Weighted Basic Parallel Processes and Combinatorial Enumeration

Lorenzo Clemente
Department of Mathematics, Mechanics, and Computer Science, University of Warsaw, Poland

Abstract

We study *weighted basic parallel processes* (WBPP), a nonlinear recursive generalisation of weighted finite automata inspired from process algebra and Petri net theory. Our main result is an algorithm of 2-EXPSPACE complexity for the WBPP equivalence problem. While (unweighted) BPP language equivalence is undecidable, we can use this algorithm to decide multiplicity equivalence of BPP and language equivalence of unambiguous BPP, with the same complexity. These are long-standing open problems for the related model of weighted context-free grammars.

Our second contribution is a connection between WBPP, power series solutions of systems of polynomial differential equations, and combinatorial enumeration. To this end we consider *constructible differentially finite* power series (CDF), a class of multivariate differentially algebraic series introduced by Bergeron and Reutenauer in order to provide a combinatorial interpretation to differential equations. CDF series generalise rational, algebraic, and a large class of D-finite (holonomic) series, for which no complexity upper bound for equivalence was known. We show that CDF series correspond to *commutative* WBPP series. As a consequence of our result on WBPP and commutativity, we show that equivalence of CDF power series can be decided with 2-EXPTIME complexity.

In order to showcase the CDF equivalence algorithm, we show that CDF power series naturally arise from combinatorial enumeration, namely as the exponential generating series of *constructible species of structures*. Examples of such species include sequences, binary trees, ordered trees, Cayley trees, set partitions, series-parallel graphs, and many others. As a consequence of this connection, we obtain an algorithm to decide multiplicity equivalence of constructible species, decidability of which was not known before.

The complexity analysis is based on effective bounds from algebraic geometry, namely on the length of chains of polynomial ideals constructed by repeated application of finitely many, not necessarily commuting derivations of a multivariate polynomial ring. This is obtained by generalising a result of Novikov and Yakovenko in the case of a single derivation, which is noteworthy since generic bounds on ideal chains are non-primitive recursive in general. On the way, we develop the theory of WBPP series and CDF power series, exposing several of their appealing properties.

2012 ACM Subject Classification Theory of computation → Quantitative automata; Theory of computation → Concurrency; Mathematics of computing → Combinatorics

Keywords and phrases weighted automata, combinatorial enumeration, shuffle, algebraic differential equations, process algebra, basic parallel processes, species of structures

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2024.18

Related Version *Full Version*: https://arxiv.org/abs/2407.03638 [24]

Funding Supported by the ERC grant INFSYS, agreement no. 950398.

Acknowledgements We warmly thank Mikołaj Bojańczyk, Arka Ghosh, Filip Mazowiecki, and Paweł Parys for their comments and support at the various stages of this work.

35th International Conference on Concurrency Theory (CONCUR 2024).
Editors: Rupak Majumdar and Alexandra Silva; Article No. 18; pp. 18:1–18:22
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

We study the equivalence problem for a class of finitely presented seriesoriginating in weighted automata, process algebra, and combinatorics. We begin with some background.

1.1 Motivation and context

Weighted automata. Classical models of computation arising in the seminal work of Turing from the 1930's [75] have a Boolean-valued semantics ("is an input accepted?") and naturally recognise languages of finite words $L \subseteq \Sigma^*$. In the 1950's a finite-memory restriction was imposed on Turing machines, leading to an elegant and robust theory of *finite automata* [64], with fruitful connections with logic [18, 28, 74] and regular expressions [47]. *Weighted finite automata* over a field $\mathbb{F}$ (WFA) [68] were introduced in the 1960's by Schützenberger as a generalisation of finite automata to a quantitative *series* semantics $\Sigma^* \to \mathbb{F}$ ("in how many ways can an input be accepted?"). This has been followed by the development of a rich theory of weighted automata and logics [27]. While the general theory can be developed over arbitrary semirings, the methods that we develop in this work are specific to fields, and in particular for effectiveness we assume the field of rational numbers $\mathbb{F} = \mathbb{Q}$.

The central algorithmic question that we study is the *equivalence problem*: Given two (finitely presented) series $f, g : \Sigma^* \to \mathbb{Q}$, is it the case that $f = g$? (In algorithmic group theory this is known as the *word problem.*) A mathematical characterisation of equivalence yields a deeper understanding of the interplay between syntax and semantics, and a decidability result means that this understanding is even encoded as an algorithm. Equivalence of weighted models generalises *multiplicity equivalence* of their unweighted counterparts ("do two models accept each input in the same number of ways?"), in turn generalising language equivalence of *unambiguous models* (each input is accepted with multiplicity 0 or 1). Since equivalence $f = g$ reduces to *zeroness* $f - g = 0$, from now on we will focus on the latter problem.

While nonemptiness of WFA is undecidable [58, Theorem 21] (later reported in Paz' book [60, Theorem 6.17]), zeroness is decidable, even in polynomial time [68] – a fact often rediscovered, e.g., [73, 76]. This has motivated the search for generalisations of WFA with decidable zeroness. However, many of them are either known to be undecidable (e.g., *weighted Petri nets* [42, Theorem 3]), or beyond the reach of current techniques (e.g., *weighted one counter automata*, *weighted context-free grammars*, and *weighted Parikh automata*). One notable exception is *polynomial weighted automata*, although zeroness has very high complexity (Ackermann-complete) [3]. In the restricted case of a unary input alphabet, decidability and complexity results can be obtained with algebraic [1] and D-finite techniques [15].

Process algebra. On a parallel line of research, the process algebra community has developed a variety of formalisms modelling different aspects of concurrency and nondeterminism. We focus on *basic parallel processes* (BPP) [21], a subset of the *calculus of communicating systems* without sequential composition [55]. BPP are also known as *communication-free Petri nets* (every transition consumes exactly one token) and *commutative context-free grammars* (nonterminals in sentential forms are allowed to commute with each other). While language equality for BPP is undecidable [40, 41], bisimulation equivalence is decidable [22] (even PSPACE-complete [70, 43]). *Multiplicity equivalence*, finer than language equality and incomparable with bisimulation, does not seem to have been studied for BPP.

Combinatorial enumeration and power series. We shall make a connection between BPP, power series, and combinatorial enumeration. For this purpose, let us recall that the study of multivariate power series in commuting variables has a long tradition at the border of combinatorics, algebra, and analysis of algorithms [72, 30]. We focus on *constructible differentially finite* power series (CDF) [6, 7], a class of differentially algebraic power series arising in combinatorial enumeration [49, 4]. Their study was initiated in the *univariate* context in [6], later extended to multivariate [7]. They generalise rational and algebraic power series, and are incomparable with D-finite power series [71, 51]. For instance, the exponential generating series $\sum_{n \in \mathbb{N}} n^{n-1} \cdot x^n / n!$ of Cayley trees is CDF, but it is neither algebraic/D-finite [16, Theorem 1], nor polynomial recursive [20, Theorem 5.3].

The theory of *combinatorial species* [45, 5] is a formalism describing families of finite structures. It arises as a categorification of power series, by noticing how primitives used to build structures – sum, combinatorial product, composition, differentiation, resolution of implicit equations – are in a one-to-one correspondence with corresponding primitives on series. Using these primitives, a rich class of *constructible species* can be defined [61]. For instance the species $\mathsf{C}[\mathcal{X}]$ of Cayley trees (rooted unordered trees) is constructible since it satisfies $\mathsf{C}[\mathcal{X}] = \mathcal{X} \cdot \mathsf{SET}[\mathsf{C}[\mathcal{X}]]$. Two species are *multiplicity equivalent* (*equipotent* [61]) if for every $n \in \mathbb{N}$ they have the same number of structures of size n. Multiplicity equivalence of species has not been studied from an algorithmic point of view.

1.2 Contributions

We study *weighted basic parallel processes* over the field of rational numbers (WBPP), a weighted extension of BPP generalising WFA. The following is our main contribution.

▶ **Theorem 1.** *The zeroness problem for WBPP is in 2-EXPSPACE.*

This elementary complexity should be contrasted with Ackermann-hardness of zeroness of polynomial automata [3], another incomparable extension of WFA. Since WBPP can model the multiplicity semantics of BPP, as an application we get the following corollary.

▶ **Corollary 2.** *Multiplicity equivalence of BPP and language equivalence of unambiguous BPP are decidable in 2-EXPSPACE.*

On a technical level, Theorem 1 is obtained by extending an ideal construction and complexity analysis from [59] from the case of a single polynomial derivation to the case of a finite set of not necessarily commuting polynomial derivations. It is remarkable that such ideal chains have elementary length, since generic bounds without further structural restrictions are only general recursive [69]. This shows that the BPP semantics is adequately captured by differential algebra. These results are presented in § 2. In § 3 we observe that *commutative* WBPP series coincide with CDF power series, thus establishing a novel connection between automata theory, polynomial differential equations, and combinatorics. This allows us to obtain a zeroness algorithm for CDF, which is our second main contribution.

▶ **Theorem 3.** *The zeroness problem for multivariate CDF power series is in 2-EXPTIME.*

The complexity improvement from 2-EXPSPACE to 2-EXPTIME is due to commutativity. In the special *univariate* case, decidability was observed in [6] with no complexity analysis, while [7] did not discuss decidability in the multivariate case. In § 4 we apply Theorem 3 to multiplicity equivalence of a class of constructible species. This follows from the observation that their exponential generating series (EGS) are effectively CDF, proved by an inductive argument based on the closure properties of CDF series. For instance, the EGS of Cayley trees satisfies $C = x \cdot e^C$; by introducing auxiliary series $D := e^C, E := (1 - C)^{-1}$ and by differentiating we obtain CDF equations $\partial_x C = D \cdot E$, $\partial_x D = D^2 \cdot E$, $\partial_x E = D \cdot E^3$.

▶ **Theorem 4.** *Multiplicity equivalence of strongly constructible species is decidable.*

1.3 Related works

There have recently been many decidability results for models incomparable with WBPP, such as multiplicity equivalence of *boundedly-ambiguous* Petri nets [26, Theorem 3]; zeroness for weighted one-counter automata with *deterministic counter updates* [52]; zeroness of *P-finite automata*, a model intermediate between WFA and polynomial automata (even in PTIME [19]); and zeroness of *orbit-finite weighted automata* in sets with atoms [9].

Regarding power series, there is a rich literature on dynamical systems satisfying differential equations in the CDF format, that is polynomial ordinary differential equations (ODE; cf. [62] and references therein). While many algorithms have been proposed for their analysis (e.g., invariant checking [63]), the complexity of the zeroness problem has not been addressed before. A decision procedure for zeroness of multivariate CDF can be obtained from first principles as a consequence of Hilbert's *finite basis theorem* [25, Theorem 4, §5, Ch. 2]. For instance, decidability follows from the algorithm of [12] computing pre- and post-conditions for restricted systems of partial differential equations (covering CDF), and also from the *Rosenfeld–Gröbner algorithm* [17], which can be used to test membership in the radical differential ideal generated by the system of CDF equations. In both cases, no complexity-theoretic analysis is provided and only decidability can be deduced. In the univariate CDF case, decidability can also be deduced from [10, 11]. Univariate CDF also arise in the coalgebraic treatment of stream equations with the shuffle product [14, 13], where an equivalence algorithm based on Hilbert's theorem is provided.

The work [34] studies *Noetherian functions*, which are analytic functions satisfying CDF equations. In fact, Noetherian functions which are analytic around the origin coincide with multivariate CDF power series. The work [77] discusses a subclass of Noetherian functions obtained by iteratively applying certain extensions to the ring of multivariate polynomials and presents a zeroness algorithm running in doubly exponential time. Theorem 3 is more general since it applies to all Noetherian power series.

