraph of $p$) is the union of the sets of internal nodes of $s$ and $t$, so that $ht r \leq \max(ht s, ht t) \leq d$. Since $r$ is a rooted subtree of $p \in L$, it follows that $r \in L$, so that $r \in L_d$. Then $p \in \bar{M}_r$. If $t \neq p$ (or, respectively, $s \neq p$), then
\[
p \in M_t \subseteq \bar{M}_r \setminus \bar{M}_t
\]
(resp., $p \in \bar{M}_s \setminus \bar{M}_t$), in contradiction to the condition $p \in \bar{M}_r$. So, $p = s = t$: this contradicts the choice of $s$ and $t$. □

Now, let us define a context-free grammar $G$ for the languages $L$ as follows. Let the sets of terminal symbols be $X$, and let $V = \{T_v | v \in L_d\} \cup \{S\}$ be the set of non-terminal symbols. The sets of rules of these grammars are the following. First, for each $v \in L_d$, let $m = m_v$ be the label of the root vertex of $v$, and let $k$ be the number of children of this vertex (so that $m \in X_k$). If $k \geq 1$ then the rule
\[
T_v \rightarrow mT_{v_1} \cdots T_{v_k}
\]
exists for some $v_1, \ldots, v_k \in L_d$ iff $mv_1 \cdots v_k \in M_v$. Next, there is a rule
\[
S \rightarrow x.
\]
The initial rules are
\[
S \rightarrow T_v
\]
for all $v \in L_d$.

**Lemma 4.5.** The grammar $G$ is unambiguous and generates the languages $L = L(X|Y)$.

**Proof.** Let us prove that for each word $w$ from $L$ there exists a unique rightmost derivation in the grammar $G$.

Let us first show that for each such word $w \in M_v$ there is a rightmost derivation beginning with $S \rightarrow T_v$. Indeed, if $w = x$, then there is unique derivation $S \rightarrow T_v \rightarrow x$. Otherwise, there are unique $v, v_1, \ldots, v_k$, and $m$ such that $w = mv_1 \cdots v_k \in M_v$. By the induction argument, for each $v_i$ there is a unique rightmost derivation. It has the form
\[
S \rightarrow T_{v_i} \rightarrow \cdots \rightarrow v_i,
\]
where $v_i \in M_{h_i}$. Then there is a rightmost derivation
\[
S \rightarrow T_v \rightarrow mT_{h_1} \cdots T_{h_{k-1}}T_{h_k} \rightarrow \cdots \rightarrow mT_{h_1} \cdots T_{h_{k-1}}v_k
\rightarrow \cdots \rightarrow mv_1 \cdots v_k = w. \tag{4.1}
\]

Let us show that this derivation is unique. By the induction argument, it is sufficient to show that the first step is unique. Moreover, we can assume by induction that the unique rightmost derivation for any word $b$ of length less than the one $w$ begins with $S \rightarrow T_c$ where $c \in M_c$.

So, assume that there is a rightmost derivation $S \rightarrow \overrightarrow{w}$ with the initial step $S \rightarrow T_c$; we need to show that $v = v'$. The next step of the derivation must be of the form $T_c \rightarrow m'T_{c'} \cdots T_{c'}$.}

```
where $m'\omega'_1 \ldots \omega'_{k'} \in M_{\omega'}$. It follows that $w = m'\omega'_1 \ldots w'_k$, where for each $i$ there exists a (rightmost) derivation $T'_{\omega'_i} \rightarrow w'_i$. Here $m'$ corresponds to the mark of the root of the tree $w$ and the subwords $w'_1, \ldots, w'_k$ correspond to its branches. It follows that $m' = m$, $k' = k$, and $w'_i = w_i$ for each $i = 1, \ldots, k$. Moreover, since for each $w_i$ there are two rightmost derivations

$$S \rightarrow T_{\omega_i} \rightarrow w'_i \equiv w_i$$

and

$$S \rightarrow T_{\omega_i} \rightarrow \omega'_i,$$

we use the induction assumption to conclude that $\omega'_i = \omega_i$. Thus, the word $m\omega_1 \ldots \omega_k = m'\omega'_1 \ldots \omega'_{k'}$ belongs to both $M_\omega$ and $M_{\omega'}$, so that $\omega = \omega'$.

