Differential Tree Automata

Rida Ait El Manssour
University of Oxford, UK

Vincent Cheval
University of Oxford, UK

Mahsa Shirmohammadi
CNRS & IRIF, Paris, France

James Worrell
University of Oxford, UK

Abstract

In this paper we introduce the notion of a differential tree automaton. Differential tree automata generalise weighted tree automata (over a field) by allowing the transition weights to be rational functions of the tree size. Whereas the class of generating functions of weighted tree automata coincides with the class of algebraic power series, our main result is that the class of generating functions of differential tree automata coincides with the class of differentially algebraic power series. As a corollary, we obtain a decision procedure for determining equivalence of differential tree automata. In the course of proving our main result we identify a class of recurrences that characterises the sequence of coefficients of a differentially algebraic power series, generalising Reutenauer's matrix representation of polynomially recursive sequences. We further identify a natural syntactic subset of differential tree automata whose generating functions are given by rational dynamical systems, that is, as components of the solution of a system of differential equations $\boldsymbol{y}' = F(\boldsymbol{y})$, where F is a vector of rational functions that is defined at $\boldsymbol{y}(0)$. We further show that this class of power series can be characterised in terms of the classical notion of weighted tree automata by using a labelled generating function on trees.

2012 ACM Subject Classification Theory of computation → Tree languages; Theory of computation → Quantitative automata; Mathematics of computing → Generating functions

Keywords and phrases Combinatorial species, Differentially algebraic series, Weighted automata

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.70

Related Version *Full Version*: `https://arxiv.org/abs/2407.08218` [22]

Funding *Rida Ait El Manssour*: Supported by the EPSRC Fellowship (EP/X033813/1).
Vincent Cheval: Supported by the ERC Synergy Grant VePaSS (agreement No. 101224640).
Mahsa Shirmohammadi: Supported by the ERC Synergy Grant VePaSS (agreement No. 101224640) and the grant VeSyAM (ANR-22-CE48-0005).
James Worrell: Supported by the EPSRC Fellowship (EP/X033813/1).

1 Introduction

1.1 Combinatorial Classes, Automata, and Power Series

Many combinatorial classes admit natural encodings as languages of words or trees. For example, by giving a bijection between triangulations of polygons and Dyck words, one can show that the number of triangulations of a given size is determined by the sequence of Catalan numbers (see [28] for many other examples of this type). This connection also provides a method for systematically extracting generating functions for combinatorial classes from their specifications and leads to a natural correspondence between classes of automata and classes of power series. For example, regular languages of words have rational

© Rida Ait El Manssour, Vincent Cheval, Mahsa Shirmohammadi, and James Worrell;
licensed under Creative Commons License CC-BY 4.0

generating functions [15, Section I.4], while regular tree languages have algebraic generating functions [15, Sections I.5]. Here, recall that for a field $\mathbb{K}$ a formal power series $f \in \mathbb{K}[\![x]\!]$ is *rational* if there are polynomials $P, Q \in \mathbb{K}[x]$ such that $f = \frac{P}{Q}$ and f is *algebraic* if there are polynomials $P_0, \ldots, P_d \in \mathbb{K}[x]$, not all zero, such that $\sum_{i=0}^{d} P_i f^i = 0$.

Independent of connections with combinatorics, the notion of generating series has played an influential role in formal language and automata theory. The celebrated theorem of Chomsky and Schützenberger showed that unambiguous context-free languages have algebraic generating functions (see, e.g., [26, Chapter IV] and [21, Chapters II–III]). This fact was exploited by Flajolet [15] to establish the inherent unambiguity of certain context-free languages, by showing that their generating functions are transcendental. By translating the question of ambiguity to a question about power series, Flajolet was able to use a powerful arsenal of analytic and number-theoretic techniques.

Despite these results, it remains an open problem to characterise the class of power series arising as generating functions of arbitrary context-free languages. A closely related open question concerns the multiplicity equivalence problem for context-free grammars. While decidability is known in the unary case, by a reduction to the existential theory of real-closed fields using generating series, no general solution is known, and the complexity of the unary case has only recently been sharpened, again using generating series and their approximation in the valued field of power series [1].

In this paper, we further develop the correspondence between automata, power series, and combinatorial classes, to relate tree automata and differentially algebraic power series. Our main contribution is to define the class of *differential tree automata* and to show that their generating functions are precisely the differentially algebraic power series. Recall here that a formal power series $f \in \mathbb{K}[\![x]\!]$ is *differentially algebraic* if it satisfies a differential equation $P(f, f', \ldots, f^{(k)}) = 0$ for some polynomial $P(y_1, \ldots, y_k)$ with coefficients in $\mathbb{K}[x]$. We also give a combinatorial characterisation of a proper subclass of differentially algebraic power series, which we call *rationally dynamically algebraic (RDA) series.* These series have generating functions that are given by rational dynamical systems, that is, as components of the solution of a system of first-order differential equations

$$y_1' = F_1(y_1, \ldots, y_k), \; \ldots, \; y_k' = F_k(y_1, \ldots, y_k), \tag{1}$$

where $F_1, \ldots, F_k \in \mathbb{K}(x, y_1, \ldots, y_k)$ are rational functions [17, 23], that are defined at $(0, y_1(0), \ldots, y_k(0))$. In [22, Appenidx H], which is the full version of this paper, we give a collection of rational dynamical systems that characterise the exponential generating functions of standard constructions of combinatorial species.

1.2 Tree Automata and their Generating Functions

Our main results concern the notion of *differential tree automata*: a generalisation of weighted tree automata [18]. Let Σ be a finite ranked alphabet and write T_Σ for the set of Σ-trees. As with a $\mathbb{K}$-weighted automaton, a differential tree automaton $\mathcal{A}$ computes a function $[\![\mathcal{A}]\!] : T_\Sigma \to \mathbb{K}$. Recall that a transition in a $\mathbb{K}$-weighted tree automaton has the form

$$\sigma(q_1, \ldots, q_\ell) \xrightarrow{a} q \,,$$

where σ is a ℓ-ary alphabet symbol, $q_1, \ldots, q_\ell$ and q are states of the automaton, and $a \in \mathbb{K}$ is the weight of the transition. Intuitively, the automaton reads the symbol σ and makes a transition from the tuple of states $q_1, \ldots, q_\ell$ to q with weight a. Generalising this pattern, in a differential tree automaton the weight of a transition is a rational function $R(x)$ whose denominator has no positive integer roots. When instantiated on a tree t, the transition has weight $R(\|t\|)$, where $\|t\|$ denotes the size of t (only counting the number of internal nodes in t). By the above assumption on the denominator of $R(x)$, $R(\|t\|)$ is well-defined.

We consider two different generating functions associated with (weighted and differential) tree automata. The *ordinary generating function* of an automaton $\mathcal{A}$ is the power series $f_{\mathcal{A}} \in \mathbb{K}[\![x]\!]$ defined by

$$f_{\mathcal{A}}(x) = \sum_{n\in\mathbb{N}} \left(\sum_{\substack{t\in T_\Sigma \\ \|t\|=n}} [\![\mathcal{A}]\!](t) \right) x^n. \tag{2}$$

A formulation of the Chomsky-Schützenberger theorem is that the class of algebraic power series in $\mathbb{K}[\![x]\!]$ coincides with the class of ordinary generating functions of tree series recognized by $\mathbb{K}$-weighted tree automata [13, Section 8].

We also introduce the notion of a *labelled generating function*. A *monotonic labelling* of a tree $t \in T_\Sigma$ of size n is a bijective mapping from the set of internal nodes of t to the set $\{1, \ldots, n\}$ such that the label of any node is greater than the label of any of its children. Define $\lambda : T_\Sigma \to \mathbb{Q}$ such that $\lambda(t)$ is the proportion of labellings of $t \in T_\Sigma$ that are monotonic. That is, the number of monotonic labellings of t is $\lambda(t)n!$. The labelled generating function of $\mathcal{A}$ is the formal power series $\widetilde{f}_{\mathcal{A}} \in \mathbb{K}[\![x]\!]$ defined by

$$\widetilde{f}_{\mathcal{A}}(x) := \sum_{n\in\mathbb{N}} \left(\sum_{\substack{t\in T_\Sigma \\ \|t\|=n}} \lambda(t)[\![\mathcal{A}]\!](t) \right) x^n. \tag{3}$$

1.3 Main Results

Our first result is as follows:

▶ **Theorem 1.** *The following are equivalent for a power series $g(x) = \sum_{n=0}^{\infty} a_n x^n$ in $\mathbb{K}[\![x]\!]$:*
1. *g is rationally dynamically algebraic;*
2. *g is equal to the ordinary generating function of a differential tree automaton in which the rational weight of every transition has the form $\frac{ax+b}{x}$ for some $a, b \in \mathbb{K}$;*
3. *g is equal to the labelled generating function of a $\mathbb{K}$-weighted tree automaton.*

From Theorem 1 it follows that every RDA power series $g \in \mathbb{Z}[\![x]\!]$ can be written $g = \widetilde{f}_{\mathcal{A}} - \widetilde{f}_{\mathcal{B}}$, where $\mathcal{A}$ and $\mathcal{B}$ are deterministic bottom-up tree automata.

Removing the restriction on the form of rational function in Item 2 of Theorem 1, we show that the ordinary generating functions of differential tree automata are precisely the differentially algebraic series.

▶ **Theorem 2.** *The following are equivalent for a power series $g(x) = \sum_{n=0}^{\infty} a_n x^n$ in $\mathbb{K}[\![x]\!]$:*
1. *g is differentially algebraic;*
2. *g is equal to the ordinary generating function of a differential tree automaton;*
3. *$(a_n)_{n=0}^{\infty}$ is a component of a sequence $(\boldsymbol{b}_n)_{n=0}^{\infty} \in \mathbb{K}^d$ of vectors that satisfies a bilinear recurrence with polynomial coefficients, of the form*

$$\boldsymbol{b}_n = \boldsymbol{b}_{n-1}\, P(n) + \sum_{k=0}^{n-1} (\boldsymbol{b}_k \otimes \boldsymbol{b}_{n-k-1}) Q(n)\,,$$

where $P \in \mathbb{K}(x)^{d\times d}$ and $Q \in \mathbb{K}(x)^{d^2\times d}$ have entries defined on positive integers.

The proof of Theorem 2 builds on a construction of Denef and Lipschitz [12], who showed that the sequence of coefficients $(a_n)_{n=0}^{\infty}$ of a differentially algebraic power series satisfies a recurrence of the form

$$P(n)\,a_n = Q_n(a_0, \dots, a_{n-1})\,,$$

where Q_n is a polynomial whose form depends on n. Working on the construction of [12], we show that $(a_n)_{n=0}^{\infty}$ can be recovered as a component of vectors satisfying a recurrence of the form shown in Item 3 of Theorem 2.