In the context of the realisability problem in control theory, Fliess has introduced the class of *differentially producible series* [32] (cf. also the exposition of Reutenauer [66]), a generalisation of WBPP series where the state and transitions are given by arbitrary power series (instead of polynomials). Such series are characterised by a notion of *finite Lie rank* and it is shown that differentially producible series of minimal Lie rank exist and are unique. Such series are not finitely presented and thus algorithmic problems, such as equivalence, cannot even be formulated.

Full proofs can be found in the technical report [24].

Preliminaries. Let $\Sigma = \{a_1, \ldots, a_d\}$ be a finite alphabet. We denote by Σ^* the set of *finite words* over Σ, a monoid under the operation of concatenation, with neutral element the empty word ε. The *Parikh image* of a word $w \in \Sigma^*$ is $\#(w) := (\#(w)_{a_1}, \ldots, \#(w)_{a_d}) \in \mathbb{N}^d$, where $\#(w)_{a_j}$ is the number of occurrences of a_j in w. Let $\mathbb{Q}$ be the field of rational numbers. Most results in the paper hold for any field, however for computability considerations we restrict our presentation to $\mathbb{Q}$. For a tuple of commuting indeterminates $x = (x_1, \ldots, x_k)$, denote by $\mathbb{Q}[x]$ the ring of *multivariate polynomials* ($\mathbb{Q}[k]$ when the name of variables does not matter) and by $\mathbb{Q}(x)$ its fraction field of *rational functions* (that is, ratios of polynomials $p(x)/q(x)$). The *one norm* $|z|_1$ of a vector $z = (z_1, \ldots, z_k) \in \mathbb{Q}^k$ is $|z_1| + \cdots + |z_k|$, and the *infinity norm* is $|z|_\infty = \max_{1 \leq i \leq k} |z_k|$. Similarly, the *infinity norm* (also called *height*) of a polynomial $p \in \mathbb{Q}[k]$, written $|p|_\infty$, is the maximal absolute value of any of its coefficients.

A *derivation* of a ring R is a linear function $\delta : R \to R$ satisfying

$$\delta(a \cdot b) = \delta(a) \cdot b + \delta(b). \qquad \text{(Leibniz rule)}$$

A derivation δ of a polynomial ring $R[x]$ is uniquely defined once we fix $\delta(x) \in R[x]$. For instance, $\partial_x : R[x] \to R[x]$ is the unique derivation δ of the polynomial ring s.t. $\delta(x) = 1$. Other technical notions will be recalled when necessary. For a general introduction to algebraic geometry we refer to [25].

2 Weighted extension of basic parallel processes

2.1 Basic parallel processes

In this section we recall the notion of basic parallel process (BPP) together with its language semantics. Let $\{X_1, X_2, \dots\}$ be a countable set of *nonterminals* (process variables) and let Σ be a finite alphabet of *terminals* (actions). A *BPP expression* is generated by the following abstract grammar (cf. [29, Sec. 5]): $E, F ::= \bot \mid X_i \mid a.E \mid E + F \mid E \parallel F$. Intuitively, $\bot$ is a constant representing the *terminated process*, $a.E$ (*action prefix*), is the process that performs action a and becomes E, $E + F$ (*choice*) is the process that behaves like E or F, and $E \parallel F$ (*merge*) is the *parallel* execution of E and F. We say that an expression E is *guarded* if every occurrence of a nonterminal X_i is under the scope of an action prefix. A *BPP* consists of a distinguished *starting nonterminal* X_1 and rules

$$X_1 \to E_1 \quad \cdots \quad X_k \to E_k, \tag{1}$$

where the r.h.s. expressions $E_1, \dots, E_k$ are guarded and contain only nonterminals $X_1, \dots, X_k$.

$$a.E \xrightarrow{a} E \qquad \frac{E \xrightarrow{a} E'}{E + F \xrightarrow{a} E'} \qquad \frac{F \xrightarrow{a} F'}{E + F \xrightarrow{a} F'}$$

$$\frac{E_i \xrightarrow{a} E'}{X_i \xrightarrow{a} E'} \qquad \frac{E \xrightarrow{a} E'}{E \parallel F \xrightarrow{a} E' \parallel F} \qquad \frac{F \xrightarrow{a} F'}{E \parallel F \xrightarrow{a} E \parallel F'}$$

Figure 1 BPP transition rules.

A BPP induces an infinite labelled transition system where states are expressions and the labelled transition relations $\xrightarrow{a}$ are the least family of relations closed under the rules from Fig. 1. The transition relation is extended naturally to words $\xrightarrow{w}$, $w \in \Sigma^*$. An expression E is *final* if there are no a, E' s.t. $E \xrightarrow{a} E'$ (e.g., $\bot \parallel \bot$); it *accepts* a word $w \in \Sigma^*$ if there is a final expression F s.t. $E \xrightarrow{w} F$. The *language* $L(E)$ recognised by an expression E is the set of words it accepts, and the language of a BPP is $L(X_1)$.

An expression E is in *(full) standard form* if it is a sum of products $a_1.\alpha_1 + \cdots + a_n.\alpha_n$, where each α_i is a merge of nonterminals; a BPP (1) is in standard form if every $E_1, \dots, E_k$ is in standard form. The standard form for BPP is analogous to the Greibach normal form for context-free grammars [35]. Every BPP can be effectively transformed to one in standard form preserving bisimilarity [21, Proposition 2.31], and thus the language it recognises.

▶ **Example 5.** Consider two input symbols $\Sigma = \{a, b\}$ and two nonterminals $N = \{S, X\}$. The following is a BPP in standard form: $S \to a.X, X \to a.(X \parallel X) + b.\bot$. An example execution is $S \xrightarrow{a} X \xrightarrow{a} X \parallel X \xrightarrow{b} \bot \parallel X \xrightarrow{b} \bot \parallel \bot$, and thus $a^2b^2 \in L(S)$.

While language equivalence is undecidable for BPP [36, Sec. 5], the finer bisimulation equivalence is decidable [22], and in fact PSPACE-complete [70, 43]. These initial results have motivated a rich line of research investigating decidability and complexity for variants of bisimulation equivalence. We consider another classical variation on language equivalence, namely multiplicity equivalence, and apply it to decide language equivalence of unambiguous BPP. We show in Corollary 2 that both problems are decidable and in 2-EXPSPACE. This is obtained by considering a more general model, introduced next.

2.2 Weighted basic parallel processes

Preliminaries. Let $\Sigma^* \to \mathbb{Q}$ be the set of *(non-commutative) series* with coefficients in $\mathbb{Q}$, also known as *weighted languages*. An alternative notation is $\mathbb{Q}\langle\langle\Sigma\rangle\rangle$. We write a series as $f = \sum_{w\in\Sigma^*} f_w \cdot w$, where the value of f at w is $f_w \in \mathbb{Q}$. Thus, $3aba - \frac{5}{2}bc$ and $1 + a + a^2 + \cdots$ are series. The set of series carries the structure of a vector space over $\mathbb{Q}$, with element-wise scalar product $c \cdot f$ ($c \in \mathbb{Q}$) and sum $f + g$. The *support* of a series f is the subset of its domain $\mathsf{supp}(f) \subseteq \Sigma^*$ where it evaluates to a nonzero value. *Polynomials* $\mathbb{Q}\langle\Sigma\rangle$ are series with finite support. The *characteristic series* of a language $L \subseteq \Sigma^*$ is the series that maps words in L to 1 and all the other words to 0.

For two words $u \in \Sigma^m$ and $v \in \Sigma^n$, let $u \sqcup\!\sqcup v$ be the multiset of all words $w = a_1 \cdots a_{m+n}$ s.t. the set of indices $\{1, \ldots, m+n\}$ can be partitioned into two subsequences $i_1 < \cdots < i_m$ and $j_1 < \cdots < j_n$ s.t. $u = a_{i_1} \cdots a_{i_m}$ and $v = a_{j_1} \cdots a_{j_m}$. The multiset semantics preserves multiplicities, e.g., $ab \sqcup\!\sqcup a = \{\{aab, aab, aba\}\}$. The *shuffle* of two series f, g is the series $f \sqcup\!\sqcup g$ defined as $(f \sqcup\!\sqcup g)_w := \sum_{w\in u \sqcup\!\sqcup v} f_u \cdot g_v$, for every $w \in \Sigma^*$, where the sum is taken with multiplicities. Shuffle product (called *Hurwitz product* in [31]) leads to the commutative *ring of shuffle series* $(\mathbb{Q}\langle\langle\Sigma\rangle\rangle; +, \sqcup\!\sqcup, 0, 1)$, whose *shuffle identity* 1 is the series mapping ε to 1 and all other words to 0. A series f has a *shuffle inverse* g, i.e., $f \sqcup\!\sqcup g = 1$, iff $f_\varepsilon \neq 0$. The *n-th shuffle power* $f^{\sqcup\!\sqcup n}$ of a series f is inductively defined by $f^{\sqcup\!\sqcup 0} := 1$ and $f^{\sqcup\!\sqcup (n+1)} := f \sqcup\!\sqcup f^{\sqcup\!\sqcup n}$.

Consider the mapping $\delta : \Sigma^* \to \mathbb{Q}\langle\langle\Sigma\rangle\rangle \to \mathbb{Q}\langle\langle\Sigma\rangle\rangle$ s.t. for every $u \in \Sigma^*$ and $f \in \mathbb{Q}\langle\langle\Sigma\rangle\rangle$, $\delta_u f \in \mathbb{Q}\langle\langle\Sigma\rangle\rangle$ is the series defined as $(\delta_u f)_w = f_{uw}$, for every $w \in \Sigma^*$. We call $\delta_u f$ the *u-derivative* of f (a.k.a. *shift* or *left-quotient*). For example, $\delta_a(ab + c) = b$. The derivative operation δ_u is linear, for every $u \in \Sigma^*$. The one-letter derivatives δ_a's are (noncommuting) derivations of the shuffle ring since they satisfy (Leibniz rule),

$$\delta_a(f \sqcup\!\sqcup g) = \delta_a f \sqcup\!\sqcup g + f \sqcup\!\sqcup \delta_a g, \quad \text{for all } a \in \Sigma, f, g \in \mathbb{Q}\langle\langle\Sigma\rangle\rangle, \tag{2}$$

Syntax and semantics. A *weighted basic parallel process* (WBPP) is a tuple $P = (\Sigma, N, S, F, \Delta)$ where Σ is a finite input alphabet of *terminal symbols/actions*, N is a finite set of *nonterminal symbols/processes*, $S \in N$ is the *initial nonterminal*, $F : N \to \mathbb{Q}$ assigns a *final weight* $FX \in \mathbb{Q}$ to each nonterminal $X \in N$, and $\Delta : \Sigma \times N \to \mathbb{Q}[N]$ is a *transition function* mapping a nonterminal $X \in N$ and an input symbol $a \in \Sigma$ to a polynomial $\Delta_a X \in \mathbb{Q}[N]$.