**Lemma 4.6.** The grammar $G$ is deterministic.

**Proof.** We have to show that each word $a$ appearing in the derivation process for some $w \in L$ (respectively, $w \in L'$) has a forced handle. If $w \in X$, then both words appearing in the derivation $S \rightarrow T_w \rightarrow w$ obviously coincide with their one-symbol handles, so that the handles are forced.

It remains to show that each word $a$ appearing in the derivation (4.1) for a word $w$ of length at least 2 has a forced handle. Let $r$ be the maximal prefix of $a$ which ends with a non-terminal, so that $a = rs$ where $r \in X \cup V^* \cup \{m\}$ and $s \in X^*$ consists of variables.

On the other side, the word $a$ has the form $a = uhq$, where $u, h, q \in (X \cup V)^*$, $h$ is the handle, and $q \in X^*$ is a word consisting of terminal symbols. The handle $h$ of $a$ is the right-hand-side of some grammar rule, so that it has either the form $T_h$ for some $v \in L_d$, or the form $\mu T_{a_1} \ldots T_{a_t}$ for some $\mu, a_1, \ldots, a_t$, or is equal to $x$.

Let us compare both decompositions $rs$ and $ahq$ of the same word $w$. Obviously, if $h = T_v$ for some $v \neq x$, then $a = T_v$ and this is the first step $S \rightarrow T_v$ of the derivation. Still, if $a = T_v$ for some $v \in L$, then $h$ is forced, because no word but a derived from $S$ can begin with $T_v$. We will not consider this case further. Next, if the handle $h$ is of the form $\mu T_{a_1} \ldots T_{a_t}$, then it consists of the rightmost sequence of non-terminals in $uhq$ accompanied with the preceding terminal symbol, so that $r = uh$ and $s = q$. Finally, if $h = x$, then $s = s_1xq$, where $s_1$ does not contain $x$, so that $s_1 \in (X \setminus X_0)^*.$

Let $t \in [1, k]$ be the maximal number such that $a$ appears before the $(k + 1)$-symbol initial segment of the word $mT_{h_1} \ldots T_{h_k}$ in (4.1) is changed, so that the derivation of $w$ splits as

$$S \rightarrow T_0 \rightarrow mT_{h_1} \ldots T_{h_k} \rightarrow \ldots \rightarrow mT_{h_1} \ldots T_{h_{t-1}}u_i \ldots u_k$$

$$\rightarrow \ldots \rightarrow a \rightarrow \ldots \rightarrow mT_{h_1} \ldots T_{h_{t-1}}u_i u_{t+1} \ldots u_k$$

$$\rightarrow \ldots \rightarrow mT_{h_1} \ldots T_{h_{t-1}}u_i u_{t+1} \ldots u_k$$

$$\rightarrow \ldots \rightarrow mT_{h_1} \ldots T_{h_{t-1}}u_i u_{t+1} \ldots u_k$$

$$\rightarrow \ldots \rightarrow a \rightarrow \ldots \rightarrow mT_{h_1} \ldots T_{h_{t-1}}u_i u_{t+1} \ldots u_k$$

$$\rightarrow \ldots \rightarrow mT_{h_1} \ldots T_{h_{t-1}}u_i u_{t+1} \ldots u_k$$

$$\rightarrow \ldots \rightarrow mT_{h_1} \ldots T_{h_{t-1}}u_i u_{t+1} \ldots u_k$$

Then $a$ has the form $a = mT_{h_1} \ldots T_{h_{t-1}}a'$ with $a' = a''u_{t+1} \ldots u_k$, where the word $a''$ appears in the derivation

$$S \rightarrow T_{h_t} \rightarrow \ldots \rightarrow a'' \rightarrow \ldots \rightarrow a_i$$

(4.2) of the word $w_i$.

Ad absurdum, suppose that the rightmost derivation of some word $\tilde{w} \in L$ (or $\tilde{w} \in L'$, in the case of grammar $G'$) contains a word $\tilde{a} = uh\tilde{q}$ (where $\tilde{q} \in X^*$) which has another handle $\tilde{h}$. If both handles $h$ and $\tilde{h}$ are of the form $\mu T_{a_1} \ldots T_{a_t}$, then each of them is uniquely defined as the rightmost subword of $w$ having this form, so that $\tilde{h} = h$, a contradiction. Now, suppose that both handles $h$ and $\tilde{h}$ are single symbols $x$. Then the handles are uniquely defined as the leftmost occurrences in the words $uh\tilde{q}$ and, respectively, $uh\tilde{q}$. Then we again get $h = \tilde{h}$, a contradiction.