From the equivalence of Item 1 and Item 2 in Theorem 2, we obtain a decision procedure for equivalence of differential tree automata (Theorem 18). The detailed proof of our main results can be found in the longer version of the paper at [22]. We further show in [22, Theorem 32 in Appendix G] that the representation of differentially algebraic power series by recurrences in Items 2 and 3 is convenient for computing the sum, Cauchy product, inverse, derivative, integration, and forward and backward shift of such series.

1.4 Related Work

The special case of RDA series in which the functions $F_1, \dots, F_k$ in (1) are polynomials is studied in [3] under the name *constructibly differentiably algebraic (CDA)* series. We observe below that the classes of RDA and CDA series coincide. A combinatorial interpretation of CDA power series via so-called *tree labelling tables*, which are an automaton-like notion on labelled non-plane trees, is given in [3, Theorem 1]. This result is closely related to Item 3 of Theorem 1.

A power series $f \in \mathbb{K}[\![x]\!]$ is *differentially finite* (D-finite) if it satisfies an equation $\sum_{i=0}^{d} P_i(x) f^{(i)}(x) = 0$ for polynomials $P_0, \dots, P_d \in \mathbb{K}[x]$. Thus differentially finite power series are a subclass of differentially algebraic series. It is well-known [19] that $f(x) = \sum_{n=0}^{\infty} a_n x^n$ is differentially finite if and only if its sequence $(a_n)_{n=0}^{\infty}$ of coefficients satisfies a linear recurrence with polynomial coefficients. Reutenauer [25] shows that such sequences can equivalently be characterised by vector recurrences corresponding to a special case of the recurrence in Item 3 of Theorem 2 in which Q is identically zero. See [8] for a development of these ideas, including an exact learning algorithm for such recurrences and their automata extensions.

Krattenhaler and Rivoal [20] develop recurrences for differentially algebraic power series of a form related to Item 3 of Theorem 2, which they use to analyse divisibility properties of the coefficients.

Clemente [10] uses techniques from differential algebra to show that multiplicity equivalence of *basic parallel processes* can be decided in doubly exponential space. He also shows that commutative weighted basic parallel processes (viewed as power series) coincide with the class of CDA series, thereby obtaining a doubly exponential time procedure for deciding zeroness of CDA series.

Boreale [5] defines a notion of bisimulation for systems of ordinary differential equations, based on the Lie derivative. He gives an algorithm to compute the largest bisimulation on a given system, and shows how this can be used for minimisation and determining equivalence. As observed in [10], this procedure decides zeroness of CDA power series; however no complexity analysis is given in [5].

Resolving a conjecture of Castiglione and Massazza [9], Bostan *et al.* [6] show that the multivariate power series arising as the generating functions of languages accepted by weakly-unambiguous Parikh automata are D-finite. This result is used to prove the inherent unambiguity of certain languages based on analytic properties of univariate D-finite series considered as functions.

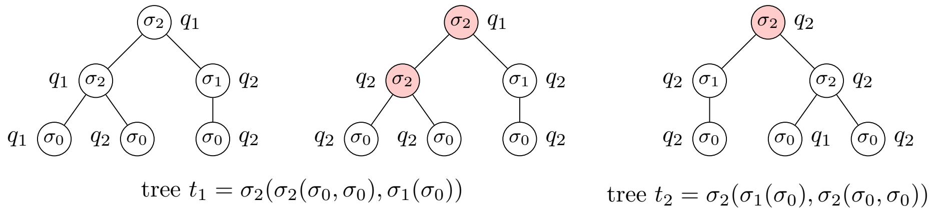

Figure 1 Trees in Example 3, where Bell numbers arise as the labelled generating function of the tree automaton.

We focus on automata in this paper. Other works consider the relationship between recurrences and generating functions on the one hand, and combinatorial classes defined by logical formulas on the other hand [2, 14].

2 Overview

2.1 Tree Automata

Let Σ be a ranked alphabet, where each symbol $\sigma \in \Sigma$ is assigned a non-negative integer arity. For $k \in \mathbb{N}$, we write Σ_k for the set of symbols of arity k. A (non-deterministic) tree automaton is a tuple $\mathcal{A} = (Q, \Delta, F)$, where Q is a finite set of states, $F \subseteq Q$ is a set of accepting states, and Δ is a set of transition rules of the form

$$\sigma(q_1, \dots, q_k) \to q,$$

with $\sigma \in \Sigma_k$ and $q_1, \dots, q_k, q \in Q$ [11].

The automaton processes a tree bottom-up, starting at the leaves. Leaf nodes are assigned states according to the rules in Δ for nullary symbols, that is, the states are chosen among all the states q that appear in a rule $\sigma \longrightarrow q$ in Δ. States are then propagated upwards using the transition rules, combining the states of child nodes to determine the state of their parent. In other words, a node labeled by σ is assigned the state q when its children are assigned the states $q_1, \dots, q_k$ in left-to-right order and $\sigma(q_1, \dots, q_k) \longrightarrow q$ is a transition rule in Δ.

A *run* of the automaton on a tree $t \in T_\Sigma$ corresponds to one state assignment of all nodes in t according to Δ. For example, when $\sigma \longrightarrow q_1$ and $\sigma \longrightarrow q_2$ are in Δ, the same tree may yield different runs of the automaton. The value $[\![\mathcal{A}]\!](t)$ of a tree $t \in T_\Sigma$ is the number of runs on t in which the root is assigned with a state in F. We say that a tree t is *accepted* by the automaton if it admits at least one valid run, that is $[\![\mathcal{A}]\!](t) > 0$.

The *ordinary generating function* of an automaton $\mathcal{A}$ is a univariate power series $\sum_{n=0}^{\infty} a_n x^n$ whose n-th coefficient a_n is the sum of $[\![\mathcal{A}]\!](t)$ over all trees of size n (as described in Equation (2)). Similarly, the n-th coefficient of the *labelled generating function* power series is the number of monotonic labelled trees accepted by the automaton divided by $n!$ (see Equation 3).

▶ **Example 3** (Partitions)**.** Recall the sequence

$$(b_n)_{n=0}^{\infty} = (1, 1, 2, 5, 15, 52, 203, \dots)$$

of Bell numbers, where b_n is the number of partitions of the set $\{1, \dots, n\}$. As noted in [3], the exponential generating function

$$f_B(x) := \sum_{n=0}^{\infty} \frac{b_n}{n!} x^n$$

is (the y_1-component of) a solution of the system of differential equations

$$y_1' = y_1 y_2, \quad y_2' = y_2.$$

Here we show that f_B arises as the labelled generating function of a tree automaton.

Let $\Sigma = \{\sigma_0, \sigma_1, \sigma_2\}$ be a ranked alphabet such that σ_0, σ_1 and σ_2 have respective arities 0, 1 and 2. Let the regular language $L \subseteq T_\Sigma$ comprise all trees t accepted by the automaton $\mathcal{A}$ with set of states $\{q_1, q_2\}$, where q_1 is accepting, with transitions

$$\sigma_0 \longrightarrow q_1 \qquad \sigma_0 \longrightarrow q_2 \qquad \sigma_1(q_2) \longrightarrow q_2 \qquad \sigma_2(q_1, q_2) \longrightarrow q_1$$

Consider the trees t_1 and t_2 depicted in Figure 1 with different state assignments. The state assignment of t_1 on the left is a run of $\mathcal{A}$, since it is consistent with Δ. However, the second state assignment of t_1 and that of t_2 are not runs of the automaton (the node colored in red violates Δ). In particular, the tree t_2 is not accepted by $\mathcal{A}$, since it admits no valid run. Moreover, this automaton is unambiguous: for every accepting tree t, its $[\![\mathcal{A}]\!](t)$ is always 1.

From a combinatorial perspective, a tree $t \in L$ can be considered as a list of chains, where each chain (depicted as a coloured block in Figure 2), consists of a set of nodes all having a common ancestor under the right-child relation. Each monotone labelling of a tree $t \in L$ thus determines a unique partition of $\{1, \ldots, n\}$: the components of the partition are the sets of labels of each chain in t. Moreover every partition arises as the labelling of a unique tree $t \in L$. We thus have

$$\sum_{t \in L,\, \|t\| = n} \lambda(t) [\![\mathcal{A}]\!](t) = \sum_{t \in L,\, \|t\| = n} \lambda(t) = \frac{b_n}{n!}$$

and we recover the series f_B as the labelled generating function $\widetilde{f}_{\mathcal{A}}$ of the automaton $\mathcal{A}$. ◁

2.2 Weighted Tree Automata

A $\mathbb{K}$-*weighted tree automaton* $\mathcal{A}$ (also called a multilinear representation [4]) is a bottom-up tree automaton (Q, Δ, q_f), with a single final state $q_f \in Q$, where the transition rules in Δ are equipped with a weight in $\mathbb{K}$. Transitions are denoted

$$\sigma(q_1, q_2, \ldots, q_k) \xrightarrow{a} q,$$

where $\sigma \in \Sigma_k$, $a \in \mathbb{K}$ and $q_1, \ldots, q_k, q \in Q$. A run of such an automaton is a run of the underlying non-deterministic tree automaton. The weight of the run is the product of the weights of the transitions comprising the run. The value $[\![\mathcal{A}]\!](t)$ of tree $t \in T_\Sigma$ is the sum of the weights of all accepting runs on t.

When reasoning with weighted tree automata it is convenient to work with a matricial presentation in which a $\mathbb{K}$-weighted tree automaton over alphabet Σ is a pair (d, μ), where $d \in \mathbb{N}$ is the dimension and μ is a function with domain Σ such that $\mu(\sigma) \in \mathbb{K}^{d^k \times d}$ is a matrix for $\sigma \in \Sigma_k$. The dimension corresponds to the number of states in the automaton, whereas μ is the matrix representation of the set of transition rules Δ with the set of states $Q = \{1, \ldots, d\}$:

$$\sigma(q_1, \ldots, q_k) \xrightarrow{a} q \in \Delta \quad \text{iff} \quad \mu(\sigma)_{(q_1, \ldots, q_k), q} = a$$

The value $[\![\mathcal{A}]\!](t)$ of $t \in T_\Sigma$ is defined using the Kronecker product. Recall that the Kronecker product of two row vectors $\boldsymbol{u} = \begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix}$ and $\boldsymbol{v} = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}$, denoted $\boldsymbol{u} \otimes \boldsymbol{v}$, is the mn-dimensional row vector defined by

$$\begin{bmatrix} u_1 v_1 & u_1 v_2 & \cdots & u_1 v_n & u_2 v_1 & \cdots & u_m v_{n-1} & u_m v_n \end{bmatrix}.$$

The operation is associative, and by convention, the nullary Kronecker product is the 1×1 identity matrix.