▶ **Example 6.** A BPP in standard form is readily converted to a WBPP with $0, 1$ weights: The BPP from Example 5 yields the WBPP with output function $FS = FX = 0$ and transitions $\Delta_a S = X, \Delta_a X = X^2, \Delta_b S = 0, \Delta_b X = 1$. Configurations reachable from S, X are of the form cX^n ($c \in \mathbb{N}$). Action "a" acts as an increment $X^n \xrightarrow{a} nX^{n+1}$ and "b" as a decrement $X^n \xrightarrow{b} nX^{n-1}$. The constant coefficient $c \in \mathbb{N}$ in a reachable configuration cX^n keeps track of the "multiplicity" of reaching this configuration, i.e., the number of distinct runs leading to it. For instance, $[\![S]\!]_{a^2b^2} = 2$ since $S \xrightarrow{a} X \xrightarrow{a} X^2 \xrightarrow{b} 2X \xrightarrow{b} 2$. In the underlying BPP,

$$S \xrightarrow{a} X \xrightarrow{a} X \parallel X \begin{array}{l} \xrightarrow{b} \bot \parallel X \xrightarrow{b} \bot \parallel \bot \\ \xrightarrow[b]{} X \parallel \bot \xrightarrow{b} \bot \parallel \bot \end{array}$$

where the branching upon reading the first symbol "b" depends on whether the first or second occurrence of X reads this symbol.

A *configuration* of a WBPP is a polynomial $\alpha \in \mathbb{Q}[N]$. The transition function extends uniquely to a derivation of the polynomial ring $\mathbb{Q}[N]$ via linearity and (Leibniz rule):

$$\begin{aligned}
&\Delta : \Sigma \times \mathbb{Q}[N] \to \mathbb{Q}[N] \\
&\Delta_a(c \cdot \alpha) = c \cdot \Delta_a(\alpha), && \forall a \in \Sigma, c \in \mathbb{Q}, \\
&\Delta_a(\alpha + \beta) = \Delta_a(\alpha) + \Delta_a(\beta), && \forall a \in \Sigma, \alpha, \beta \in \mathbb{Q}[N], \\
&\Delta_a(\alpha \cdot \beta) = \Delta_a(\alpha) \cdot \beta + \alpha \cdot \Delta_a(\beta), && \forall a \in \Sigma, \alpha, \beta \in \mathbb{Q}[N]. \qquad (3)
\end{aligned}$$

For example, from configuration $X \cdot Y$ we can read a and go to $\Delta_a(X \cdot Y) = \Delta_a(X) \cdot Y + X \cdot \Delta_a(Y)$; this is models the fact that either X reads a and Y is unchanged, or vice versa. The transition function is then extended homomorphically to words:

$$\begin{aligned}
&\Delta : \Sigma^* \times \mathbb{Q}[N] \to \mathbb{Q}[N] \\
&\Delta_\varepsilon \alpha := \alpha,\ \Delta_{a \cdot w} \alpha := \Delta_w(\Delta_a \alpha), \quad \forall (a \cdot w) \in \Sigma^*, \alpha \in \mathbb{Q}[N]. \qquad (4)
\end{aligned}$$

Sometimes we write $\alpha \xrightarrow{w} \beta$ when $\beta = \Delta_w(\alpha)$. For instance, from configuration α we can read $ab \in \Sigma^*$ visiting configurations $\alpha \xrightarrow{a} \Delta_a(\alpha) \xrightarrow{b} \Delta_b(\Delta_a(\alpha))$. The order of reading symbols matters: For the transition function $\Delta_a(X) = 0$, $\Delta_b(X) = Y$, and $\Delta_a(Y) = \Delta_b(Y) = 1$, we have $X \xrightarrow{a} 0 \xrightarrow{b} 0$ but $X \xrightarrow{b} Y \xrightarrow{a} 1$. The *semantics* of a WBPP is the mapping

$$\begin{aligned}
&[\![_]\!] : \mathbb{Q}[N] \to \mathbb{Q}\langle\langle \Sigma \rangle\rangle \\
&[\![\alpha]\!]_w := F(\Delta_w \alpha), \quad \forall \alpha \in \mathbb{Q}[N], w \in \Sigma^*. \qquad (5)
\end{aligned}$$

Here F is extended homomorphically from nonterminals to configurations: $F(\alpha + \beta) = F(\alpha) + F(\beta)$ and $F(\alpha \cdot \beta) = F(\alpha) \cdot F(\beta)$. We say that configuration α *recognises* the series $[\![\alpha]\!]$. The series recognised by a WBPP is the series recognised by its initial nonterminal. A *WBPP series* is a series which is recognised by some WBPP.

▶ **Example 7.** We show a WBPP series which is not a WFA series. In particular, its support is nonregular support since WFA supports include the regular languages. Consider the WBPP from Example 6. The language $L := \mathsf{supp}([\![S]\!]) \cap a^*b^*$ is the set of words of the form $a^n b^n$, which is not regular, and thus $\mathsf{supp}([\![S]\!])$ is not regular either. Moreover, $[\![S]\!]$ is not a WFA series: 1) the set M of words of the form $a^m b^n$ with $m \neq n$ is a WFA support, 2) if a language and its complement are WFA supports, then they are regular by a result of Restivo and Reutenauer [65, Theorem 3.1], and 3) since M is not regular, it follows that its complement is not a WFA support, and thus $L = (\Sigma^* \setminus M) \cap a^*b^*$ is not a WFA support either.

2.3 Basic properties

We present some basic properties of the semantics of WBPP. First of all, applying the derivative δ_w to the semantics corresponds to applying Δ_w to the configuration.

▶ **Lemma 8** (Exchange). *For every $\alpha \in \mathbb{Q}[N]$ and $w \in \Sigma^*$, $\delta_w [\![\alpha]\!] = [\![\Delta_w \alpha]\!]$.*

As a consequence, the semantics is a homomorphism from configurations to series.

▶ **Lemma 9** (Homomorphism). *The semantics function $[\![\_]\!]$ is a homomorphism from the polynomial to the shuffle series ring:*

$$\begin{aligned}
&[\![_]\!] : (\mathbb{Q}[N]; +, \cdot) \to (\mathbb{Q}\langle\langle N \rangle\rangle; +, ⧢) \\
&[\![c \cdot \alpha]\!] = c \cdot [\![\alpha]\!], \quad [\![\alpha + \beta]\!] = [\![\alpha]\!] + [\![\beta]\!], \quad [\![\alpha \cdot \beta]\!] = [\![\alpha]\!] ⧢ [\![\beta]\!].
\end{aligned}$$

Lemmas 8 and 9 illustrate the interplay between the syntax and semantics of WBPP, and they can be applied to obtain some basic closure properties for the class of WBPP series.

▶ **Lemma 10** (Closure properties). *Let $f, g \in \mathbb{Q}\langle\langle\Sigma\rangle\rangle$ be WBPP series. The following series are also WBPP: $c \cdot f$, $f + g$, $f \sqcup\!\sqcup g$, $\delta_a f$, the shuffle inverse of f (when defined).*

WBPP series generalise the rational series (i.e., recognised by finite weighted automata [8]), which in fact correspond to WBPP with a linear transition relation.

▶ **Example 11.** The shuffle of two WBPP series with context-free support can yield a WBPP series with non-context-free support. Consider the WBPP from Example 6 over $\Sigma = \{a, b\}$. Make a copy of this WBPP over a disjoint alphabet $\Gamma = \{c, d\}$ with nonterminals $\{T, Y\}$. Now consider the shuffle $f := [\![S]\!] \sqcup\!\sqcup [\![T]\!] \in \mathbb{Q}\langle\langle\Sigma \cup \Gamma\rangle\rangle$. It is WBPP recognisable by Lemma 10. (For instance we can add a new initial nonterminal U with rules $\Delta_a U = X \cdot T$, $\Delta_c U = S \cdot Y$, and $\Delta_b U = \Delta_d U = 0$.) $\mathsf{supp}(f)$ is not context free, since intersecting it with the regular language $a^* c^* b^* d^*$ yields $\{a^m c^n b^m d^n \mid m, n \in \mathbb{N}\}$, which is not context-free by the pumping lemma for context-free languages [37, Theorem 7.18] (cf. [57, Problem 101]).

2.4 Differential algebra of shuffle-finite series

Differential algebra allows us to provide an elegant characterisation of WBPP series. An *algebra* (over $\mathbb{Q}$) is a vector space equipped with a bilinear product. Shuffle series are a commutative algebra, called *shuffle series algebra*. A subset of $\mathbb{Q}\langle\langle\Sigma\rangle\rangle$ is a *subalgebra* if it contains $\mathbb{Q}$ and is closed under scalar product, addition, and shuffle product. It is *differential* if it is closed under derivations δ_a ($a \in \Sigma$). By Lemma 10, WBPP series are a differential subalgebra. Let $\mathbb{Q}[f^{(1)}, \ldots, f^{(k)}] \subseteq \mathbb{Q}\langle\langle\Sigma\rangle\rangle$ be the smallest subalgebra containing $f^{(1)}, \ldots, f^{(k)} \in \mathbb{Q}\langle\langle\Sigma\rangle\rangle$. Algebras of this form are called *finitely generated*. A series is *shuffle finite* if it belongs to a finitely generated differential subalgebra of shuffle series.

▶ **Theorem 12.** *A series is shuffle finite iff it is WBPP.*

The characterisation above provides an insight into the algebraic structure of WBPP series. Other classes of series can be characterised in a similar style. For instance, a series is accepted by a WFA iff it belongs to a finitely generated differential vector space over $\mathbb{Q}$ [8, Proposition 5.1]; by a weighted context-free grammar iff it belongs to a δ_a-closed, finitely generated subalgebra of the algebra of series with (noncommutative) *Cauchy product* ($(f * g)_w := \sum_{w=u \cdot v} f_u \cdot f_v$); and by a polynomial automaton [3] iff its reversal ($f^R_{a_1 \cdots a_n} := f_{a_n \cdots a_1}$) belongs to a δ_a-closed, finitely generated subalgebra of the algebra of series with *Hadamard product* ($(f \odot g)_w := f_w \cdot g_w$). Considering other products yields novel classes of series, too. For instance, the *infiltration product* [2] yields the class of series that belong to a δ_a-closed, finitely generated subalgebra of the algebra of series with infiltration product.

2.5 Equivalence and zeroness problems

The *WBPP equivalence problem* takes in input two WBPP P, Q and amounts to determine whether $[\![P]\!] = [\![Q]\!]$. In the special case where $[\![Q]\!] = 0$, we have an instance of the *zeroness problem*. Since WBPP series form an effective vector space, equivalence reduces to zeroness, and thus we concentrate on the latter.

Evaluation and word-zeroness problems. We first discuss a simpler problem, which will be a building block in our zeroness algorithm. The *evaluation problem* takes in input a WBPP with initial configuration α and a word $w \in \Sigma^*$, and it amounts to compute $[\![\alpha]\!]_w$. The *word-zeroness* problem takes the same input, and it amounts to decide whether $[\![\alpha]\!]_w = 0$.

▶ **Theorem 13.** *The evaluation and word-zeroness problems for WBPP are in PSPACE.*

The proof follows from the following three ingredients: The construction of an algebraic circuit of exponential size computing the polynomial $\Delta_w\alpha$ (Lemma 14), the fact that this polynomial has polynomial degree (Lemma 15), and the fact that circuits computing multivariate polynomials of polynomial degree can be evaluated in NC [44, Theorem 2.4.5].