Now, it remains to consider the case when both handles are of different kinds. We can assume that $h = \mu T_{a_1} \ldots T_{a_t}$ and $\tilde{h} = x$, so that $a = uhs$ and $\tilde{a} = uh\tilde{q}$. Then $h$ is the handle of the word $a' = uhx'$ appearing in the derivation (4.2) of the word $w_i$, where $s = s'v_{t+1} \ldots v_k$. On the other hand, since the initial symbol of the word $\tilde{w}$ is $m$, then $\tilde{w} = m\tilde{v}_1 \ldots \tilde{v}_k$ for some $\tilde{v}_1, \ldots, \tilde{v}_k$. Then the derivation of $\tilde{w}$ has the following form similar to (4.1)

$$S \rightarrow T_0 \rightarrow mT_{h_{k-1}} \ldots T_{h_1} \rightarrow \ldots$$

$$\rightarrow mT_{h_{k-1}} \ldots T_{h_{k-1}} \tilde{v}_k \rightarrow \ldots \rightarrow mT_{h_{k-1}} \ldots T_{h_{k-1}} \tilde{v}_{t+1} \ldots \tilde{v}_k$$

$$\rightarrow \ldots \rightarrow \tilde{a} \rightarrow \ldots \rightarrow mT_{h_{k-1}} \ldots T_{h_{k-1}} \tilde{v}_{t+1} \ldots \tilde{v}_k$$

$$\rightarrow \ldots \rightarrow m\tilde{v}_1 \ldots \tilde{v}_k = \tilde{w}.$$
Two patterns are enumerating equivalent if their enumerating generating function of Wilf equivalence. Wikipedia of Integer Sequences as A161746, see https://oeis.org/A161746. the calculations for $n$ calculated up to the number conjecture.

Since the $k$-th component $P_k$ of the operad is spanned by trees with $k$ leaves, the generation function of the operad is

$$G_{P}(z) = \sum_{n \geq 1} a_n z^{(n-1)/2} = A_{\mathbb{F}}(\sqrt{z})/\sqrt{z}.$$ 

The Wilf classes of these operads correspond to the Wilf classes of the patterns.

Now, let $a_{n,k}$ be the number of binary trees with $n$ vertices which contains exactly $k$ copies of pattern $t$. The enumerating generating function of $t$ is

$$E_{\mathbb{F}}(x, y) = \sum_{n \geq 1, k \geq 0} a_{n,k} x^n y^k.$$ 

Two patterns are enumerating equivalent if their enumerating generating functions are equal. As $A_{\mathbb{F}}(x) = E_{\mathbb{F}}(x, 0)$, this is a stronger version of Wilf equivalence.

The next conjecture states that the strong and weak version of Wilf classes should coincide.

**Conjecture 5.1 (Rowland).** If two patterns $s$ and $t$ are Wilf equivalent, then they are enumerating equivalent.

In the next section, we discuss a homological interpretation of the enumerating generating function and the above Rowland conjecture.

In [18], the algebraic equations defining the functions $A_{\mathbb{F}}(x)$ and $E_{\mathbb{F}}(x, y)$ are listed for all patterns with at most 6 leaves. Moreover, the number $A(n)$ of Wilf classes of the $n$-leaves patterns are calculated up to $n = 8$. These numbers are 1, 1, 2, 3, 7, 15, 44. Moreover, the calculations for $n \leq 7$ confirm Conjecture 5.1.

Note that the sequence $A(n)$ is listed in Sloane’s On-Line Encyclopedia of Integer Sequences as A161746, see https://oeis.org/A161746.

Now, we have tried to take the next step toward the enumeration of Wilf classes of one-relator binary operads.