The map μ induces a function $\boldsymbol{\mu} : T_\Sigma \to \mathbb{K}^{1\times d}$ that is defined inductively by

$$\boldsymbol{\mu}(\sigma(t_1,\ldots,t_k)) := (\boldsymbol{\mu}(t_1) \otimes \cdots \otimes \boldsymbol{\mu}(t_k)) \cdot \mu(\sigma)$$

for a k-ary symbol σ. The operator $\cdot$ denotes standard matrix multiplication and is sometimes omitted when clear from context. In other words, $\boldsymbol{\mu}(t)$ is given by an iterated matrix product determined by parsing the tree t from the leaves to the root. From the properties of the Kronecker product, $\boldsymbol{\mu}(t)$ is a vector $\begin{bmatrix} w_1 & \ldots & w_d \end{bmatrix}$ where w_i is the sum of the weights of all runs over t that label the root with state i. Without loss of generality, we will consider that the final state q_f corresponds to index 1, meaning that $[\![\mathcal{A}]\!](t) = \mu(t)_1$. By convention, in this paper, a vector will be represented by a bold symbol, say $\boldsymbol{v}$, and its coordinate will be denoted $v_1, v_2, \ldots$.

▶ **Example 4.** The automaton given in Example 3 can be interpreted as a weighted tree automaton with unit weights on all transitions.

$$\sigma_0 \xrightarrow{1} q_1 \qquad \sigma_0 \xrightarrow{1} q_2 \qquad \sigma_1(q_2) \xrightarrow{1} q_2 \qquad \sigma_2(q_1,q_2) \xrightarrow{1} q_1$$

The matrix representation of this weighted automaton is $(2,\mu)$ where $\mu(\sigma_0) = \begin{bmatrix} 1 & 1 \end{bmatrix}$,

$$\mu(\sigma_1) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \qquad \text{and} \qquad \mu(\sigma_2) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

where the entry $\mu(\sigma_2)_{(q_i,q_j),q_k}$ is located in the $2(i-1)+j$-th row and k-th column. The vectors $\boldsymbol{\mu}(t_1)$ and $\boldsymbol{\mu}(t_2)$ are computed by the vectors $\boldsymbol{\mu}$ of their subtrees bottom-up with $\boldsymbol{\mu}(\sigma_0) = \begin{bmatrix} 1 & 1 \end{bmatrix}$ and

$$\boldsymbol{\mu}(\sigma_1(\sigma_0)) = \boldsymbol{\mu}(\sigma_0) \cdot \mu(\sigma_1) = \begin{bmatrix} 1 & 1 \end{bmatrix} \cdot \mu(\sigma_1) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \end{bmatrix}$$

$$\boldsymbol{\mu}(\sigma_2(\sigma_0,\sigma_0)) = (\boldsymbol{\mu}(\sigma_0) \otimes \boldsymbol{\mu}(\sigma_0)) \cdot \mu(\sigma_2) = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \cdot \mu(\sigma_2) = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

giving that

$$\boldsymbol{\mu}(t_1) = (\boldsymbol{\mu}(\sigma_2(\sigma_0,\sigma_0)) \otimes \boldsymbol{\mu}(\sigma_1(\sigma_0))) \cdot \mu(\sigma_2) = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \cdot \mu(\sigma_2) = \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad \triangleleft$$

The generating function of a weighted tree automaton, given by Equation (2), is an algebraic power series [4, Proposition 7.2]. Note that two automata that represent different tree series can have the same generating function, since the latter aggregates all trees of the same size.

2.3 Differential Tree Automata

The notion of differential tree automata generalises that of weighted tree automata. A $\mathbb{K}$-*differential tree automaton* over alphabet Σ is a pair $\mathcal{A} = (d,\mu)$, where $d \in \mathbb{N}$ is the dimension and μ is a map with domain Σ such that

- $\mu(\sigma) \in \mathbb{K}^{1\times d}$ for $\sigma \in \Sigma_0$ and,
- $\mu(\sigma) \in \mathbb{K}(x)^{d^\ell \times d}$ is a matrix of rational functions, having no positive integer poles, for all $\ell \geq 1$ and $\sigma \in \Sigma_\ell$.

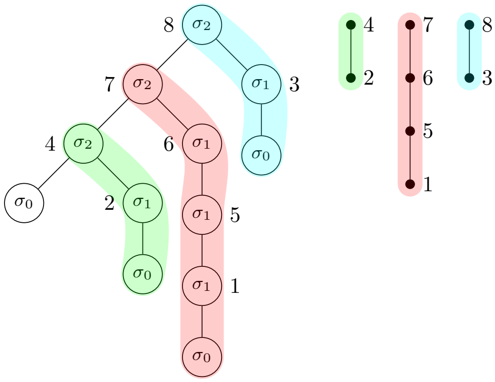

Figure 2 The ranked tree with monotone labelling, on the left, encodes the partition of $\{1, \ldots, 8\}$ on the right.

The map μ induces a function $\boldsymbol{\mu} : T_\Sigma \to \mathbb{K}^{1\times d}$, inductively defined by specifying that for all $\sigma \in \Sigma_0$, $\boldsymbol{\mu}(\sigma) := \mu(\sigma)$; and for all $k \geq 1$ and $\sigma \in \Sigma_k$ and $t_0 = \sigma(t_1, \ldots, t_k)$,

$$\boldsymbol{\mu}(t_0) := (\boldsymbol{\mu}(t_1) \otimes \cdots \otimes \boldsymbol{\mu}(t_k)) \cdot \mu(\sigma)(\|t_0\|) \, .$$

We define the formal tree series represented by $\mathcal{A}$ to be the function $[\![\mathcal{A}]\!] : T_\Sigma \to \mathbb{K}$ defined by $[\![\mathcal{A}]\!](t) := \boldsymbol{\mu}(t)_1$.

We also introduce the notion of a generalised differential tree automaton, we will see later that this does not change the expressiveness. To set up this definition, we first define $\mathbb{K}_u(x_0, \ldots, x_k)$ to be the subring of $\mathbb{K}(x_0, \ldots, x_k)$ consisting of rational functions of the form

$$\frac{P(x_0, \ldots, x_k)}{Q_0(x_0) \cdots Q_k(x_k)}$$

where $P \in \mathbb{K}[x_0, \ldots, x_k]$ is a multivariate polynomial and $Q_i \in \mathbb{K}[x_i]$ for $i \in \{0, \ldots, k\}$ are univariate polynomials such that Q_0 has no positive integer root and $Q_1, \ldots, Q_k$ have no nonnegative integer root. Intuitively, for a tree $t_0 = \sigma(t_1, \ldots, t_k)$, the variables $x_0, \ldots, x_k$ are instantiated with the sizes of the trees $t_0, \ldots, t_k$ respectively. The conditions on the roots of $Q_0, \ldots, Q_k$ therefore ensure that the transition weight is well defined.

A generalised $\mathbb{K}$-differential tree automaton is defined as above except that for $\sigma \in \Sigma_\ell$ with $\ell \geq 1$ we now take $\mu(\sigma)$ to be an element of $\mathbb{K}_u(x_0, \ldots, x_k)^{d^\ell \times d}$. Such an automaton likewise induces a function $\boldsymbol{\mu} : T_\Sigma \to \mathbb{K}^{1\times d}$, which is defined as above except that now we have

$$\boldsymbol{\mu}(t_0) := (\boldsymbol{\mu}(t_1) \otimes \cdots \otimes \boldsymbol{\mu}(t_k)) \cdot \mu(\sigma)(\|t_0\|, \ldots, \|t_k\|) \, .$$

▶ **Proposition 5.** *For every generalised differential tree automaton $\mathcal{A}$, there exists a differential tree automaton $\mathcal{B}$ such that $[\![\mathcal{A}]\!] = [\![\mathcal{B}]\!]$.*

The proof is provided in [22, Appendix E]. Henceforth we drop the term *generalised* when referring to differential tree automata.

The definition of the *generating function* $f_\mathcal{A}$ and *labelled generating function* $\tilde{f}_\mathcal{A}$ of a differential tree automaton $\mathcal{A}$ is exactly as for a weighted tree automaton, namely via Equations (2) and (3). In general $f_\mathcal{A}$ need not be algebraic, unlike for weighted tree automata, but we will show that it is differentially algebraic.

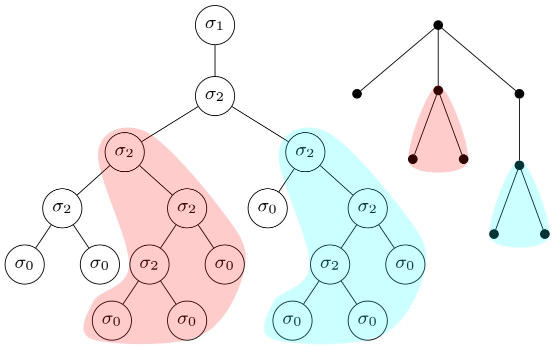

Figure 3 Ranked binary tree (on left), encoding unlabelled rooted plane tree (on right).

▶ **Example 6** (Labelled trees)**.** A labelled rooted non-plane tree with n nodes is a tree whose nodes are labelled from 1 to n and where the children of each node are unordered. Let $(t_n)_{n=0}^{\infty} = (0, 1, 2, 9, 64, 625, \ldots)$ be the sequence whose general term counts the number of labelled rooted non-plane trees with n nodes and write

$$f_T(x) = \sum_{n=0}^{\infty} \frac{t_n}{n!} x^n$$

for its exponential generating function. This generating function has been studied, for instance in the context of combinatorial species [16], and satisfies the equation $f_T = x \exp(f_T)$. Differentiating, we see that f_T is a solution of the differential equation

$$x(1-y)y' = y\,. \tag{4}$$

From there, the sequence $(a_n)_{n=0}^{\infty} = (\frac{t_n}{n!})_{n=0}^{\infty}$ satisfies the non-linear recurrence relation

$$a_{n+1} = \frac{1}{n} \sum_{k=0}^{n-1} (n-k) a_{k+1} a_{n-k} \tag{5}$$

obtained as follows. Write $f_T = \sum_{n=0}^{\infty} a_n\, x^n$ and plug it in (4). Since $x f_T'(x) = \sum_{n=0}^{\infty} n a_n\, x^n$, we get

$$\left(\sum_{n=0}^{\infty} n a_n\, x^n\right)\left(1 - \sum_{n=0}^{\infty} a_n\, x^n\right) = \sum_{n=0}^{\infty} a_n\, x^n$$

Since $a_0 = 0$ and by definition of the Cauchy product,

$$\sum_{n=0}^{\infty} n a_n\, x^n - \sum_{n=0}^{\infty} \left(\sum_{k=1}^{n-1} a_k (n-k) a_{n-k}\right) x^n = \sum_{n=0}^{\infty} a_n\, x^n$$

To express the coefficients a_{n+1} with $n \geq 0$, we extract these coefficients from the three sums

$$(n+1)a_{n+1} - \sum_{k=1}^{n} a_k (n+1-k) a_{n+1-k} = a_{n+1}$$

from which we retrieve (5) by a simple index shift.