▶ **Lemma 14.** *Fix a word $w \in \Sigma$ and an initial configuration $\alpha \in \mathbb{Q}[N]$ of a WBPP, where $\alpha, \Delta_a X_i \in \mathbb{Q}[N]$ are the outputs of an algebraic circuit of size n. We can construct an algebraic circuit computing $\Delta_w\alpha$ of size $\leq 4^{|w|} \cdot n$. The construction can be done in space polynomial in $|w|$ and logarithmic in n.*

▶ **Lemma 15.** *Let $D \in \mathbb{N}$ be the maximum of the degree of the transition relation Δ and the initial configuration α. The configuration $\Delta_w\alpha \in \mathbb{Q}[N]$ reached by reading a word $w \in \Sigma^n$ of length n has total degree $O(n \cdot D)$.*

Decidability of the zeroness problem. Fix a WBPP and a configuration $\alpha \in \mathbb{Q}[N]$. Suppose we want to decide whether $[\![\alpha]\!]$ is zero. An algorithm for this problem follows from first principles. Recall that an *ideal* $I \subseteq \mathbb{Q}[N]$ is a subset closed under addition, and multiplication by arbitrary polynomials [25, §4, Ch. 1]. Let $\langle S\rangle$ be the smallest ideal including $S \subseteq \mathbb{Q}[N]$. Intuitively, this is the set of "logical consequences" of the vanishing of polynomials in S. Build a chain of polynomial ideals

$$I_0 \subseteq I_1 \subseteq \cdots \subseteq \mathbb{Q}[N], \text{ with } I_n := \langle \Delta_w\alpha \mid w \in \Sigma^{\leq n}\rangle, n \in \mathbb{N}. \tag{6}$$

Intuitively, I_n is the set of polynomials that vanish as a consequence of the vanishing of $\Delta_w\alpha$ for all words w of length $\leq n$. The chain above has some important structural properties, essentially relying on the fact that the Δ_a's are derivations of the polynomial ring.

▶ **Lemma 16.** *1. $\Delta_a I_n \subseteq I_{n+1}$. 2. $I_{n+1} = I_n + \langle\bigcup_{a\in\Sigma} \Delta_a I_n\rangle$. 3. $I_n = I_{n+1}$ implies $I_n = I_{n+1} = I_{n+2} = \cdots$.*

By Hilbert's finite basis theorem [25, Theorem 4, §5, Ch. 2], there is $M \in \mathbb{N}$ s.t. $I_M = I_{M+1} = \cdots$. By Lemma 16 (3) and decidability of ideal inclusion [53], M can be computed. This suffices to decide WBPP zeroness. Indeed, let $\Delta_{w_1}\alpha, \ldots, \Delta_{w_m}\alpha$ be the generators of I_M. For every input word $w \in \Sigma^*$ there are $\beta_1, \ldots, \beta_m \in \mathbb{Q}[N]$ s.t. $\Delta_w\alpha = \beta_1 \cdot \Delta_{w_1}\alpha + \cdots \beta_m \cdot \Delta_{w_m}\alpha$. By applying the output function F on both sides, we have $[\![\alpha]\!]_w = F(\Delta_w\alpha) = F\beta_1 \cdot [\![\alpha]\!]_{w_1} + \cdots + F\beta_m \cdot [\![\alpha]\!]_{w_m}$. It follows that if $[\![\alpha]\!]_w = 0$ for all words of length $\leq M$, then $[\![\alpha]\!] = 0$. One can thus enumerate all words w of length $\leq M$ and check $[\![\alpha]\!]_w = 0$ with Theorem 13. So far we only know that M is computable. In the next section we show that in fact M is an elementary function of the input WBPP.

Elementary upper bound for the zeroness problem. We present an elementary upper bound on the length of the chain of polynomial ideals (6). This is obtained by generalising the case of a single derivation from Novikov and Yakovenko [59, Theorem 4] to the situation of several, not necessarily commuting derivations Δ_a, $a \in \Sigma$. The two main ingredients in the proof of [59, Theorem 4] are 1) a structural property of the chain (6) called *convexity*, and 2) a degree bound on the generators of the n-th ideal I_n (which we have already established in Lemma 15). For two sets $I, J \subseteq \mathbb{Q}[N]$ consider the *colon set* $I : J := \{f \in \mathbb{Q}[N] \mid \forall g \in J, f \cdot g \in I\}$ [25, Def. 5, §4, Ch. 4]. If I, J are ideals of $\mathbb{Q}[N]$ then $I : J$ is also an ideal. An ideal chain $I_0 \subseteq I_1 \subseteq \cdots$ is *convex* if the colon ideals $I_n : I_{n+1}$ form themselves a chain $I_0 : I_1 \subseteq I_1 : I_2 \subseteq \cdots$. Chain of ideals obtained by iterated application of a single derivation are convex by [59, Lemma 7]. We extend this observation to a finite set of derivations.

▶ **Lemma 17** (generalisation of [59, Lemma 7]). *The ideal chain* (6) *is convex.*

Proof. We extend the argument from [59] to the case of many derivations. Assume $f \in I_{n-1} : I_n$ and let $h \in I_{n+1}$ be arbitrary. We have to show $f \cdot h \in I_n$.

▷ Claim. $f \cdot \Delta_a g \in I_n$, for all $a \in \Sigma$ and $g \in I_n$.

Proof of the claim. Since Δ_a is a derivation (4), $\Delta_a(f \cdot g) = \Delta_a f \cdot g + f \cdot \Delta_a g$, and by solving for $f \cdot \Delta_a g$ we can write $f \cdot \Delta_a g = \underbrace{\Delta_a(\overbrace{f \cdot g}^{\text{(a) } I_{n-1}})}_{\text{(b) } I_n} - \underbrace{\Delta_a f \cdot g}_{\text{(c) } I_n}$. Condition (a) follows from the definition of colon ideal, (b) from point (1) of Lemma 16, and (c) from I_n being an ideal. ◁

Since $h \in I_{n+1}$, by point (2) of Lemma 16, we can write $h = h_0 + h_1$ with $h_0 \in I_n$ and $h_1 \in \langle \bigcup_{a \in \Sigma} \Delta_a I_n \rangle$. In particular, $h_1 = \sum_i p_i \cdot \Delta_{a_i} g_i$ with $g_i \in I_n$, By the claim, $f \cdot h_1 = \sum_i p_i \cdot f \cdot \Delta_{a_i} g_i \in I_n$. Consequently, $f \cdot h = f \cdot h_0 + f \cdot h_1 \in I_n$ as well. ◀

Thanks to Lemma 17 we can generalise the whole proof of [59, Theorem 4], eventually arriving at the following elementary bound. The *order* of a WBPP is the number of nonterminals and its *degree* is the maximal degree of the polynomials $\Delta_a X$ $(a \in \Sigma, X \in N)$.

▶ **Theorem 18.** *Consider a WBPP of order $\leq k$ and degree $\leq D$. The length of the ideal chain* (6) *is at most* $D^{k^{O(k^2)}}$.

The elementary bound above may be of independent interest. Already in the case of a single derivation, it is not known whether the bound from [59] is tight, albeit it is expected not to be so. We provide a proof sketch of Theorem 18 in order to illustrate the main notions from algebraic geometry which are required.

Proof sketch. We recall some basic facts from algebraic geometry. The *radical* $\sqrt{I}$ of an ideal I is the set of elements r s.t. $r^m \in I$ for some $m \in \mathbb{N}$; note that $\sqrt{I}$ is itself an ideal. An ideal I is *primary* if $p \cdot q \in I$ and $p \notin I$ implies $q \in \sqrt{I}$. A *primary decomposition* of an ideal I is a collection of primary ideals $\{Q_1, \ldots, Q_s\}$, called *primary components*, s.t. $I = Q_1 \cap \cdots \cap Q_s$. The *dimension* $\mathsf{dim}\, I$ of a polynomial ideal $I \subseteq \mathbb{Q}[k]$ is the dimension of its associated variety $V(I) = \{x \in \mathbb{C}^k \mid \forall p \in I.p(x) = 0\}$. Since the operation of taking the variety of an ideal is inclusion-reversing, ideal inclusion is dimension-reversing: $I \subseteq J$ implies $\mathsf{dim}\, I \geq \mathsf{dim}\, J$. Consider a convex chain of polynomial ideals as in (6). By convexity, the colon ideals also form a chain $I_0 : I_1 \subseteq I_1 : I_2 \subseteq \cdots \subseteq \mathbb{Q}[k]$. The colon dimensions are at most k and non-increasing, $k \geq \mathsf{dim}\,(I_0 : I_1) \geq \mathsf{dim}\,(I_1 : I_2) \geq \cdots$. Divide the original ideal chain (6) into segments, where in the i-th segment the colon dimension is a constant m_i:

$$\underbrace{I_0 \subseteq \cdots \subseteq I_{n_0-1}}_{\mathsf{dim}\,(I_n\,:\,I_{n+1})=m_0} \subseteq \underbrace{I_{n_0} \subseteq \cdots \subseteq I_{n_1-1}}_{\mathsf{dim}\,(I_n\,:\,I_{n+1})=m_1} \subseteq \cdots \subseteq \underbrace{I_{n_i} \subseteq \cdots \subseteq I_{n_{i+1}-1}}_{\mathsf{dim}\,(I_n\,:\,I_{n+1})=m_i} \subseteq \cdots . \tag{7}$$

Since the colon dimension can strictly decrease at most k times, there are at most k segments. In the following claim we show that the length of a convex ideal chain with equidimensional colon ideal chain can be bounded by the number of primary components of the initial ideal.

▷ Claim 19 ([59, Lemmas 8+9]). Consider a strictly ascending convex chain of ideals $I_0 \subsetneq I_1 \subsetneq \cdots \subsetneq I_\ell$ of length ℓ where the colon ratios have the same dimension $m := \mathsf{dim}\,(I_0 : I_1) = \cdots = \mathsf{dim}\,(I_{\ell-1} : I_\ell)$. Then ℓ is at most the number of primary components of any primary ideal decomposition of the initial ideal I_0 (counted with multiplicities[1]).

We apply Claim 19 to the i-th segment (7) and obtain that its length $\ell_i := n_{i+1} - n_i$ is at most the number of primary components in any primary ideal decomposition of its starting ideal I_{n_i}. We now use a result from effective commutative algebra showing that we can compute primary ideal decompositions of size bounded by the degree of the generators.

▷ Claim 20 (variant of [59, Corollary 2]). An ideal $I \subseteq \mathbb{C}[k]$ generated by polynomials of degree $\leq D$ admits a primary ideal decomposition of size $D^{k^{O(k)}}$ (counted with multiplicities).

By Claim 20, I_{n_i} admits some primary decomposition of size $d_i^{k^{O(k)}}$, where d_i is the maximal degree of the generators of I_{n_i}. By Lemma 15, d_i is at most $O\,(D \cdot n_i)$. All in all, the i-th segment has length $\ell_i = n_{i+1} - n_i \leq (D \cdot n_i)^{k^{O(k)}}$. We have $n_i \leq O\,(f_i)$ where f_i satisfies $f_{i+1} \leq a \cdot f_i^b$ with $a = D^b$ and $b = k^{O(k)}$. Thus $f_k \leq a \cdot a^b \cdots a^{b^{k-1}} \leq a^{b^{O(k)}}$, yielding the required upper bound on the length of the ideal chain $n_k \leq D^{k^{O\left(k^2\right)}}$. ◀

Thanks to the bound from Theorem 18, we obtain the main contribution of the paper, which was announced in the introduction.