**Table 1: Lower bounds for the numbers Wilf classes of one-relator quadratic operads (the generating functions are calculated up to $o(x^k)$)**

| $n$ | 8   | 9   | 10  | 11  | 12  |
|-----|-----|-----|-----|-----|-----|
| $A(n)$ | 43  | 136 | 458 | 1662| 6096|
| $k$ | 257 | 257 | 257 | 201 | 201 | 157 |
| $E(n)$ | 43  | 136 | 458 | 1662| 6096|
| $k$ | 257 | 257 | 257 | 201 | 201 | 157 |

First, we have provided a calculation similar to Rowland’s. We have used the methods from [18] to construct the systems of algebraic equations. The systems of equations were generated by an ad hoc C# software. We then proceeded with the Wolfram Mathematica elimination of variables procedure. As both Groebner bases methods and computer performance have been improved within the last decade, we hoped to calculate the number $A(n)$ of Wilf classes $A(n)$ and the number $E(n)$ of enumeration classes for the next values of $n$. However, we have done no more than confirm Rowland’s calculation, that is, to list 15 equations for $En(x, y)$ for $n = 7$ and 44 equations for $A0(x)$ for $n = 8$.

The systems of algebraic equations over the polynomials give recurrent equations for the coefficients of the power series solutions. As our systems are of a rather special kind, these recurrent equations are rather simple in our case. Following this, we developed a Python package (called Friend reduce, as we have parallelized the calculations among our friends’ laptops) which, for each of our systems of algebraic equations, finds its solution in the form of the collection of truncated formal power series. Given a number $n$ of the leaves, for each pattern $t$ with $n$ leaves we have, particularly, calculated all formal power series $A_{\mathbb{F}}(x)$ and $E_{\mathbb{F}}(x, y)$ truncated up to some power $x^k$. The numbers $A(n, k)$ and $E(n, k)$ of such truncated series give lower bounds for the numbers $A(n)$ and $E(n)$.

Both packages can be downloaded at https://github.com/atcherkasov/tree-Wilf-classes.

The results of the calculations are presented in Table 1. 

Note that for $n = 8$, we give the exact value $A(8) = 43$ in place of the lower bound. Moreover, this equality contradicts the value 44 of $A(8)$ listed in [18, p. 756]. Nevertheless, we are sure of this equality for the following reasons. After the elimination of variables, we get 44 different equations for $A_r(x)$ with 8-leave patterns $t$. Still, one can show that two of these equations define the same algebraic power series. These are the equations

\[
\begin{align*}
&(-x + x^3 - x^5 - x^7 + x^9 + x^{13}) \\
+ G \cdot (1 - 2x^2 + 4x^4 + 2x^6 - 6x^8 + 2x^{10} - 4x^{12} + 3x^{14}) \\
+ G^2 \cdot (-3x^3 + 9x^7 - 8x^9 + 7x^{11} - 8x^{13} + 3x^{15}) \\
+ G^3 \cdot (-3x^6 + 7x^8 - 8x^{10} + 6x^{12} - 3x^{14} + x^{16}) \\
+ G^4 \cdot (3x^9 - 2x^{11} - 3x^{13} + 2x^{15}) + G^5 \cdot (2x^{12} - 3x^{14} + x^{16}) = 0
\end{align*}
\] (5.1)
The purpose of this section is to discuss a direction for future research. We believe that this research would give methods and algorithms to study refined versions of the Wilf classes based on homological invariants.

In [6] Dotsenko and Khoroshkin have introduced a differential graded resolution of a shuffle monomial operad $P$ (moreover, the construction is generalized to the case of a general operad using Groebner bases). The version of this construction for non-symmetric operads is described in [12, 2.2.3]. It is a free operad generated by the generators of $P$ and additional trees that are in one-to-one correspondence with the monomials covered by the monomial relations of $P$ in an “indecomposable” way. Let $b_{k,n}$ denote the number of the degree $n$ generators containing exactly $k$ copies of the relations. For some particular monomial operads (such as the quadratic ones and the operads defined by the patterns of Class 4.2 from [18]), the above resolution is minimal, so that the numbers $b_{k,n}$ are equal to the Betti numbers $β_{k,n}$ of $P$ in the sense of Quillen homology (that is, the number of the minimal generators of internal degree $n$ and homological degree $k$ in the minimal differential graded model of $P$). In general, we have useful inequalities $b_{k,n} ≥ β_{k,n}$.

Let $B_P(z,y) = \sum_{k,n} b_{k,n} z^k y^n$ be the generating function. One can consider it as a version of the Poincare series of $P$. One can consider “homological” Wilf classes of the operads by saying that two operads are homologically equivalent if their functions $B_P(z,y)$ coincide.