From this recurrence relation we obtain a differential tree automaton $\mathcal{A}$ over alphabet $\Sigma = \{\sigma_0, \sigma_1, \sigma_2\}$, with states q_1, q_2 and transitions

$$\sigma_0 \xrightarrow{1} q_2, \qquad \sigma_1(q_2) \xrightarrow{1} q_1, \qquad \sigma_2(q_2, q_2) \xrightarrow{\frac{x_2+1}{x_0}} q_2$$

whose ordinary generating function $f_{\mathcal{A}}$ is equal to f_T. In its matrix representation, the automaton $\mathcal{A}$ has dimension 2 with the transition function μ defined as $\mu(\sigma_0) = \begin{bmatrix} 0 & 1 \end{bmatrix}$ and

$$\mu(\sigma_1) = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad \mu(\sigma_2) = \begin{bmatrix} \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ 0 & \frac{x_2+1}{x_0} \end{bmatrix}$$

By a simple induction given below, we notice that for all $n \in \mathbb{N}$, the vector $\boldsymbol{a}_n$ summing all $\boldsymbol{\mu}(t)$ for trees t of size n, satisfies:

$$\boldsymbol{a}_n = \sum_{t \in T_\Sigma, \|t\|=n} \boldsymbol{\mu}(t) = \begin{bmatrix} \frac{t_n}{n!} & \frac{t_{n+1}}{(n+1)!} \end{bmatrix} .$$

Indeed, decomposing the sum over all trees according to their root symbol, we have

$$\sum_{t \in T_\Sigma, \|t\|=n} \boldsymbol{\mu}(t) = \sum_{t \in T_\Sigma, \|t\|=n-1} \boldsymbol{\mu}(\sigma_1(t)) + \sum_{k=0}^{n-1} \sum_{\substack{t_1 \in T_\Sigma \\ \|t_1\|=k}} \sum_{\substack{t_2 \in T_\Sigma \\ \|t_2\|=n-k-1}} \boldsymbol{\mu}(\sigma_2(t_1, t_2))$$

The definition of $\boldsymbol{\mu}$ and the bilinearity of the Kronecker product yields

$$\boldsymbol{a}_n = \boldsymbol{a}_{n-1}\mu(\sigma_1) + \sum_{k=0}^{n-1} (\boldsymbol{a}_k \otimes \boldsymbol{a}_{n-k-1}) \cdot \mu(\sigma_2)$$

The inductive proof is completed using (5).

The equality $f_{\mathcal{A}} = f_T$ can also be established using combinatorial reasoning. To this end, consider the set

$$S_{\mathcal{A}} := \{t \in T_\Sigma : [\![\mathcal{A}]\!](t) \neq 0\}$$

comprising those trees $t \in T_\Sigma$ in which the root has arity one and all other internal nodes have arity two. There is a natural bijective map $\Phi : S_{\mathcal{A}} \to \mathcal{T}_{rp}$, where $\mathcal{T}_{rp}$ is the set of rooted unlabelled plane trees. For $t \in S_{\mathcal{A}}$, $\Phi(t)$ is defined as follows (see Figure 3):

- the nodes of $\Phi(t)$ are the non-leaf nodes of t,
- the root of $\Phi(t)$ is the root of t,
- the two children of a binary node in t are respectively the next sibling and first child of that node in $\Phi(t)$.

Let $\mathcal{T}_{rn}$ be the set of rooted unlabelled non-plane trees. There is a natural map $\Phi' : \mathcal{T}_{rp} \to \mathcal{T}_{rn}$, which maps a rooted plane tree to its underlying non-plane tree by forgetting the order of the children of each node. Thus we have a map $\Phi' \circ \Phi : S_{\mathcal{A}} \to \mathcal{T}_{rn}$, describing how elements of $S_{\mathcal{A}}$ encode rooted unlabelled non-plane trees. The key design feature of $\mathcal{A}$ is that for every $\tau \in \mathcal{T}_{rn}$ we have[1]

$$\sum_{\substack{t \in S_{\mathcal{A}} \\ \Phi'(\Phi(t))=\tau}} [\![\mathcal{A}]\!](t) = 1 .$$

[1] This can be shown via the identity $\sum_{\sigma \in S_k} \prod_{j=1}^{k} \frac{n_{\sigma(j)}}{\sum_{i=1}^{j} n_{\sigma(i)}} = 1$ for positive integers k and $n_1, \ldots, n_k$. with S_k the set of all permutations of $\{1, \ldots, k\}$ [24].

Multiplying both sides of the above equation by $n!$ (the number of labellings of τ) and summing over all $\tau \in \mathcal{T}_{rn}$ of size n, we have

$$n! \sum_{\substack{t \in T_\Sigma \\ \|t\|=n}} [\![\mathcal{A}]\!](t) = t_n$$

for all n, and we conclude that $f_\mathcal{A} = f_T$. ◁

In the previous example the power series f_T satisfies the equation $y' = \frac{y}{x(1-y)}$ (see Equation (4)). This does not yet imply that f_T is RDA since $\frac{y}{x(1-y)}$ is not defined at $(0, y(0))$. However, by introducing a second power series variable z such that $y = x\,z$, we can see that Equation (4) is equivalent to

$$y' = z + \frac{x\,z^2}{1 - x\,z} \qquad z' = \frac{z^2}{1 - x\,z}$$

which shows that f_T is RDA.

One approach to show that a power series f is RDA consists of differentiating once the polynomial differential equation satisfied by f to obtain a new polynomial equation where the degree of highest order is always one.

▶ **Example 7.** Consider the solution $y_1(x)$ of the differential equation $y_1'^3 + y_1^3 = 1$ such that $y_1(0) = 0$ and $y_1'(0) = 1$. Differentiating the equation we deduce that

$$3(y_1')^2 y_1'' + 3(y_1)^2 y_1' = 0\,,$$

which leads to the following rational dynamical system:

$$y_1' = y_2 \qquad\qquad y_2' = -\frac{y_1^2}{y_2} \tag{6}$$

As $y_2(0) = y_1'(0) \neq 0$, the rational functions in the system are well defined at $(0, y_1(0), y_2(0))$, implying that $y_1(x)$ is RDA. The proofs of Propositions 10 and 11 construct a differential tree automaton $\mathcal{A}$ over the alphabet $\Sigma = \{\sigma_0, \sigma_1, \sigma_3, \sigma_5\}$ with states q_1, q_2, q_3 and transitions

$$\sigma_0 \xrightarrow{1} q_2 \qquad \sigma_0 \xrightarrow{1} q_3 \qquad \sigma_1(q_2) \xrightarrow{\frac{1}{x_0}} q_1$$

$$\sigma_3(q_1, q_1, q_3) \xrightarrow{-\frac{1}{x_0}} q_2 \qquad \sigma_5(q_1, q_1, q_3, q_3, q_3) \xrightarrow{\frac{1}{x_0}} q_3$$

such that $f_\mathcal{A}(x) = y_1(x)$. Section 3 presents a detailed account of the construction of $\mathcal{A}$. Running the automaton, we compute its generating function $f_\mathcal{A}$ as:

$$x - \frac{2}{4!}x^4 - \frac{20}{7!}x^7 - \frac{3320}{10!}x^{10} - \frac{1598960}{13!}x^{13} - \dots . ◁$$

Thus far, all power series presented in this section were RDA. However, not all differentially algebraic power series are RDA. Indeed, we show in Proposition 10 that RDA power series actually coincide with CDA series, the latter being incomparable with the class of differentially finite power series. (It was shown in [27, Example 2.5] that $\frac{1}{\cos(x)}$ is not D-finite but CDA, whereas $\sum_{n=0}^{\infty} n! x^n$ is D-finite but not CDA.) Our automata allow us to go strictly beyond these two classes.

▶ **Example 8.** Consider the power series $f(x) = \sum_{n=0}^{\infty} b_n x^n$ satisfying the equation

$$y - 1 = x^2 y' y + x y^2\,. \tag{7}$$

Below we show that $f(x)$ is neither D-finite nor CDA.

From (7) the sequence $(b_n)_{n=0}^{\infty}$ satisfies the recurrence

$$b_n = \sum_{k=0}^{n-1} (n-k) b_k b_{n-k-1} \tag{8}$$

with $b_0 = 1$. This ensures that $b_n \geq n!$ for all n.

If f were CDA then we would have $b_n \leq \alpha^n$ for some constant $\alpha > 0$ [3, Theorem 3]. Hence f is not CDA. Moreover, differentiating Equation (7), we have that for each positive integer k, f satisfies an equation of the form $y^{(k)} = \sum_{i=0}^{2k} P_{k,i}(x) y^{i+1-2k}$ with $P_{k,0}(x)$ not identically zero. Suppose for a contradiction that $f(x)$ is differentially finite. Then it satisfies the equation: $\sum_{k=0}^{d} Q_k(x) y^{(k)} = 0$. which, together with the above expression for each $y^{(k)}$, entails that $f(x)$ is the root of a non-zero polynomial $P(x,y)$, i.e., f is algebraic. But the growth of an algebraic power series is $O(c^n n^d)$ for some constants c, d, which contradicts $a_n \geq n!$.

Notice the similarity between the recurrences in Equations (5) and (8). This is explained by the fact that the corresponding power series are respectively ordinary and labelled generating functions of a single common automaton. Specifically, define automaton $\mathcal{A}$ over the alphabet $\Sigma = \{\sigma_0, \sigma_2\}$ with states q_1 and transitions

$$\sigma_0 \xrightarrow{1} q_1, \qquad \sigma_2(q_1, q_1) \xrightarrow{x_2+1} q_1$$

The power series $f(x)$ is the ordinary generating function of $\mathcal{A}$. On the other hand, the following power series

$$\sum_{n=0}^{\infty} \frac{t_{n+1}}{(n+1)!} x^n$$

with $(t_n)_n^{\infty}$ as defined in Example 6 is the labelled generating function of $\mathcal{A}$. This observation is generalized in Proposition 12. ◁

2.4 Multi-holonomic sequences

Motivated by the relation between differentially algebraic power series and recurrence relations in the previous examples, we introduce the notion of a *multi-holonomic recurrence* for a sequence of vectors $(\boldsymbol{a}_n)_{n=0}^{\infty}$ in $\mathbb{K}^d$.