▶ **Theorem 1.** *The zeroness problem for WBPP is in 2-EXPSPACE.*

Proof. The bound on the length of the ideal chain (6) from Theorem 18 implies that if the WBPP is not zero, then there exists a witnessing input word of length at most doubly exponential. We can guess this word and verify its correctness in 2-EXPSPACE by Theorem 13. This is a nondeterministic algorithm, but by courtesy of Savitch's theorem [67] we obtain a *bona fide* deterministic 2-EXPSPACE algorithm. ◀

Application to BPP. The *multiplicity semantics* of a BPP is its series semantics as an $\mathbb{N}$-WBPP. Intuitively, one counts all possible ways in which an input is accepted by the model. The *BPP multiplicity equivalence problem* takes as input two BPP P, Q and returns "yes" iff P, Q have the same multiplicity semantics. Decidability of BPP multiplicity equivalence readily follows from Theorem 1. We say that a BPP is *unambiguous* if its multiplicity semantics is $\{0, 1\}$-valued. While BPP language equivalence is undecidable [36, Sec. 5], we obtain decidability for unambiguous BPP. We have thus proved Corollary 2. This generalises decidability for deterministic BPP, which follows from decidability of bisimulation equivalence [22]. Language equivalence of unambiguous context-free grammars, the sequential counterpart of BPP (sometimes called BPA in process algebra), is a long-standing open problem, as well as the more general multiplicity equivalence problem (cf. [33, 23, 1]).

[1] We refer to [59, Sec. 4.1] for the notion of *multiplicity* of a primary component.

3 Constructible differentially finite power series

In this section we study a class of multivariate power series in commuting variables called *constructible differentially finite* (CDF) [6, 7]. We show that CDF power series arise naturally as the commutative variant of WBPP series from § 2. Stated differently, the novel WBPP can be seen as the noncommutative variant of CDF, showing a connection between the theory of weighted automata and differential equations. As a consequence, by specialising to the commutative context the 2-EXPSPACE WBPP zeroness procedure, we obtain an algorithm to decide zeroness for CDF power series in 2-EXPTIME. This is the main result of the section, which was announced in the introduction (Theorem 3).

On the way, we recall and further develop the theory of CDF power series. In particular, we provide a novel closure under regular support restrictions (Lemma 24). In § 4 we illustrate a connection between CDF power series and combinatorics, by showing that the generating series of a class of constructible species of structures are CDF, which will broaden the applicability of the CDF zeroness algorithm to multiplicity equivalence of species.

Preliminaries. In the rest of the section, we consider commuting variables $x = (x_1, \ldots, x_d)$, $y = (y_1, \ldots, y_k)$. We denote by $\mathbb{Q}[\![x]\!]$ the set of multivariate power series in x, endowed with the structure of a commutative ring $(\mathbb{Q}[\![x]\!]; +, \cdot, 0, 1)$ with pointwise addition and (Cauchy) product. The partial derivatives ∂_{x_j}'s satisfy (Leibniz rule), and thus form a family of commuting derivations of this ring. To keep notations compact, we use vector notation: For a tuple of naturals $n = (n_1, \ldots, n_d) \in \mathbb{N}^d$, define $n! := n_1! \cdots n_d!$, $x^n := x_1^{n_1} \cdots x_d^{n_d}$, and $\partial_x^n := \partial_{x_1}^{n_1} \cdots \partial_{x_d}^{n_d}$. We write a power series as $f = \sum_{n \in \mathbb{N}^d} f_n \cdot \frac{x^n}{n!} \in \mathbb{Q}[\![x]\!]$, and define the *(exponential) coefficient extraction* operation $[x^n]f := f_n$, for every $n \in \mathbb{N}^d$. This is designed in order to have the following simple commuting rule with partial derivative:

$$[x^m](\partial_x^n f) = [x^{m+n}]f, \quad \text{for all } m, n \in \mathbb{N}^d. \tag{8}$$

Coefficient extraction is linear, and *constant term* extraction $[x^0]$ is even a homomorphism since $[x^0](f \cdot g) = [x^0]f \cdot [x^0]g$. The *Jacobian matrix* of a tuple of power series $f = (f^{(1)}, \ldots, f^{(k)}) \in \mathbb{Q}[\![x]\!]^k$ is the matrix $\partial_x f \in \mathbb{Q}[\![x]\!]^{k \times d}$ where entry (i, j) is $\partial_{x_j} f^{(i)}$. Consider commuting variables $y = (y_1, \ldots, y_k)$. For a set of indices $I \subseteq \{1, \ldots, k\}$, by y_I we denote the tuple of variables y_i s.t. $i \in I$ and by $y_{\setminus I}$ we denote the tuple of variables y_i s.t. $i \notin I$. A power series $f \in \mathbb{Q}[\![y]\!]$ is *locally polynomial w.r.t.* y_I if $f \in \mathbb{Q}[y_I][\![y_{\setminus I}]\!]$ (f is a power series in $y_{\setminus I}$ with coefficients polynomial in y_I), and that it is *polynomial w.r.t.* y_I if $f \in \mathbb{Q}[\![y_{\setminus I}]\!][y_I]$ (f is a polynomial in y_I with coefficients which are power series in $y_{\setminus I}$). For instance $\frac{1}{1 - y_1 \cdot y_2} = 1 + y_1 y_2 + (y_1 y_2)^2 + \cdots$ is not polynomial, but it is locally polynomial in $y_{\{1\}}$ (and $y_{\{2\}}$). A power series $f \in \mathbb{Q}[\![x, y]\!]$ and a tuple $g = (g^{(1)}, \ldots, g^{(k)}) \in \mathbb{Q}[\![x]\!]^k$ are *y-composable* if f is locally polynomial w.r.t. y_I, where I is the set of indices i s.t. $g^{(i)}(0) \neq 0$; *strongly y-composable* is obtained by replacing "locally polynomial" with "polynomial". As a corner case often arising in practice, f, g are always strongly y-composable when $g(0) = 0$. When f, g are y-composable, their *composition* $f \circ_y g \in \mathbb{Q}[\![x]\!]$ obtained by replacing y_i in f with $g^{(i)}$, for every $1 \leq i \leq k$, exists. Composition extends component-wise to vectors and matrices.

3.1 Multivariate CDF power series

A power series $f^{(1)} \in \mathbb{Q}[\![x]\!]$ is CDF [6, 7] if it is the first component of a solution $f = (f^{(1)}, \ldots, f^{(k)}) \in \mathbb{Q}[\![x]\!]^k$ of a system of polynomial partial differential equations

$$\partial_x f = P \circ_y f, \quad \text{where } P \in \mathbb{Q}[x, y]^{k \times d}. \tag{9}$$

We call k the *order* of the system and d its *dimension*; in the univariate case $d = 1$, (9) is a system of ordinary differential equations. The matrix P is called the *kernel* of the system. The *degree* the system is the maximum degree of polynomials in the kernel, and so it is its *height*. When the kernel does not contain x the system is called *autonomous*, otherwise *non-autonomous*. There is no loss of expressive power in considering only autonomous systems. Many analytic functions give rise to univariate CDF power series, such as polynomials, the exponential series $f := e^x = 1 + x + x^2/2! + \cdots$ (since $\partial_x f = f$), the trigonometric series $\sin x$, $\cos x$, $\sec x := 1/\cos x$, $\arcsin, \arccos, \arctan$ their hyperbolic variants $\sinh, \cosh, \tanh$, $\operatorname{sech} = 1/\cosh$, arsinh, artanh, the non-elementary error function $\operatorname{erf}(x) := \int_0^x e^{-t^2} dt$ (since $\partial_x \operatorname{erf} = e^{-x^2}$ and $\partial_x (e^{-x^2}) = -2x \cdot e^{-x^2}$). Multivariate CDF power series include polynomials, rational power series, constructible algebraic series (in the sense of [31, Sec. 2]; [6, Theorem 4],[7, Corollary 13]), and a large class of D-finite series ([7, Lemma 6]; but not all of them). Moreover, we demonstrate in Theorem 31 that the generating series of strongly constructible species are CDF. We recall some basic closure properties for the class of CDF power series.

▶ **Lemma 21** (Closure properties; [6, Theorem 2], [7, Theorem 11]). *(1) If $f, g \in \mathbb{Q}[\![x]\!]$ are CDF, then are also CDF: $c \cdot f$ for $c \in \mathbb{Q}$, $f + g$, $f \cdot g$, $\partial_{x_j} f$ for $1 \le j \le d$, $1/f$ (when defined). (2) If $\partial_{x_1} f, \ldots, \partial_{x_d} f$ are CDF, then so is f. (3) Closure under strong composition: If $f \in \mathbb{Q}[\![x, y]\!], g \in \mathbb{Q}[\![x]\!]^k$ are strongly y-composable and CDF, then $f \circ_y g$ is CDF.*

▶ Remark 22. In the univariate case $d = 1$, [6, Theorem 2] proves closure under composition under the stronger assumption $g(0) = 0$. In the multivariate case, [7, Theorem 11] claims without proof closure under composition (when defined). We leave it open whether CDF power series are closed under composition.

Of the many pleasant closure properties above, especially composition is remarkable, since this does not hold for other important classes of power series, such as the algebraic and the D-finite power series. For instance, e^x and $e^x - 1$ are D-finite, but $e^{e^x - 1}$ is not [46, Problem 7.8]. On the other hand, CDF power series are not closed under *Hadamard product*, already in the univariate case [6, Sec. 4]. (The *Hadamard product* of $f = \sum_{n \in \mathbb{N}^d} f_n \cdot x^n, g = \sum_{n \in \mathbb{N}^d} g_n \cdot x^n \in \mathbb{Q}[\![x]\!]$ is $f \odot g = \sum_{n \in \mathbb{N}^d} (f_n \cdot g_n) \cdot x^n$.) Another paramount closure property regards resolution of systems of power series equations. A system of equations of the constructible form $y = f$ with $f \in \mathbb{Q}[\![x, y]\!]^k$ is *well posed* if $f(0, 0) = 0$ and the Jacobian matrix evaluated at the origin $\partial_y f(0, 0)$ is nilpotent. A *canonical solution* is a series $g \in \mathbb{Q}[\![x]\!]^k$ solving the system for $y := g(x)$ s.t. $g(0) = 0$. The following is a slight generalisation of [7, Corollary 13].

▶ **Lemma 23** (Constructible power series theorem). *A well-posed system of equations $y = f(x, y)$ has a unique canonical solution $y := g(x)$. Moreover, if f is CDF, then g is CDF.*

For example, the unique canonical solution of the well-posed equation $y = f := x \cdot e^y$ is CDF.

3.2 Support restrictions

We discuss a novel closure property for CDF power series, which will be useful later in the context of combinatorial enumeration (§ 4). The *restriction* of $f \in \mathbb{Q}[\![x]\!]$ by a *support constraint* $S \subseteq \mathbb{N}^d$ is the series $f|_S \in \mathbb{Q}[\![x]\!]$ which agrees with f on the coefficient of x^n for every $n \in S$, and is zero otherwise. We introduce a small constraint language in order to express a class of support constraints. The set of *constraint expressions* of dimension $d \in \mathbb{N}$ is generated by the following abstract grammar,

$$\varphi, \psi \ ::= \ z_j = n \mid z_j \equiv n \pmod{m} \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \neg \varphi, \tag{10}$$

where $1 \leq j \leq d$ and $m, n \in \mathbb{N}$ with $m \geq 1$. Expressions $z_j \leq n$ and $z_j \geq n$ can be derived. The semantics of a constraint expressions φ of dimension d, written $[\![\varphi]\!] \subseteq \mathbb{N}^d$, is defined by structural induction in the expected way. For instance, the semantics of $z_1 \geq 2 \wedge z_2 \equiv 1 \ (\mathsf{mod}\ 2)$ is the set of pairs $(a, b) \in \mathbb{N}^2$ where $a \geq 2$ and b is odd. Call a set $S \subseteq \mathbb{N}^d$ *regular* if it is denoted by a constraint expression.