**Proposition 6.1.** The functions $B_P(z,y)$ and $E_n(x,y)$ can be uniquely defined in terms of each other. Therefore, two one-relator binary operads belong to the same homological Wilf class if and only if their relations are enumeration equivalent.

Then Rowland’s Conjecture 5.1 implies

**Conjecture 6.2.** In the class of one-relator binary operads $P$, the Betti numbers $b_{n,k}$ are uniquely defined by the generating series $G_P(z)$.

The calculations described in Section 5 confirm this conjecture provided that the degree of the operad relation is less or equal to 12. If the conjecture turns out to be true, it will be interesting to get an answer to the question: are Quillen Betti numbers $b_{k,n}$ of one-relator binary operads also determined by the Wilf class?

For the direct calculation of Quillen homology using the irreducible elements of this resolution, one can use the simplicial homology (cf. chain complexes of simplices mentioned in Subsection 2.1 of [6]). It is our future plan to apply computational topology software for evaluating the Poincare series of operads and to provide a stronger homological classification.

**REFERENCES**

[1] M. Bremner and V. Dotsenko. 2016. Algebraic operads: an algorithmic companion. CRC Press.

[2] Noam Chomsky and Marcel P. Schützenberger. 1963. The algebraic theory of context-free languages. In Studies in Logic and the Foundations of Mathematics. Vol. 26. Elsevier, 118–161.

[3] Flavio D’Alessandro, Benedetto Intrigila, and Stefano Varricchio. 2006. On the structure of the counting function of sparse context-free languages. Theoretical computer science 356, 1-2 (2006), 104–117.

[4] V. Dotsenko. 2012. Pattern avoidance in labelled trees. Séminaire Lotharingien de Combinatoire 67 (2012). 6B76.

[5] V. Dotsenko and A. Khoroshkin. 2010. Gröbner bases for operads. Duke Math. J. 153, 2 (2010), 363–396.

[6] Vladimir Dotsenko and Anton Khoroshkin. 2013. Quillen Homology for Operads via Gröbner Bases. Documenta Mathematica 18 (2013), 707–747.

[7] Y. Dotsenko and A. Khoroshkin. 2015. Shuffle algebras, homology, and consecutive pattern avoidance. Algebra and Number Theory 7 (2013), 673–700.

[8] V. Dotsenko and M. Vejdemo-Johansson. 2010. Operadic Groebner bases: an implementation. In International Congress on Mathematical Software. Springer, 249–252.

[9] Ph. Flajolet. 1987. Analytic models and ambiguity of context-free languages. Theoret. Comput. Sci. 49, 2–3 (1987), 283–309.

[10] Nathan Gabriel, Katherine Peske, Lara Pavlovic, and Samuel Tay. 2012. Pattern avoidance in ternary trees. Journal of Integer Sequences 15, 2 (2012), 3.

[11] Samuele Giraudi. 2020. Tree series and pattern avoidance in syntax trees. Journal of Combinatorial Theory, Series A 176 (2020), 105285.

[12] A. Khoroshkin and D. Piontkovski. 2015. On generating series of finitely presented operads. Journal of Algebra 426 (2015), 377–429.

[13] Jean-Louis Loday. 2005. Inversion of integral series enumerating planar trees. Séminaire Lotharingien de Combinatoire 53 (2005), B53d.

[14] J. C. Lagardère and B. Vallette. 2012. Algebraic Operads. Springer, Heidelberg. Grundl. Math. Wiss. 346.

[15] M. Markl, S. Shnider, and J. Stasheff. 2012. Operads and Algebraic Geometry. Mathematical Surveys and Monographs, Vol. 96. AMS, Providence, RI.

[16] Susan Field Parker. 1993. The combinatorics of functional composition and inversion. Ph.D. Dissertation. Brandeis University.

[17] Dmitri Piontkovski. 2017. Growth in varieties of multioperator algebras and Groebner bases in operads. In ISSAC ’17: Proceedings of the 2017 ACM on International Symposium on Symbolic and Algebraic Computation, 373–380.

[18] Eric S Rowland. 2010. Pattern avoidance in binary trees. Journal of Combinatorial Theory, Series A 117, 6 (2010), 741–758.