Fix $d \in \mathbb{N}$. A sequence $(\boldsymbol{a}_n)_{n=0}^{\infty}$ of elements of $\mathbb{K}^d$ is *multi-holonomic* if there exist $m \in \mathbb{N}$, and for all $\ell \in \{1, \ldots, m\}$ a matrix of polynomials $\mu_\ell \in \mathbb{K}_u(x)^{d^\ell \times d}$ such that for all $n \geq 1$,

$$\boldsymbol{a}_n = \sum_{\substack{\ell \in \{1,\ldots,m\} \\ \boldsymbol{j} \in J_n(\ell)}} (\boldsymbol{a}_{j_1} \otimes \ldots \otimes \boldsymbol{a}_{j_\ell}) \cdot \mu_\ell(n)\,, \tag{9}$$

where $J_n(\ell)$ is the set of tuples $(j_1, \ldots, j_\ell) \in \mathbb{N}^\ell$ such that $j_1 + \ldots + j_\ell = n-1$. We call $d \in \mathbb{N}$ the *order* of the recurrence and m its arity. The *degree* of the recurrence is the maximum degree of the numerator and denominator over all entries of the matrices μ_ℓ. The common denominator of the recurrence is the polynomial $A(x) \in \mathbb{K}[x]$ of smallest degree such that each μ_ℓ can be expressed as a matrix with polynomial entries divided by $A(x)$, that is, $\mu_\ell \in \frac{1}{A(x)} \mathbb{K}[x]^{d^\ell \times d}$.

Syntactically speaking, a multi-holonomic recurrence can be regarded a differential tree automaton over an alphabet with exactly one symbol of each arity. The difference is that we view the recurrence as defining a sequence of vectors rather than a formal tree series. As with automata, we introduce a generalisation of multi-holonomic recurrences that permits additional predicates on the summation indices. Formally, a sequence $(\boldsymbol{a}_n)_{n=0}^{\infty}$ of elements of $\mathbb{K}^d$ is *generalised multi-holonomic* if there exist a finite set $I \subseteq \bigcup_{\ell\geq 0} \mathbb{Z}\times\mathbb{N}^\ell$ and for all tuples $\boldsymbol{s} = (s_0,\dots,s_\ell) \in I$ a matrix of polynomials

$$\mu_{\boldsymbol{s}} \in \mathbb{K}_u(x_0,\dots,x_\ell)^{d^\ell\times d}$$

such that for all $n \geq 1$,

$$\boldsymbol{a}_n = \sum_{\substack{\boldsymbol{s}\in I\\ \boldsymbol{j}\in G_n(\boldsymbol{s})}} (\boldsymbol{a}_{j_1}\otimes\ldots\otimes\boldsymbol{a}_{j_\ell})\cdot\mu_{\boldsymbol{s}}(n,\boldsymbol{j})\,,$$

where $G_n(s_0,\dots,s_\ell)$ is the set of tuples $(j_1,\dots,j_\ell)\in\mathbb{N}^\ell$ such that $j_1+\ldots+j_\ell = n-1+s_0$ and for all $k\in\{1,\dots,\ell\}$, $j_k \leq n-1-s_k$.

A multi-holonomic sequence of arity m defined by matrices $\mu_1,\dots,\mu_m$ can be seen as a generalised multi-holonomic sequence by taking the set $I = \{\mathbf{0}_2,\dots,\mathbf{0}_{(m+1)}\}$, with $\mathbf{0}_i$ being the tuple composed of i integers 0, and for all $\ell\in\{1,\dots,m\}$,

$$\mu_{\mathbf{0}_{(\ell+1)}}(x_0,\dots,x_\ell) = \mu_\ell(x_0)\,.$$

The third main result of our paper compares these notions of multi-holonomic sequences. For this purpose, given $d_1 \geq d_2$ we say that a sequence $(\boldsymbol{b}_n)_{n=0}^{\infty}$ of elements of $\mathbb{K}^{d_1}$ *recognises* a sequence $(\boldsymbol{a}_n)_{n=0}^{\infty}$ of elements of $\mathbb{K}^{d_2}$ if $\pi(\boldsymbol{b}_n) = \boldsymbol{a}_n$ for all n, where $\pi : \mathbb{K}^{d_1}\to\mathbb{K}^{d_2}$ is the projection onto the first d_2 coordinates. The following result is a useful step in the proof of Theorem 2. Detailed proofs can be found in [22, Appendix F].

▶ **Theorem 9.** *Let $(\boldsymbol{a}_n)_{n=0}^{\infty}$ be a sequence of elements of $\mathbb{K}^d$. The following statements are equivalent:*
1. *a multi-holonomic sequence of arity 2 recognises $(\boldsymbol{a}_n)_{n=0}^{\infty}$;*
2. *a multi-holonomic sequence recognises $(\boldsymbol{a}_n)_{n=0}^{\infty}$;*
3. *a generalised multi-holonomic sequence recognises $(\boldsymbol{a}_n)_{n=0}^{\infty}$.*

3 Proof of Theorem 1

In this section we provide an overview of the proof of Theorem 1. The omitted proofs are provided in [22, Appendix B].

The first step of the proof of Theorem 1 consists of showing that constructibly differentiably algebraic (CDA) power series coincide with rationally dynamically algebraic (RDA) power series. Recall that a RDA power series $f(x)$ arises as the first component of a solution $(y_1(x),\dots,y_k(x))$ of a system of differential equations

$$y_1' = \frac{P_1(y_1,\dots,y_k)}{Q_1(y_1,\dots,y_k)},\dots,y_k' = \frac{P_k(y_1,\dots,y_k)}{Q_k(y_1,\dots,y_k)} \tag{10}$$

where $P_1,Q_1,\dots,P_k,Q_k\in\mathbb{K}[x,y_1,\dots,y_k]$ are polynomial functions with $Q_1,\dots,Q_k$ defined at $(0,y_1(0),\dots,y_k(0))$. A power series is CDA when $Q_1=\ldots=Q_k=1$. The CDA power series are thus RDA power series by definition. The converse is given by the following proposition.

▶ **Proposition 10.** *The class of RDA and CDA power series coincide.*

The proof follows a straightforward idea: to transform the system in (10) into an equivalent form where all denominators disappear, we introduce auxiliary variables that replace the denominators and differentiate the equations. We show the idea of the proof through the following RDA series, which we also use as a running example throughout this section. Consider the RDA series $y_1(x)$ defined in Equation (6) of Example 7.

We show that $y_1(x)$ is a CDA by introducing a new variable $y_3 := \frac{1}{y_2}$. Since $y_3' = \frac{-y_2'}{y_2^2} = y_1^2 y_3^3$, the system can be rewritten as

$$y_1' = y_2 \qquad y_2' = -y_1^2 y_3 \qquad y_3' = y_1^2 y_3^3, \tag{11}$$

with initial conditions $y_1(0) = 0$, $y_2(0) = 1$, and $y_3(0) = 1$.

In combination with Proposition 10, the following result establishes the implication $1 \Rightarrow 2$ in Theorem 1.

▶ **Proposition 11.** *A CDA power series $f(x)$ is the ordinary generating function of a differential tree automaton in which the rational weight of every transition has the form $\frac{a}{x}$ for some $a \in \mathbb{K}$.*

Below, we present a simplified version of the proof for the CDA power series solution $y_1(x)$ of the system given in (11). We denote by m the maximal total degree of the polynomials, yielding $m = 5$. Write $y_j(x) = \sum_{n=0}^{\infty} a_{n,j} x^n$ for all $j \in \{1, 2, 3\}$ for a power series solution of the system. Substituting this solution in the defining system entails that

$$\begin{aligned} n\, a_{n,1} &= a_{n-1,2} \\ n\, a_{n,2} &= -\sum_{n_1+n_2+n_3=n-1} a_{n_1,1} a_{n_2,1} a_{n_3,3} \\ n\, a_{n,3} &= \sum_{n_1+\ldots+n_5=n-1} a_{n_1,1} a_{n_2,1} a_{n_3,3} a_{n_4,3} a_{n_5,3}\,. \end{aligned}$$

Following the above observation, we build a differential tree automaton $\mathcal{A} = (4, \mu)$ over an alphabet Σ such that the n-th coefficient of $\boldsymbol{f}_{\mathcal{A}}(x)$ is the vector $\begin{bmatrix} a_{n,1} & a_{n,2} & a_{n,3} & 1 \end{bmatrix}$. In this construction the alphabet Σ contains σ_ℓ for ℓ ranging between 0 and m; in our running example above, we have $\Sigma = \{\sigma_\ell \,|\, 0 \le \ell \le 5\}$. Now, roughly speaking, we introduce a transition rule for each monomial $\alpha\, a_{n_1,i_1} \ldots a_{n_\ell,i_\ell}$ with $\ell > 1$ appearing in the summand of $n a_{n,j}$ as follows

$$\sigma_\ell(q_{i_1}, \ldots, q_{i_\ell}) \xrightarrow{\frac{\alpha}{x_0}} q_j$$

This procedure outputs the automaton given in Example 7 for the system in (11), where the following transitions follow from the above rule:

$$\sigma_1(q_2) \xrightarrow{\frac{1}{x_0}} q_1 \qquad\qquad \sigma_3(q_1, q_1, q_3) \xrightarrow{-\frac{1}{x_0}} q_2 \qquad\qquad \sigma_5(q_1, q_1, q_3, q_3, q_3) \xrightarrow{\frac{1}{x_0}} q_3 \tag{12}$$

We define the transitions of nullary and unary symbols (corresponding to $\ell = 0, 1$) in a specific way, as detailed in the proof provided in [22, Appendix B]. Here, the additional transitions $\sigma_0 \xrightarrow{1} q_2$ and $\sigma_0 \xrightarrow{1} q_3$ initialize the sequence.

To establish the remaining implications in Theorem 1 we use Proposition 12, which shows that the labelled and ordinary generating function of an automaton induces multi-linear recurrence relations that solely differ by a factor of $\frac{1}{n}$. To this end, for a differential tree automaton $\mathcal{A} = (d, \mu)$, define $\boldsymbol{f}_{\mathcal{A}}(x) := \sum_{n=0}^{\infty} \boldsymbol{a}_n x^n$ and $\widetilde{\boldsymbol{f}}_{\mathcal{A}}(x) := \sum_{n=0}^{\infty} \boldsymbol{b}_n x^n$, where

$$\boldsymbol{a}_n = \sum_{\substack{t \in T_\Sigma \\ \|t\| = n}} \boldsymbol{\mu}(t) \quad \text{and} \quad \boldsymbol{b}_n = \sum_{\substack{t \in T_\Sigma \\ \|t\| = n}} \lambda(t) \boldsymbol{\mu}(t)\,. \tag{13}$$

Denote by $a_{n,i}$ the i-th coordinate of the coefficient vector $\boldsymbol{a}_n$, and define $f_{\mathcal{A},i}(x) := \sum_{n=0}^{\infty} a_{n,i}x^n$ to be the i-th coordinate power series, that is, the power series with coefficients being the projection of $\boldsymbol{a}_n$ to the i-th coordinate. Observe that $f_{\mathcal{A},1}(x)$ is the ordinary generating function of $\mathcal{A}$, and similarly $\widetilde{f}_{\mathcal{A},1}(x)$ is the labelled generating function of $\mathcal{A}$. More generally, the power series $f_{\mathcal{A},i}(x)$ and $\widetilde{f}_{\mathcal{A},i}(x)$ can be seen respectively as the ordinary and labelled generating functions of $\mathcal{A}$ when q_i is considered the final state.