▶ **Lemma 24.** *CDF power series are closed under regular support restrictions.*

For instance, since e^x is CDF also $\sinh x = e^x|_{[\![z_1 \equiv 1 \ (\mathsf{mod}\ 2)]\!]}$ is CDF. CDF are not closed under more general *semilinear support restrictions*. E.g., restricting to the semilinear set $\{(m, \ldots, m) \in \mathbb{N}^d \mid m \in \mathbb{N}\}$ amounts to taking the *diagonal*, which in turn can be used to express the Hadamard product of power series [50, remark (2) on pg. 377], and CDF are closed under none of these operations.

3.3 CDF = Commutative WBPP series

We demonstrate that CDF power series correspond to WBPP series satisfying a commutativity condition. In particular, they coincide in the univariate case $x = (x_1)$ and $\Sigma = \{a_1\}$. A series $f \in \mathbb{Q}\langle\!\langle \Sigma \rangle\!\rangle$ over a finite alphabet $\Sigma = \{a_1, \ldots, a_d\}$ is *commutative* if $f_u = f_v$ whenever $\#(u) = \#(v)$; in this case we associate to it a power series $\mathsf{s2p}\,(f) \in \mathbb{Q}[\![x]\!]$ in commuting variables $x = (x_1, \ldots, x_d)$ by $\mathsf{s2p}\,(f) := \sum_{n \in \mathbb{N}^d} f_n \cdot \frac{x^n}{n!}$ where $f_n := f_w$ for any $w \in \Sigma^*$ s.t. $\#(w) = n$. Conversely, to any power series $f \in \mathbb{Q}[\![x]\!]$ we associate a commutative series $\mathsf{p2s}\,(f) \in \mathbb{Q}\langle\!\langle \Sigma \rangle\!\rangle$ by $\mathsf{p2s}\,(f) := \sum_{w \in \Sigma^*} [x^{\#(w)}]f \cdot w$. These two mappings are mutual inverses and by the following lemma we can identify CDF power series with commutative WBPP series, thus providing a bridge between the theory of weighted automata and differential equations.

▶ **Lemma 25.** *If $f \in \mathbb{Q}\langle\!\langle \Sigma \rangle\!\rangle$ is a commutative WBPP series, then $\mathsf{s2p}\,(f) \in \mathbb{Q}[\![x]\!]$ is a CDF power series. Conversely, if $f \in \mathbb{Q}[\![x]\!]$ is a CDF power series, then $\mathsf{p2s}\,(f) \in \mathbb{Q}\langle\!\langle \Sigma \rangle\!\rangle$ is a commutative WBPP series.*

3.4 Zeroness of CDF power series

Coefficient computation. We provide an algorithm to compute CDF power series coefficients. While a PSPACE algorithm follows from Theorem 13, we are interested here in the precise complexity w.r.t. degree, height, and order. This will allow us obtain the improved 2-EXPTIME complexity for zeroness (Theorem 3).

▶ **Lemma 26.** *Given a tuple of d-variate CDF power series $f \in \mathbb{Z}[\![x]\!]^k$ satisfying an integer system of CDF equations (9) of degree D, order k, height H, and a bound N, we can compute all coefficients $[x^n]f \in \mathbb{Z}^k$ with total degree $|n|_1 \leq N$ in deterministic time $\leq (N + d \cdot D + k)^{O(d \cdot D + k)} \cdot (\log H)^{O(1)}$.*

The lemma is proved by a dynamic programming algorithm storing all required coefficients in a table, which is feasible since numerators and denominators are not too big. This rough estimation shows that the complexity is exponential in d, D, k and *polynomial* in N.

Zeroness. The *zeroness problem* for CDF power series takes as input a polynomial $p \in \mathbb{Q}[y]$ and a system of equations (9) with an initial condition $c \in \mathbb{Q}^k$ extending to a (unique) power series series solution f s.t. $f(0) = c$, and asks whether $p \circ_y f = 0$.

▶ Remark 27. This is a *promise problem*: We do not decide solvability in power series. In our application in § 4 this is not an issue since power series solutions exist by construction. In the univariate case $d = 1$ the promise is always satisfied. We leave it as future work to investigate the problem of solvability in power series of CDF equations.

The following lemma gives short nonzeroness witnesses. It follows immediately from the WBPP ideal construction (6). Together with Lemma 26 it yields the announced Theorem 3.

▶ **Lemma 28.** *Consider a CDF $f \in \mathbb{Q}[\![x]\!]^k$ and $p \in \mathbb{Q}[y]$, both of degree $\leq D$. The power series $g := p \circ_y f$ is zero iff $[x^n]g = 0$ for all monomials x^n of total degree $|n|_1 \leq D^{k^{O(k^2)}}$.*

4 Constructible species of structures

The purpose of this section is to show how a rich combinatorial framework for building classes of finite structures (called *species*) gives rise in a principled way to a large class of CDF power series. The main result of this section is that multiplicity equivalence is decidable for a large class of species (Theorem 4). *Combinatorial species of structures* [45] are a formalisation of combinatorics based on category theory, designed in such a way as to expose a bridge between combinatorial operations on species and corresponding algebraic operations on power series. Formally, a *d-sorted species* is a d-ary endofunctor $\mathcal{F}$ in the category of finite sets and bijections. In particular, $\mathcal{F}$ defines a mapping from d-tuples of finite sets $U = (U_1, \ldots, U_d)$ to a finite set $\mathcal{F}[U]$, satisfying certain naturality conditions which ensure that $\mathcal{F}$ is independent of the names of the elements of U. In particular, the cardinality of the output $|\mathcal{F}[U_1, \ldots, U_d]|$ depends only on the cardinality of the inputs $|U_1|, \ldots, |U_d|$, which allows one to associate to $\mathcal{F}$ the *exponential generating series* (EGS) $\mathsf{EGS}[\mathcal{F}] := \sum_{n \in \mathbb{N}^d} \mathcal{F}_n \cdot \frac{x^n}{n!}$, where $\mathcal{F}_{n_1,\ldots,n_d} := |\mathcal{F}[U_1, \ldots, U_d]|$ for some (equivalently, all) finite sets of cardinalities $|U_1| = n_1, \ldots, |U_d| = n_d$. We refer to [61, Sec. 1] for an introduction to species tailored towards combinatorial enumeration (cf. also the book [5]). Below we present the main ingredients relevant for our purposes by means of examples.

Species can be built from basic species by applying species operations and solving species equations. Examples of *basic species* are the *zero* species $\mathbf{0}$ with EGS 0, the *one* species $\mathbf{1}$ with EGS 1, the *singleton* species $\mathcal{X}_j$ of sort j with EGS x_j, the *sets* species SET with EGS $e^x = 1 + x + x^2/2! + \cdots$ (since there is only one set of size n for each n), and the *cycles* species CYC with EGS $-\log(1 - x)$. New species can be obtained by the operations of *sum* (disjoint union) $\mathcal{F} + \mathcal{G}$, *combinatorial product* $\mathcal{F} \cdot \mathcal{G}$ (generalising the Cauchy product for words), *derivative* $\partial_{X_j}\mathcal{F}$ (cf. [61, Sec. 1.2 and 1.4] for formal definitions), and *cardinality restriction* $\mathcal{F}|_S$ (for a *cardinality constraint* $S \subseteq \mathbb{N}^d$). Regarding the latter, $\mathcal{F}|_S$ equals $\mathcal{F}$ on inputs $(U_1, \ldots, U_d)$ satisfying $(|U_1|, \ldots, |U_d|) \in S$, and is $\emptyset$ otherwise; we use the notation $\mathcal{F}_{\sim n}$ for the constraint $|U_1| + \cdots + |U_d| \sim n$, for $\sim$ a comparison operator such as $=$ or $\geq$.

Another important operation is that of composition of species [61, Sec. 1.5]. Consider sorts $\mathcal{X} = (\mathcal{X}_1, \ldots, \mathcal{X}_d)$ and $\mathcal{Y} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_k)$. Let $\mathcal{F}$ be a $(\mathcal{X}, \mathcal{Y})$-sorted species and let $\mathcal{G} = (\mathcal{G}_1, \ldots, \mathcal{G}_k)$ be a k-tuple of $\mathcal{X}$-sorted species. For a set of indices $I \subseteq \{1, \ldots, k\}$, we write $\mathcal{Y}_I$ for the tuple of those $\mathcal{Y}_i$'s s.t. $i \in I$. We say that $\mathcal{F}$ is *polynomial* w.r.t. $\mathcal{Y}_I$ if $\mathsf{EGS}[\mathcal{F}]$ is polynomial w.r.t. y_I, and similarly for *locally polynomial.* We say that $\mathcal{F}, \mathcal{G}$ are *$\mathcal{Y}$-composable* if $\mathcal{F}$ is locally polynomial w.r.t. $\mathcal{Y}_I$, where I is the set of indices i s.t. $\mathcal{G}_i[\emptyset] \neq \emptyset$. The notion of *strongly $\mathcal{Y}$-composable* is obtained by replacing "locally polynomial" with "polynomial". For two $\mathcal{Y}$-composable species $\mathcal{F}, \mathcal{G}$ their *composition* $\mathcal{F} \circ_{\mathcal{Y}} \mathcal{G}$ is a well-defined $\mathcal{X}$-sorted species. Informally, it is obtained by replacing each $\mathcal{Y}_i$ in $\mathcal{F}$ by $\mathcal{G}_i$.

We will not need the formal definitions of these operations, but we will use the fact that each of these has a corresponding operation on power series [5, Ch. 1]: $\mathsf{EGS}[\mathcal{F}+\mathcal{G}] = \mathsf{EGS}[\mathcal{F}] + \mathsf{EGS}[\mathcal{G}]$, $\mathsf{EGS}[\mathcal{F}\cdot\mathcal{G}] = \mathsf{EGS}[\mathcal{F}]\cdot\mathsf{EGS}[\mathcal{G}]$, $\mathsf{EGS}[\partial_{X_j}\mathcal{F}] = \partial_{x_j}\mathsf{EGS}[\mathcal{F}]$, $\mathsf{EGS}[\mathcal{F}\circ_{\mathcal{Y}}\mathcal{G}] = \mathsf{EGS}[\mathcal{F}]\circ_y \mathsf{EGS}[\mathcal{G}]$, and $\mathsf{EGS}[\mathcal{F}|_S] = \mathsf{EGS}[\mathcal{F}]|_S$. For instance, $\mathsf{SET}[\mathcal{X}]_{\geq 1}$ is the species of nonempty sets, with EGS $e^x - 1$; $\mathsf{SET}[\mathcal{X}]\cdot\mathsf{SET}[\mathcal{X}]$ is the species of *subsets* with EGS $e^x\cdot e^x = \sum_{n\in\mathbb{N}} 2^n\cdot x^n/n!$ since subsets correspond to partitions of a set into two parts and there are 2^n ways to do this for a set of size n; $\mathcal{X}\cdot\mathcal{X}$ is the species of pairs with EGS $2!\cdot x^2/2!$ since there are two ways to organise a set of size 2 into a pair; $\mathsf{SEQ}[\mathcal{X}] = 1+\mathcal{X}+\mathcal{X}\cdot\mathcal{X}+\cdots$ is the species of lists with EGS $(1-x)^{-1} = 1+x+x^2+\cdots$ since there are $n!$ ways to organise a set of size n into a tuple of n elements; $\mathsf{SET}[\mathcal{Y}]\circ_{\mathcal{Y}}\mathsf{SET}[\mathcal{X}]_{\geq 1}$ is the species of set partitions with EGS e^{e^x-1} since a set partition is a collection of nonempty sets which are pairwise disjoint and whose union is the whole set.