▶ **Proposition 12.** *Let $\mathcal{A} = (d, \mu)$ be a differential tree automaton over Σ of maximal arity m. Write $\boldsymbol{f}_{\mathcal{A}}(x) = \sum_{n=0}^{\infty} \boldsymbol{a}_n x^n$ and $\widetilde{\boldsymbol{f}}_{\mathcal{A}}(x) = \sum_{n=0}^{\infty} \boldsymbol{b}_n x^n$ as in (13). Then*

$$\boldsymbol{b}_0 = \boldsymbol{a}_0 = \sum_{\sigma\in\Sigma_0} \mu(\sigma)\,,$$

and for all $n \geq 1$

$$\boldsymbol{a}_n = \sum_{\substack{\ell\in\{1,\ldots,m\}\\ \boldsymbol{j}\in J_n(\ell)}} (\boldsymbol{a}_{j_1} \otimes \ldots \otimes \boldsymbol{a}_{j_\ell}) \cdot \sum_{\sigma\in\Sigma_\ell} \mu(\sigma)(n)$$

$$\boldsymbol{b}_n = \frac{1}{n} \sum_{\substack{\ell\in\{1,\ldots,m\}\\ \boldsymbol{j}\in J_n(\ell)}} (\boldsymbol{b}_{j_1} \otimes \ldots \otimes \boldsymbol{b}_{j_\ell}) \cdot \sum_{\sigma\in\Sigma_\ell} \mu(\sigma)(n)$$

where $J_n(\ell)$ is the set of tuples $(j_1, \ldots, j_\ell) \in \mathbb{N}^\ell$ such that $j_1 + \ldots + j_\ell = n - 1$.

The proof follows by bilinearity of the Kronecker product, and a combinatorial reasoning on the number of monotonic labellings of the set $\{1, \ldots, n\}$. For our running example, we constructed a differential tree automaton over $\Sigma = \{\sigma_0, \sigma_1, \sigma_3, \sigma_5\}$ above by transitions in Equation (12) (see also Example 7). The multi-holonomic sequence of the ordinary generating function of this automaton is defined by $\boldsymbol{a}_0 = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$ and

$$\begin{aligned}\boldsymbol{a}_n = \boldsymbol{a}_{n-1}\, \mu(\sigma_1)(n) &+ \sum_{\boldsymbol{j}\in J_n(3)} (\boldsymbol{a}_{j_1} \otimes \boldsymbol{a}_{j_2} \otimes \boldsymbol{a}_{j_3}) \cdot \mu(\sigma_3)(n) \\ &+ \sum_{\boldsymbol{j}\in J_n(5)} (\boldsymbol{a}_{j_1} \otimes \ldots \otimes \boldsymbol{a}_{j_5}) \cdot \mu(\sigma_5)(n) \end{aligned} \tag{14}$$

where matrix $\mu(\sigma_1) \in \mathbb{K}[x]^{3\times3}$ is zero everywhere except at the entry in row q_2 and column q_1, where it is $\frac{1}{x_0}$. Similarly, the matrix $\mu(\sigma_3) \in \mathbb{K}[x]^{3^3\times3}$ also has a single nonzero entry $\frac{-1}{x_0}$, in row (q_1, q_1, q_3) and column q_2. Finally, the matrix $\mu(\sigma_5) \in \mathbb{K}[x]^{3^5\times3}$ is again zero everywhere except at the entry in row $(q_1, q_1, q_3, q_3, q_3)$ and column q_3, where it is $\frac{1}{x_0}$.

Since the sequences $(\boldsymbol{a}_n)_{n=0}^{\infty}$ and $(\boldsymbol{b}_n)_{n=0}^{\infty}$ defined in Proposition 12 are multi-holonomic with defining matrices $\mu_\ell = \sum_{\sigma\in\Sigma_\ell} \mu(\sigma)$, with $\ell \in \{0, \ldots, m\}$, the implication $2 \Rightarrow 1$ in Theorem 1 follows directly from this proposition combined with the following result.

▶ **Proposition 13.** *Let $(\boldsymbol{a}_n)_{n=0}^{\infty}$ be a multi-holonomic sequence of degree at most 1 with common denominator x. All coordinate power series of $\boldsymbol{f}(x) = \sum_{n=0}^{\infty} \boldsymbol{a}_n\, x^n$ are RDA.*

Below, we present the main ideas of the proof for the multi-holonomic sequence computed in Equation (14) for our running example. Let $\boldsymbol{f} = \sum_{n=0}^{\infty} \boldsymbol{a}_n x^n$ where $\boldsymbol{a}_n$ satisfies Equation (14). We first derive a system of differential equations for the power series $\boldsymbol{f}$ from the recurrence relation for $\boldsymbol{a}_n$; we give an account of this computation, which is a simple variant of the proof of Proposition 14 presented in the next section.

Denote by Θf the differential operator defined by $\Theta f(x) := xf'(x)$. Hence,

$$\begin{aligned}
\Theta \boldsymbol{f} &= \sum_{n=1}^{\infty} n\boldsymbol{a}_n x^n = \sum_{n=0}^{\infty}(n+1)\boldsymbol{a}_{n+1}x^{n+1} \\
&= \sum_{n=0}^{\infty}(n+1)x^{n+1} \sum_{\ell\in\{1,3,5\}} \sum_{\boldsymbol{j}\in J_{n+1}(\ell)} (\boldsymbol{a}_{j_1}\otimes\ldots\otimes\boldsymbol{a}_{j_\ell})\cdot\mu(\sigma_\ell)(n+1) \qquad \text{by Eq. (14)} \\
&= \sum_{n=0}^{\infty}(n+1)x^{n+1} \sum_{\ell\in\{1,3,5\}} \sum_{\boldsymbol{j}\in J_{n+1}(\ell)} (\boldsymbol{a}_{j_1}\otimes\ldots\otimes\boldsymbol{a}_{j_\ell})\cdot\frac{1}{n+1}\nu_\ell \quad \text{writing } \mu(\sigma_\ell)=\frac{1}{x}\nu_\ell \\
&= x\sum_{n=0}^{\infty} \sum_{\ell\in\{1,3,5\}} \sum_{\boldsymbol{j}\in J_{n+1}(\ell)} (\boldsymbol{a}_{j_1}x^{j_1}\otimes\ldots\otimes\boldsymbol{a}_{j_\ell}x^{j_\ell})\cdot\nu_\ell \\
&= x\sum_{\ell\in\{1,3,5\}} (\sum_{j_1=0}^{\infty}\boldsymbol{a}_{j_1}x^{j_1}\otimes\ldots\otimes\sum_{j_\ell=0}^{\infty}\boldsymbol{a}_{j_\ell}x^{j_\ell})\cdot\nu_\ell \qquad \text{by bilinearity of } \otimes \\
&= x\sum_{\ell\in\{1,3,5\}} (\boldsymbol{f}\otimes\ldots\otimes\boldsymbol{f})\cdot\nu_\ell
\end{aligned}$$

In this simple example, the derived system of differential equations for the power series $\boldsymbol{f}$, above, already implies that $\boldsymbol{f}' = \sum_{\ell\in\{1,3,5\}}(\boldsymbol{f}\otimes\ldots\otimes\boldsymbol{f})\cdot\nu_\ell$, proving that the coordinate series of $\boldsymbol{f}$ are all RDA. In general, such a system can be reorganized as

$$\begin{cases}
\Theta f_1 = xQ_1(f_1,\ldots,f_d) + \sum_{i=1}^{d}\Theta f_i\cdot xP_{1,i}(f_1,\ldots,f_d) \\
\vdots \\
\Theta f_d = xQ_d(f_1,\ldots,f_d) + \sum_{i=1}^{d}\Theta f_i\cdot xP_{d,i}(f_1,\ldots,f_d)
\end{cases}$$

where the f_i are the coordinate power series of $\boldsymbol{f}$, and the $Q_i, P_{i,j}$ are polynomials in $\mathbb{K}[y_1,\ldots,y_d]$.

This can be seen as a linear system of equations in $\Theta f_1,\ldots,\Theta f_d$ with coefficients in the field of rational functions $\mathbb{K}(x,f_1,\ldots,f_d)$. As such, we can compute the determinant $\det(M)$, where M represents the coefficient matrix of the system. Crucially we show that $\det(M)$ is a polynomial in $1+x\mathbb{K}[x,f_1,\ldots,f_d]$. Hence, $\det(M)$ does not vanish at $(0,f_1(0),\ldots,f_d(0))$, which in turn implies that

$$\begin{bmatrix}\Theta f_1\\ \vdots \\ \Theta f_d\end{bmatrix} = M^{-1}\begin{bmatrix}xQ_1\\ \vdots \\ xQ_d\end{bmatrix} = x\begin{bmatrix}U_1\\ \vdots \\ U_d\end{bmatrix}$$

where the rational functions $U_i\in\mathbb{K}(x,f_1,\ldots,f_d)$ are defined at $(0,f_1(0),\ldots,f_d(0))$.

We complete the overview on proof of Theorem 1 by discussing the equivalence $2\Leftrightarrow 3$ in the theorem. Given any multi-holonomic sequence $(\boldsymbol{a}_n)_{n=0}^{\infty}$ of order d and arity m, defined by matrices $\mu_1,\ \ldots,\ \mu_m$, Proposition 12 implies that the sequence arises as the ordinary generating function $\boldsymbol{f}_{\mathcal{A}}(x)$ of some differential tree automaton $\mathcal{A}$. More precisely, define $\mathcal{A}=(d,\mu)$ over the alphabet $\Sigma=\{\sigma_0,\ldots,\sigma_m\}$ such that σ_i has arity i, $\mu(\sigma_0)=\boldsymbol{a}_0$, and $\mu(\sigma_\ell)(x)=\mu_\ell(x)$ for all $\ell\in\{1,\ldots,m\}$. Replacing transition matrix $\mu(\sigma_\ell)(x)$ in $\mathcal{A}$ with $\mu(\sigma_\ell)(x)=x\cdot\mu_\ell(x)$ gives another automaton $\mathcal{B}$ whose labelled generating function $\widetilde{\boldsymbol{f}}_{\mathcal{B}}(x)$ is induced by $(\boldsymbol{a}_n)_{n=0}^{\infty}$.