Finally, species can be defined as unique solutions of systems of species equations. E.g., the species of sequences $\mathsf{SEQ}[\mathcal{X}]$ is the unique species satisfying $\mathcal{Y} = \mathbf{1}+\mathcal{X}\cdot\mathcal{Y}$ since a nonempty sequence decomposes uniquely into a first element together with the sequence of the remaining elements; *binary trees* is the unique species solution of $\mathcal{Y} = \mathbf{1}+\mathcal{X}\cdot\mathcal{Y}^2$; *ordered trees* is the unique species solution of $\mathcal{Y} = \mathbf{1}+\mathcal{X}\cdot\mathsf{SEQ}[\mathcal{Y}]$; *Cayley trees* (rooted unordered trees) is the unique species satisfying $\mathcal{Y} = \mathcal{X}\cdot\mathsf{SET}[\mathcal{Y}]$ since a Cayley tree uniquely decomposes into a root together with a set of Cayley subtrees. For a more elaborate example, the species of *series-parallel graphs* is the unique solution for $\mathcal{Y}_1$ of the following system [61, Sec. 0]:

$$\begin{cases} \mathcal{Y}_1 &= \mathcal{X}+\mathcal{Y}_2+\mathcal{Y}_3, & \text{(sp graphs)}\\ \mathcal{Y}_2 &= \mathsf{SEQ}[\mathcal{X}+\mathcal{Y}_3]_{\geq 2}, & \text{(series graphs)}\\ \mathcal{Y}_3 &= \mathsf{SET}[\mathcal{X}+\mathcal{Y}_2]_{\geq 2}. & \text{(parallel graphs)} \end{cases} \tag{11}$$

Joyal's *implicit species theorem* [45] (cf. [61, Theorem 2.1], [5, Theorem 2 of Sec. 3.2]), which we now recall, provides conditions guaranteeing existence and uniqueness of solutions to species equations. Let a system of species equations $\mathcal{Y} = \mathcal{F}(\mathcal{X},\mathcal{Y})$ (with $\mathcal{F}$ a k-tuple of species) be *well posed* if $\mathcal{F}(\mathbf{0},\mathbf{0}) = \mathbf{0}$ and the Jacobian matrix $\partial_{\mathcal{Y}}\mathcal{F}$ (defined as for power series [61, Sec. 1.6]) is nilpotent at $(\mathbf{0},\mathbf{0})$. A *canonical solution* is a solution $\mathcal{Y} := \mathcal{G}(\mathcal{X})$ s.t. $\mathcal{G}(\mathbf{0}) = \mathbf{0}$.

▶ **Theorem 29** (Implicit species theorem [45]). *A well-posed system of species equations $\mathcal{Y} = \mathcal{F}(\mathcal{X},\mathcal{Y})$ admits a unique canonical solution $\mathcal{Y} := \mathcal{G}(\mathcal{X})$.*

The implicit species theorem is a direct analogue of the implicit function theorem for power series. Furthermore, if $\mathcal{Y} = \mathcal{F}(\mathcal{X},\mathcal{Y})$ is a well-posed system of species equations then $y = \mathsf{EGS}[\mathcal{F}](x,y)$ is a well-posed system of power series equations; moreover the EGS of the canonical species solution of the former is the canonical power series solution of the latter. We now have enough ingredients to define a large class of combinatorial species. *Strongly constructible species* are the smallest class of species (1) containing the basic species $\mathbf{0},\mathbf{1},\mathcal{X}_j$ $(j\in\mathbb{N}),\mathsf{SET},\mathsf{CYC}$; (2) closed under sum, product, strong composition, regular cardinality restrictions; and (3) closed under canonical resolution of well-posed systems $\mathcal{Y} = \mathcal{F}(\mathcal{X},\mathcal{Y})$ with $\mathcal{F}$ a tuple of strongly constructible species. Note that the equation $\mathcal{Y} = \mathbf{1}+\mathcal{X}\cdot\mathcal{Y}$ for sequences is not well posed, nonetheless sequences are strongly constructible: Nonempty sequences $\mathsf{SEQ}[\mathcal{X}]_{\geq 1}$ are the unique canonical solution of the well-posed species equation $\mathcal{Z} = \mathcal{X}+\mathcal{X}\cdot\mathcal{Z}$ and $\mathsf{SEQ}[\mathcal{X}] = \mathbf{1}+\mathsf{SEQ}[\mathcal{X}]_{\geq 1}$. Similar manipulations show that all the examples mentioned are strongly constructible.

▶ Remark 30. The class of strongly constructible species is incomparable with the class from [61, Definition 7.1]. On the one hand, [61] considers as cardinality restrictions only finite unions of intervals, while we allow general regular restrictions, e.g. periodic constraints such as "even size"; moreover, constraints in [61] are applied only to basic species, while we allow arbitrary strongly constructible species. On the other hand, we consider well-posed systems, while [61] considers more general *well-founded systems*. Finally, we consider strong composition, while [61] considers composition.

Since CDF power series include the the basic species EGS 0, 1, x_j ($j \in \mathbb{N}$), $(1-x)^{-1}$, e^x, and $-\log(1-x)$, from the CDF closure properties Lemmas 21, 23, and 24 and the discussion above, we have:

▶ **Theorem 31.** *The EGS of a strongly constructible species is effectively CDF.*

▶ Remark 32. *Constructible species* are obtained by considering composition instead of strong composition. We conjecture that even the EGS of constructible species are CDF, which would follow by generalising Lemma 21(3) from "strongly composable" to "composable".

For instance, the well-posed species equation $\mathcal{Y} = \mathcal{X} \cdot \mathsf{SET}[\mathcal{Y}]$ for Cayley trees translates to the well-posed power series equation $y = x \cdot e^y$ for its EGS. The well-posed species equations for series-parallel graphs (11) translate to the following well-posed power series equations for their EGS:

$$\begin{cases} y_1 = x + y_2 + y_3, \\ y_2 = \frac{1}{1-(x+y_3)} - 1 - (x + y_3), \\ y_3 = e^{x+y_2} - 1 - (x + y_2). \end{cases} \tag{12}$$

We conclude this section by deciding multiplicity equivalence of species. Two d-sorted species $\mathcal{F}, \mathcal{G}$ are *multiplicity equivalent* (*equipotent* [61]) if $\mathcal{F}_n = \mathcal{G}_n$ for every $n \in \mathbb{N}^d$. Decidability of multiplicity equivalence of strongly constructible species, announced in Theorem 4, follows from Theorems 3 and 31.

5 Conclusions

We have presented two related computation models, WBPP series and CDF power series. We have provided decision procedures of elementary complexity for their zeroness problems (Theorems 1 and 3), which are based on a novel analysis on the length of chains of polynomial ideals obtained by iterating a finite set of possibly noncommuting derivations (Theorem 18). On the way, we have developed the theory of WBPP and CDF, showing in particular that the latter arises as the commutative variant of the former. Finally, we have applied WBPP to the multiplicity equivalence of BPP (Corollary 2), and CDF to the multiplicity equivalence of constructible species (Theorem 4). Many directions are left for further work. Some were already mentioned in the previous sections. We highlight here some more.

Invariant ideal. Fix a WBPP (or CDF). Consider the *invariant ideal* of all configurations evaluating to zero $Z := \{\alpha \in \mathbb{Q}[N] \mid [\![\alpha]\!] = 0\}$. Zeroness is just membership in Z. Since Z is a polynomial ideal, it has a finite basis. The most pressing open problem is whether we can compute one such finite basis, perhaps leveraging on differential algebra [48]. Z is computable in the special case of WFA [38, 39], however for polynomial automata it is not [56].

Regular support restrictions. BPP languages are not closed under intersection with regular languages [21, proof of Proposition 3.11], and thus it is not clear for instance whether we can decide BPP multiplicity equivalence within a given regular language. We do not know whether WBPP series are closed under regular support restriction, and thus also zeroness of WBPP series within a regular language is an open problem.

WBPP with edge multiplicities. One can consider a slightly more expressive BPP model where one transition can remove more than one token from the same place [54]. It is conceivable that zeroness stays decidable, however a new complexity analysis is required since the corresponding ideal chains may fail to be convex.

References

1 Nikhil Balaji, Lorenzo Clemente, Klara Nosan, Mahsa Shirmohammadi, and James Worrell. Multiplicity problems on algebraic series and context-free grammars. In *Proc. of LICS'23*, pages 1–12, 2023. doi:10.1109/LICS56636.2023.10175707.

2 Henning Basold, Helle Hvid Hansen, Jean-Éric Pin, and Jan Rutten. Newton series, coinductively: a comparative study of composition. *Mathematical Structures in Computer Science*, 29(1):38–66, June 2017. doi:10.1017/s0960129517000159.

3 Michael Benedikt, Timothy Duff, Aditya Sharad, and James Worrell. Polynomial automata: Zeroness and applications. In *Proc. of LICS'17*, pages 1–12, June 2017. doi:10.1109/LICS.2017.8005101.

4 François Bergeron, Philippe Flajolet, and Bruno Salvy. Varieties of increasing trees. In J. C. Raoult, editor, *CAAP'92*, pages 24–48, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

5 François Bergeron, Gilbert Labelle, Pierre Leroux, and Margaret Readdy. *Combinatorial Species and Tree-like Structures*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1998.

6 François Bergeron and Christophe Reutenauer. Combinatorial resolution of systems of differential equations iii: A special class of differentially algebraic series. *European Journal of Combinatorics*, 11(6):501–512, 1990.

7 François Bergeron and Ulrike Sattler. Constructible differentially finite algebraic series in several variables. *Theoretical Computer Science*, 144(1):59–65, 1995.

8 J. Berstel and C. Reutenauer. *Noncommutative rational series with applications*. CUP, 2010.

9 Mikołaj Bojańczyk, Bartek Klin, and Joshua Moerman. Orbit-finite-dimensional vector spaces and weighted register automata. In *Proceedings of the 36th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '21. IEEE Press, 2021. doi:10.1109/LICS52264.2021.9470634.

10 Michele Boreale. Algebra, coalgebra, and minimization in polynomial differential equations. *Logical Methods in Computer Science*, Volume 15, Issue 1, February 2019.

11 Michele Boreale. Complete algorithms for algebraic strongest postconditions and weakest preconditions in polynomial odes. *Science of Computer Programming*, 193:102441, 2020.

12 Michele Boreale. Automatic pre- and postconditions for partial differential equations. *Information and Computation*, 285:104860, 2022.

13 Michele Boreale, Luisa Collodi, and Daniele Gorla. Products, polynomials and differential equations in the stream calculus. *ACM Trans. Comput. Logic*, 25(1), January 2024. doi:10.1145/3632747.

14 Michele Boreale and Daniele Gorla. Algebra and Coalgebra of Stream Products. In Serge Haddad and Daniele Varacca, editors, *32nd International Conference on Concurrency Theory (CONCUR 2021)*, volume 203 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2021.19.

15 Alin Bostan, Arnaud Carayol, Florent Koechlin, and Cyril Nicaud. Weakly-Unambiguous Parikh Automata and Their Link to Holonomic Series. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *Proc. of ICALP'20*, volume 168 of *LIPIcs*, pages 114:1–114:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2020.114.

16 Alin Bostan and Antonio Jiménez-Pastor. On the exponential generating function of labelled trees. *Comptes Rendus. Mathématique*, 358(9-10):1005–1009, 2020. doi:10.5802/crmath.108.

17 François Boulier, Daniel Lazard, François Ollivier, and Michel Petitot. Computing representations for radicals of finitely generated differential ideals. *Applicable Algebra in Engineering, Communication and Computing*, 20(1):73, 2009. doi:10.1007/s00200-009-0091-7.

18 Richard Büchi. Weak second-order arithmetic and finite automata. *Z. Math. Logik und grundl. Math.*, 6:66–92, 1960. doi:10.1002/malq.19600060105.

19 Alex Buna-Marginean, Vincent Cheval, Mahsa Shirmohammadi, and James Worrell. On learning polynomial recursive programs. *Proceedings of the ACM on Programming Languages*, 8(POPL):1001–1027, January 2024. doi:10.1145/3632876.

20 Michaël Cadilhac, Filip Mazowiecki, Charles Paperman, Michał Pilipczuk, and Géraud Sénizergues. On polynomial recursive sequences. *Theory of Computing Systems*, 2021. doi:10.1007/s00224-021-10046-9.

21 Søren Christensen. *Decidability and Decomposition in Process Algebras*. PhD thesis, Department of Computer Science, University of Edinburgh, 1993.

22 Søren Christensen, Yoram Hirshfeld, and Faron Moller. Bisimulation equivalence is decidable for basic parallel processes. In *CONCUR'93*, pages 143–157. Springer Berlin Heidelberg, 1993. doi:10.1007/3-540-57208-2_11.

23 Lorenzo Clemente. On the complexity of the universality and inclusion problems for unambiguous context-free grammars. In Laurent Fribourg and Matthias Heizmann, editors, Proceedings 8th International Workshop on *Verification and Program Transformation* and 7th Workshop on *Horn Clauses for Verification and Synthesis,* Dublin, Ireland, 25-26th April 2020, volume 320 of *EPTCS*, pages 29–43. Open Publishing Association, 2020. doi:10.4204/EPTCS.320.2.

24 Lorenzo Clemente. Weighted basic parallel processes and combinatorial enumeration. *arXiv e-prints*, page arXiv:2407.03638, July 2024. doi:10.48550/arXiv.2407.03638.

25 David A. Cox, John Little, and Donal O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer International Publishing, 4 edition, 2015.

26 Wojciech Czerwiński and Piotr Hofman. Language Inclusion for Boundedly-Ambiguous Vector Addition Systems Is Decidable. In Bartek Klin, Sławomir Lasota, and Anca Muscholl, editors, *33rd International Conference on Concurrency Theory (CONCUR 2022)*, volume 243 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:22, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CONCUR.2022.16.

27 Manfred Droste, Werner Kuich, and Heiko Vogler, editors. *Handbook of Weighted Automata*. Monographs in Theoretical Computer Science. Springer, 2009.

28 Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961. doi:doi:10.1090/S0002-9947-1961-0139530-9.

29 Javier Esparza. Petri nets, commutative context-free grammars, and basic parallel processes. *Fundamenta Informaticae*, 31(1):13–25, 1997. doi:10.3233/fi-1997-3112.

30 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

31 Michel Fliess. Sur divers produits de séries formelles. *Bulletin de la Société Mathématique de France*, 102:181–191, 1974. doi:10.24033/bsmf.1777.

32 Michel Fliess. Réalisation locale des systèmes non linéaires, algèbres de lie filtrées transitives et séries génératrices non commutatives. *Inventiones Mathematicae*, 71(3):521–537, March 1983. doi:10.1007/bf02095991.

33 Vojtěch Forejt, Petr Jančar, Stefan Kiefer, and James Worrell. Language equivalence of probabilistic pushdown automata. *Information and Computation*, 237:1–11, 2014. doi:10.1016/j.ic.2014.04.003.

34 Andrei Gabrielov and Nicolai Vorobjov. Complexity of computations with pfaffian and noetherian functions. In Y Ilyashenko and C Rousseau, editors, *Normal Forms, Bifurcations and Finiteness Problems in Differential Equations*, NATO Science Series II, page 211. Springer, January 2004.

35 Sheila A. Greibach. A new normal-form theorem for context-free phrase structure grammars. *Journal of the ACM*, 12(1):42–52, January 1965. doi:10.1145/321250.321254.

36 Yoram Hirshfeld. Petri nets and the equivalence problem. In Egon Börger, Yuri Gurevich, and Karl Meinke, editors, *Computer Science Logic*, pages 165–174, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

37 John Hopcroft, Rajeev Motwani, and Jeffrey Ullman. *Introduction to Automata Theory, Languages, and Computation.* Addison-Wesley, 2000.

38 Ehud Hrushovski, Joël Ouaknine, Amaury Pouly, and James Worrell. Polynomial invariants for affine programs. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 530–539, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3209108.3209142.

39 Ehud Hrushovski, Joël Ouaknine, Amaury Pouly, and James Worrell. On strongest algebraic program invariants. *J. ACM*, August 2023. Just Accepted. doi:10.1145/3614319.

40 Hans Hüttel. Undecidable equivalences for basic parallel processes. In *Theoretical Aspects of Computer Software. TACS 1994*, pages 454–464. Springer Berlin Heidelberg, 1994. doi:10.1007/3-540-57887-0_110.

41 Hans Hüttel, Naoki Kobayashi, and Takashi Suto. Undecidable equivalences for basic parallel processes. *Information and Computation*, 207(7):812–829, July 2009. doi:10.1016/j.ic.2008.12.011.

42 Petr Jančar. Nonprimitive recursive complexity and undecidability for petri net equivalences. *Theoretical Computer Science*, 256(1):23–30, 2001. ISS. doi:10.1016/S0304-3975(00)00100-6.

43 Petr Jančar. Strong bisimilarity on basic parallel processes in PSPACE-complete. In *Proc. of LICS'03*, pages 218–227, 2003. doi:10.1109/LICS.2003.1210061.

44 Johannes Mittmann. *Independence in Algebraic Complexity Theory.* PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, December 2013. URL: https://hdl.handle.net/20.500.11811/5810.

45 André Joyal. Une théorie combinatoire des séries formelles. *Advances in Mathematics*, 42(1):1–82, 1981.

46 Manuel Kauers and Peter Paule. *The Concrete Tetrahedron: Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates.* Texts and Monographs in Symbolic Computation. Springer-Verlag Wien, 1 edition, 2011.

47 S. C. Kleene. Representation of events in nerve nets and finite automata. In Shannon and Mccarthy, editors, *Automata Studies*, pages 3–41. Princeton Univ. Press, 1956. URL: http://www.rand.org/pubs/research_memoranda/RM704.html.

48 E. R. Kolchin. *Differential Algebra and Algebraic Groups.* Pure and Applied Mathematics 54. Academic Press, Elsevier, 1973.

49 Pierre Leroux and Gérard X. Viennot. Combinatorial resolution of systems of differential equations, i. ordinary differential equations. In Gilbert Labelle and Pierre Leroux, editors, *Combinatoire énumérative*, pages 210–245, Berlin, Heidelberg, 1986. Springer Berlin Heidelberg. doi:10.1007/BFb0072518.

50 L Lipshitz. The diagonal of a d-finite power series is d-finite. *Journal of Algebra*, 113(2):373–378, 1988. doi:10.1016/0021-8693(88)90166-4.

51 Leonard Lipshitz. D-finite power series. *Journal of Algebra*, 122(2):353–373, 1989. doi:10.1016/0021-8693(89)90222-6.

52 Prince Mathew, Vincent Penelle, Prakash Saivasan, and A.V. Sreejith. Weighted One-Deterministic-Counter Automata. In Patricia Bouyer and Srikanth Srinivasan, editors, *Proc. of FSTTCS'23*, volume 284 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 39:1–39:23, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2023.39.

53 Ernst Mayr. Membership in polynomial ideals over q is exponential space complete. In B. Monien and R. Cori, editors, *In Proc. of STACS'89*, pages 400–406, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg. doi:10.1007/BFb0029002.

54 Ernst W. Mayr and Jeremias Weihmann. *Completeness Results for Generalized Communication-Free Petri Nets with Arbitrary Edge Multiplicities*, pages 209–221. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-41036-9_19.

55 Robin Milner. *A calculus of communicating systems*. Lecture Notes in Computer Science 92. Springer-Verlag Berlin Heidelberg, 1 edition, 1980.

56 Julian Müllner, Marcel Moosbrugger, and Laura Kovács. Strong Invariants Are Hard: On the Hardness of Strongest Polynomial Invariants for (Probabilistic) Programs. *arXiv e-prints*, page arXiv:2307.10902, July 2023. doi:10.48550/arXiv.2307.10902.

57 Filip Murlak, Damian Niwiński, and Wojciech Rytter, editors. *200 Problems on Languages, Automata, and Computation*. Cambridge University Press, March 2023. doi:10.1017/9781009072632.

58 Masakazu Nasu and Namio Honda. Mappings induced by pgsm-mappings and some recursively unsolvable problems of finite probabilistic automata. *Information and Control*, 15(3):250–273, September 1969. doi:10.1016/s0019-9958(69)90449-5.

59 Dmitri Novikov and Sergei Yakovenko. Trajectories of polynomial vector fields and ascending chains of polynomial ideals. *Annales de l'Institut Fourier*, 49(2):563–609, 1999.

60 Azaria Paz. *Introduction to Probabilistic Automata*. Computer Science and Applied Mathematics. Elsevier Inc, Academic Press Inc, 1971.

61 Carine Pivoteau, Bruno Salvy, and Michèle Soria. Algorithms for combinatorial structures: Well-founded systems and Newton iterations. *Journal of Combinatorial Theory, Series A*, 119(8):1711–1773, 2012.

62 André Platzer. *Logical Foundations of Cyber-Physical Systems*. Springer International Publishing, 1st ed. edition, 2018.

63 André Platzer and Yong Kiam Tan. Differential equation invariance axiomatization. *J. ACM*, 67(1), April 2020.

64 Michael O. Rabin and Dana Scott. Finite automata and their decision problems. *IBM J. Res. Dev.*, 3(2):114–125, April 1959. doi:10.1147/rd.32.0114.

65 Antonio Restivo and Christophe Reutenauer. On cancellation properties of languages which are supports of rational power series. *J. Comput. Syst. Sci.*, 29(2):153–159, October 1984. doi:10.1016/0022-0000(84)90026-6.

66 Christophe Reutenauer. *The Local Realization of Generating Series of Finite Lie Rank*, pages 33–43. Springer Netherlands, 1986. doi:10.1007/978-94-009-4706-1_2.

67 Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4(2):177–192, 1970. doi:10.1016/S0022-0000(70)80006-X.

68 Marcel Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2–3):245–270, September 1961. doi:10.1016/s0019-9958(61)80020-x.

69 A. Seidenberg. Constructions in algebra. *Transactions of the American Mathematical Society*, 197:273–313, 1974. doi:10.2307/1996938.

70 Jiří Srba. Strong bisimilarity and regularity of basic parallel processes is PSPACE-hard. In *STACS 2002*, pages 535–546. Springer Berlin Heidelberg, 2002. doi:10.1007/3-540-45841-7_44.

71 R. P. Stanley. Differentiably finite power series. *European Journal of Combinatorics*, 1(2):175–188, 1980. doi:10.1016/S0195-6698(80)80051-5.

72 Richard Stanley. *Enumerative combinatorics*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, 2ed edition, 2011.

73 R. E. Stearns and H. B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM Journal on Computing*, 14(3):598–611, August 1985. `doi:10.1137/0214044`.

74 B. A. Trakhtenbrot. Finite automata and the logic of one-place predicates. *Siberian Math. J.*, 1962.

75 A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937. `doi:10.1112/plms/s2-42.1.230`.

76 Wen-Guey Tzeng. A polynomial-time algorithm for the equivalence of probabilistic automata. *SIAM J. Comput.*, 21(2):216–227, April 1992. `doi:10.1137/0221017`.

77 Joris van der Hoeven and John Shackell. Complexity bounds for zero-test algorithms. *Journal of Symbolic Computation*, 41(9):1004–1020, 2006.