Since the proof of the equivalence Item 1 $\Leftrightarrow$ Item 2 in the theorem relies on Proposition 11, we can now establish $2\Rightarrow 3$ by restricting our attention to differential tree automata in which the rational weight of every transition has the form $\frac{a}{x}$ for some $a\in\mathbb{K}$. Given such a differential tree automaton $\mathcal{A}$ we apply Proposition 12 to show that the vector of power series $\boldsymbol{f}_{\mathcal{A}}(x) = \sum_{n=0}^{\infty} \boldsymbol{a}_n x^n$ induces a multi-holonomic sequence $(\boldsymbol{a}_n)_{n=0}^{\infty}$ in which every entry of the defining matrices also has the form $\frac{a}{x}$ for some $a \in \mathbb{K}$. We can thus apply the above observation to show the existence of a $\mathbb{K}$-weighted automaton $\mathcal{B}$ whose labelled generating function $\widetilde{\boldsymbol{f}}_{\mathcal{B}}(x)$ equals $\boldsymbol{f}_{\mathcal{A}}(x)$. The converse direction $3 \Rightarrow 2$ is analogous (application of Proposition 12 followed by the above observation).

4 Proof of Theorem 2

This theorem shows the equivalence of differentially algebraic power series, ordinary generating function of differential tree automata, and multi-holonomic sequences of arity 2. The most challenging development in this theorem is to establish $1 \Leftrightarrow 3$. Before discussing this equivalence, we outline the key ingredients required to establish $2 \Leftrightarrow 3$ in Theorem 2. Detailed proofs can be found in [22, Appendix C].

By Proposition 12, the ordinary generating function of any differential tree automaton $\mathcal{A}$ is the first component of the vector of power series $\boldsymbol{f}_{\mathcal{A}}(x) = \sum_{n=0}^{\infty} \boldsymbol{a}_n x^n$. The latter induces a multi-holonomic sequence $(\boldsymbol{a}_n)_{n=0}^{\infty}$. By Theorem 9, this sequence is recognised by a multi-holonomic sequence of arity 2 which concludes the proof of $2 \Rightarrow 3$ in Theorem 2. We note in passing that the main milestone of the above argument is addressed in Theorem 9.

Conversely, consider the sequence $(\boldsymbol{b}_n)_{n=0}^{\infty}$ of elements in $\mathbb{K}^d$ satisfying for all $n \geq 1$ the recurrence

$$\boldsymbol{b}_n = \boldsymbol{b}_{n-1} \cdot P(n) + \sum_{k=0}^{n-1} (\boldsymbol{b}_k \otimes \boldsymbol{b}_{n-k-1}) \cdot Q(n) \tag{15}$$

with P and Q having entries defined on positive integers. We define the alphabet $\Sigma = \{\sigma_0, \sigma_1, \sigma_2\}$ with σ_0, σ_1 and σ_2 having, respectively, arity 0, 1 and 2. Finally, we define the differential tree automaton $\mathcal{A} = (d, \mu)$ over Σ such that $\mu(\sigma_0) = \boldsymbol{b}_0$, and

$$\mu(\sigma_1)(x) = P(x) \quad \text{and} \quad \mu(\sigma_2)(x) = Q(x)\,.$$

From Proposition 12, writing $\boldsymbol{f}_{\mathcal{A}}(x)$ as $\sum_{n=0}^{\infty} \boldsymbol{a}_n x^n$, we directly obtain that the sequence $(\boldsymbol{a}_n)_{n=0}^{\infty}$ satisfies the recurrence relation for $(\boldsymbol{b}_n)_{n=0}^{\infty}$ stated in Equation (15). Moreover, the two sequences have the same initial value, that is $\boldsymbol{a}_0 = \boldsymbol{b}_0$. Thus $(\boldsymbol{b}_n)_{n=0}^{\infty} = (\boldsymbol{a}_n)_{n=0}^{\infty}$, which concludes the proof of the direction $3 \Rightarrow 2$ in Theorem 2.

4.1 From Multi-Holonomic Sequences to Differentially Algebraic Power Series

This section focuses on proving $3 \Rightarrow 1$ in Theorem 2. In other words, we show that given a multi-holonomic sequence $(\boldsymbol{a}_n)_{n=0}^{\infty}$, all induced coordinate power series $\sum_{n=0}^{\infty} a_{n,i} x^n$, for $i \in \{1, \ldots, d\}$, are differentially algebraic. For this, we rely once again on the differential operator Θf defined by $\Theta f(x) := x f'(x)$. For all $i \in \mathbb{N}$ we write $\Theta^{(i+1)} f(x)$ for the $i+1$-th iteration of this operator:

$$\Theta^{(i+1)} f(x) := \Theta(\Theta^{(i)} f)(x)\,.$$

Let $(\boldsymbol{a}_n)_{n=0}^{\infty}$ be a multi-holonomic sequence of arity m and degree r. Let the sequence be defined by matrices $\mu_1, \ldots, \mu_m$, sharing a common denominator $A(x) \in \mathbb{K}[x]$. Since $(\boldsymbol{a}_n)_{n=0}^{\infty}$ has degree r, we may assume that

$$A(x) = \sum_{i=0}^{r} c_i x^i \tag{16}$$

for some $c_0, \ldots, c_r \in \mathbb{K}$. We rewrite each μ_ℓ with $\ell \in \{1, \ldots, m\}$ by collecting the coefficients of the monomial x^i in a matrix $\mu_{\ell,i}$ as follows

$$\mu_\ell(x) = \frac{1}{A(x)} \sum_{i=0}^{r} x^i \cdot \mu_{\ell,i} \tag{17}$$

for some $\mu_{\ell,i} \in \mathbb{K}^{d^\ell \times d}$. Recall that the set $J_n(\ell)$ includes all ℓ-tuples $(j_1, \ldots, j_\ell) \in \mathbb{N}^\ell$ satisfying $j_1 + \ldots + j_\ell = n - 1$.

▶ **Proposition 14.** *Let $(\boldsymbol{a}_n)_{n=0}^\infty$ be a multi-holonomic sequence defined as above, where $A(x)$ and $\mu_\ell(x)$ are rewritten as in* (16) *and* (17)*. The vector of power series $\boldsymbol{f}(x) = \sum_{n=0}^\infty \boldsymbol{a}_n\, x^n$ satisfies the following system of differential equations:*

$$\sum_{i=0}^{r} c_i \Theta^{(i)} \boldsymbol{f} - c_0 \boldsymbol{a}_0 = x \sum_{\substack{\ell \in \{1,\ldots,m\} \\ i \in \{0,\ldots,r\} \\ \boldsymbol{j} \in J_{i+1}(\ell+1)}} \binom{i}{\boldsymbol{j}} (\Theta^{(j_1)} \boldsymbol{f} \otimes \ldots \otimes \Theta^{(j_\ell)} \boldsymbol{f}) \cdot \mu_{\ell,i} \tag{18}$$

In Proposition 14, we showed that given a multi-holonomic sequence $(\boldsymbol{a}_n)_{n=0}^\infty$, the vector of power series $\boldsymbol{f}(x) = \sum_{n=0}^\infty \boldsymbol{a}_n\, x^n$ is a solution of the system as in Equation (18). In [22, Proposition 22 in Appendix C], we show that this vector is the unique solution of the system, given the initial value $\boldsymbol{a}_0$.

We use the uniqueness of the power series solution, given an initial value, to prove that all induced coordinate power series of the solution are differentially algebraic. This is based on a differential version of the Artin approximation theorem given in [12].

▶ **Theorem 15** ([12, Theorem 2.1])**.** *Let S be a system of differential polynomials in the differential variables $y_1, \ldots, y_k$ with coefficients in $\mathbb{K}[x]$. Let $(y_1(x), \ldots, y_k(x)) \in \mathbb{K}[\![x]\!]^k$ be a solution of S and $m \in \mathbb{N}$. Then there exists another solution $(\bar{y}_1(x), \ldots, \bar{y}_k(x)) \in \mathbb{K}[\![x]\!]^k$ such that all $\bar{y}_i(x)$'s are differentially algebraic and*

$$(y_1(x), \ldots, y_k(x)) \equiv (\bar{y}_1(x), \ldots, \bar{y}_k(x)) \pmod{x^m}.$$

This theorem states that any solution (not necessarily differentially algebraic) of a system of a differential equations can be approximated by a differentially algebraic solution, such that both solutions agree on the first m coefficients. This differential version of the Artin approximation theorem is used in the following proposition to complete the proof of the implication 3 ⇒ 1 in Theorem 2.

▶ **Proposition 16.** *Let $(\boldsymbol{a}_n)_{n=0}^\infty$ be a multi-holonomic sequence. All components of $\boldsymbol{f}(x) = \sum_{n=0}^\infty \boldsymbol{a}_n\, x^n$ are differentially algebraic power series.*

4.2 From Differentially Algebraic Power Series to Multi-Holonomic Sequences

It remains to show that any differentially algebraic power series is the component of a multi-holonomic sequence of arity 2, that is, the implication 1 ⇒ 3 in Theorem 2. Our proof builds on a construction of Denef and Lipschitz [12], who showed that the sequence of coefficients of any differentially algebraic power series $f(x) = \sum_{n=0}^\infty a_n x^n$ satisfies a recurrence of the form:

$$P(n)\, a_n = Q_n(a_0, \ldots, a_{n-1}),$$

where Q_n is a polynomial whose form depends on n. Our proof makes explicit this recurrence and shows how it can be transformed into a generalised multi-holonomic sequence. The proof will then conclude by applying Theorem 9.

Observe that the n-th coefficient of a power series $f(x) = \sum_{n=0}^{\infty} a_n x^n$ satisfies $a_n = n!\, f^{(n)}(0)$. Hence, to obtain a recurrence relation between the coefficients, it suffices to obtain a recurrence relation between the derivatives of f evaluated at 0. One way to do so is to differentiate the differential polynomial P on which the differentially algebraic power series $f(x)$ is a root, that is, the polynomial $P(y, y^{(1)}, \ldots, y^{(d)})$ with coefficients in $\mathbb{K}[x]$ such that $P(f(x), f'(x), \ldots, f^{(d)}(x)) = 0$. In particular, using the chain rule, notice that

$$P' = \frac{\partial P}{\partial y^{(d)}} y^{(d+1)} + R', \tag{19}$$

with both R and $\frac{\partial P}{\partial y^{(d)}}$ being polynomials in $\mathbb{K}[x][y, \ldots, y^{(d)}]$, thus of order strictly smaller than $d+1$. The partial derivative of P by its highest-order variable, namely $\frac{\partial P}{\partial y^{(d)}}$, is called the *separant of* P. Since $f(x)$ and its derivatives form a root of P then they are also a root of P'. Hence, instantiating Equation (19) with f and its derivatives evaluated at 0 leads to the following equation.

$$\frac{\partial P}{\partial y^{(d)}}(f(0), \ldots, f^{(d)}(0)) f^{(d+1)}(0) = -R(f(0), \ldots, f^{(d)}(0))$$

We can see here how the recurrence takes shape as $f^{(d+1)}(0)$ is expressed using an equation that depends solely on $f^{(i)}(0)$ for $i \leq d$. However, the main hurdle comes from the fact that the separant can vanish at 0.

To address this issue, we can start by assuming that the separant of P is not identically zero at the solution. In other words,

$$\frac{\partial P}{y^{(d)}}(f, f', \ldots, f^{(d)}) \neq 0\,.$$

This property holds if P has minimal order and degree among differential polynomials that annihilate f (see [22, Proposition 25 in Appendix C]).

Then, the core idea of the Denef and Lipschitz [12] construction consists of iteratively differentiating the polynomial P until the resulting differential polynomial meets a certain condition (requiring that the coefficient of highest order term never vanishes at 0). The number of successive differentiations depends on the separant.

Given $k \in \mathbb{N}$ and differential polynomials $A \in \mathbb{K}[x][y^{(\infty)}]$, denote by $A \mod y^{(k)}$ the polynomials obtained from A by removing all monomials of order at least k (all monomials that are multiples of $y^{(o)}$ for some $o \geq k$).

▶ **Proposition 17.** *Let $f \in \mathbb{K}[\![x]\!]$ be a power series and $P \in \mathbb{K}[x][y, \ldots, y^{(d)}]$ a differential polynomial such that*

$$P(x, f, \ldots, f^{(d)}) = 0 \quad \text{and} \quad \frac{\partial P}{\partial y^{(d)}}(x, f, \ldots, f^{(d)}) \neq 0\,.$$

Then there exist $m, s, N \in \mathbb{N}$ and polynomial $A \in \mathbb{K}[x]$ such that for all $n \geq N$ we have $A(n) \neq 0$ and

$$f^{(n+s+1)}(0) = \frac{1}{A(n)}\, Q_n(0, f(0), \ldots, f^{(n+s)}(0)) \tag{20}$$

where $Q_n = P^{(m+n)} \mod y^{(n+s+1)}$.

To complete the proof of the implication $1 \Rightarrow 3$ in Theorem 2, we show that the polynomial Q_n defined in Proposition 17 naturally induces a generalised multi-holonomic sequence. Indeed, using Pascal's identity, we have for every monomial $M = y^{(i_1)} \ldots y^{(i_\ell)}$ in P and for all $n \in \mathbb{N}$,

$$M^{(n)} = \sum_{\boldsymbol{j} \in J_{n+1}(\ell)} \binom{n}{\boldsymbol{j}} y^{(i_1+j_1)} \cdots y^{(i_\ell+j_\ell)} .$$

Computing $M^{(m+n)}$ modulo $y^{(s+n+1)}$ intuitively corresponds to adding guard conditions on the indices $j_1, \ldots, j_\ell$. As such, $M^{(m+n)} \mod y^{(s+n+1)}$ becomes

$$\sum_{\boldsymbol{j} \in G_{n+1}(\boldsymbol{m})} \binom{n+m}{\boldsymbol{j}} y^{(i_1+j_1)} \cdots y^{(i_\ell+j_\ell)}$$

with $\boldsymbol{m} = (m, i_1 - 1 - s, \ldots, i_\ell - 1 - s)$. Recall that $G_n(s_0, \ldots, s_\ell)$ is used in the definition of generalised multi-holonomic sequence and is the set of tuples $(j_1, \ldots, j_\ell) \in \mathbb{N}^\ell$ such that $j_1 + \ldots + j_\ell = n - 1 + s_0$ and for all $k \in \{1, \ldots, \ell\}$, $j_k \leq n - 1 - s_k$.

The final generalised multi-holonomic sequence is obtained by summing over all monomials and applying an additional shift by N since Equation (20) holds only when $n \geq N$. The proof concludes by invoking Theorem 9.

4.3 Equivalence of differential tree automata and their generating function

We can use Theorem 2 to decide the zeroness and equivalence of the generating functions and formal tree series associated to a differential tree automaton. See the detailed proof in [22, Appendix D], including the reduction from zeroness of a differential tree automaton to zeroness of the ordinary generating function of such an automaton.

▶ **Theorem 18.** *The problems of zeroness and equivalence of formal tree series for $\mathbb{Q}$-differential tree automata are decidable. The problems of zeroness and equivalence of ordinary and labelled generating functions for $\mathbb{Q}$-differential tree automata are decidable.*

Theorem 18 relies on a procedure of Denef and Lipschitz [12] for determining zeroness of the solution of a system of differential algebraic equations. Intuitively, the zeroness of a formal tree series of a differential tree automaton $\mathcal{A}$ is reduced to the zeroness of the ordinary generating function of the differential tree automaton $\mathcal{B}$ such that $[\![\mathcal{B}]\!](t) = [\![\mathcal{A}]\!](t)^2$. As all outputs of $\mathcal{B}$ are nonnegative, zeroness of $\mathcal{B}$ is equivalent to the zeroness of its ordinary generating function. Proposition 14 then translates the ordinary generating function of $\mathcal{B}$ into a finite system of algebraic differential equations. Therefore, zeroness is decided by Denef–Lipshitz procedure. This procedure was recently implemented in [7]; however, we are not aware of an analysis of the complexity of this problem. This would be an interesting question for further research.

References

1 Nikhil Balaji, Lorenzo Clemente, Klara Nosan, Mahsa Shirmohammadi, and James Worrell. Multiplicity Problems on Algebraic Series and Context-Free Grammars. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–12. IEEE, 2023. doi:10.1109/LICS56636.2023.10175707.

2 Jason P. Bell, Stanley Burris, and Karen A. Yeats. Monadic second-order classes of forests with a monadic second-order 0-1 law. *Discret. Math. Theor. Comput. Sci.*, 14(1):87–108, 2012. doi:10.46298/DMTCS.566.

3 François Bergeron and Christophe Reutenauer. Combinatorial Resolution of Systems of Differential Equations III: a Special Class of Differentially Algebraic series. *Eur. J. Comb.*, 11:501–512, 1990. doi:10.1016/S0195-6698(13)80035-2.

4 Jean Berstel and Christophe Reutenauer. Recognizable Formal Power Series on Trees. *Theor. Comput. Sci.*, 18:115–148, 1982. doi:10.1016/0304-3975(82)90019-6.

5 Michele Boreale. Algebra, coalgebra, and minimization in polynomial differential equations. *Log. Methods Comput. Sci.*, 15(1), 2019. doi:10.23638/LMCS-15(1:14)2019.

6 Alin Bostan, Arnaud Carayol, Florent Koechlin, and Cyril Nicaud. Weakly-Unambiguous Parikh Automata and Their Link to Holonomic Series. In *47th International Colloquium on Automata, Languages, and Programming, ICALP*, volume 168 of *LIPIcs*, pages 114:1–114:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.114.

7 François Boulier. The DenefLipshitz Algorithm: (Software Demonstration Support). *ACM Commun. Comput. Algebra*, 59(3):47–52, January 2026. doi:10.1145/3787957.3787959.

8 Alex Buna-Marginean, Vincent Cheval, Mahsa Shirmohammadi, and James Worrell. On Learning Polynomial Recursive Programs. *Proc. ACM Program. Lang.*, 8(POPL):1001–1027, 2024. doi:10.1145/3632876.

9 Giusi Castiglione and Paolo Massazza. On a class of languages with holonomic generating functions. *Theoretical Computer Science*, 658:74–84, 2017. doi:10.1016/J.TCS.2016.07.022.

10 Lorenzo Clemente. Weighted Basic Parallel Processes and Combinatorial Enumeration. In Rupak Majumdar and Alexandra Silva, editors, *35th International Conference on Concurrency Theory, CONCUR*, volume 311 of *LIPIcs*, pages 18:1–18:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.CONCUR.2024.18.

11 Hubert Comon. Tree automata techniques and applications, 1997. URL: https://api.semanticscholar.org/CorpusID:2092186.

12 Jan Denef and Leonard Lipshitz. Power series solutions of algebraic differential equations. *Mathematische annalen*, pages 213–238, 1984.

13 Zoltán Ésik and Werner Kuich. Formal Tree Series. *J. Autom. Lang. Comb.*, 8(2):219–285, 2003. doi:10.25596/JALC-2003-219.

14 Eldar Fischer, Tomer Kotek, and Johann A Makowsky. Application of logic to combinatorial sequences and their recurrence relations. *Model Theoretic Methods in Finite Combinatorics*, 558:1–42, 2011.

15 Philippe Flajolet and Robert Sedgewick. *Analytic combinatorics*. cambridge University press, 2009.

16 Philippe Flajolet, Paul Zimmermann, and Bernard Van Cutsem. A Calculus for the Random Generation of Labelled Combinatorial Structures. *Theor. Comput. Sci.*, 132(2):1–35, 1994. doi:10.1016/0304-3975(94)90226-7.

17 Krister Forsman. On Rational State Space Realizations. *IFAC Proceedings Volumes*, 25(13):341–346, 1992. 2nd IFAC Symposium on Nonlinear Control Systems Design 1992, Bordeaux, France, 24-26 June. doi:10.1016/S1474-6670(17)52305-8.

18 Zoltán Fülöp and Heiko Vogler. Weighted Tree Automata – May it be a little more?, 2024. arXiv:2212.05529.

19 Manuel Kauers. *D-finite Functions*. Springer, 2023.

20 Christian Krattenthaler and Tanguy Rivoal. Arithmetic properties of the taylor coefficients of differentially algebraic power series, 2025. arXiv:2502.09259.

21 Werner Kuich and Arto Salomaa. *Semirings, automata, languages*, volume 5. Springer Science & Business Media, 2012.

22 Rida Ait El Manssour, Vincent Cheval, Mahsa Shirmohammadi, and James Worrell. Differential tree automata, 2025. arXiv:2407.08218.

23 Alexey Ovchinnikov, Gleb Pogudin, and Thieu N Vo. Bounds for elimination of unknowns in systems of differential-algebraic equations. *International Mathematics Research Notices*, pages 12342–12377, 2022.

24 Robin Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975. URL: `http://www.jstor.org/stable/2346567`.

25 Christophe Reutenauer. On a Matrix Representation for Polynomially Recursive Sequences. *The Electronic Journal of Combinatorics [electronic only]*, 19, September 2012. `doi:10.37236/2721`.

26 Arto Salomaa and Matti Soittola. *Automata-theoretic aspects of formal power series*. Springer Science & Business Media, 2012.

27 Richard P. Stanley. Differentiably Finite Power Series. *Eur. J. Comb.*, 1(2):175–188, 1980. `doi:10.1016/S0195-6698(80)80051-5`.

28 Richard P Stanley. *Catalan numbers*. Cambridge University Press, 2015.