Weakly-unambiguous Parikh automata and their link to holonomic series

Alin Bostan, Arnaud Carayol, Florent Koechlin, Cyril Nicaud

Weakly-Unambiguous Parikh Automata and Their Link to Holonomic Series

Alin Bostan
INRIA Saclay Île-de-France, Palaiseau, France
Alin.Bostan@inria.fr

Arnaud Carayol
LIGM, Univ. Gustave Eiffel, CNRS, Marne-la-Vallée, France
arnaud.carayol@u-pem.fr

Florent Koechlin
LIGM, Univ. Gustave Eiffel, CNRS, Marne-la-Vallée, France
florent.koechlin@u-pem.fr

Cyril Nicaud
LIGM, Univ. Gustave Eiffel, CNRS, Marne-la-Vallée, France
cyril.nicaud@u-pem.fr

Abstract

We investigate the connection between properties of formal languages and properties of their generating series, with a focus on the class of *holonomic* power series. We first prove a strong version of a conjecture by Castiglione and Massazza: weakly-unambiguous Parikh automata are equivalent to unambiguous two-way reversal bounded counter machines, and their multivariate generating series are holonomic. We then show that the converse is not true: we construct a language whose generating series is algebraic (thus holonomic), but which is inherently weakly-ambiguous as a Parikh automata language. Finally, we prove an effective decidability result for the inclusion problem for weakly-unambiguous Parikh automata, and provide an upper-bound on its complexity.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Mathematics of computing → Generating functions

Keywords and phrases generating series, holonomicity, ambiguity, reversal bounded counter machine, Parikh automata

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.114

Category Track B: Automata, Logic, Semantics, and Theory of Programming

1 Introduction

This article investigates the link between *holonomic* (or *D-finite*) power series and formal languages. We consider the classical setting in which this connection is established via the generating series $L(x) = \sum_{n\geq 0} \ell_n x^n$ counting the number ℓ_n of words of length n in a given language $\mathcal{L}$.

On the languages side, the Chomsky–Schützenberger hierarchy [10] regroups languages in classes of increasing complexity: regular, context-free, context-sensitive and recursively enumerable. For power series, a similar hierarchy exists, consisting of the rational, algebraic and holonomic series. The first two levels of each hierarchy share a strong connection, as the generating series of a regular (resp. unambiguous context-free) language is a rational (resp. algebraic) power series.

This connection has borne fruits both in formal language theory and in combinatorics. In combinatorics, finite automata and unambiguous grammars are routinely used to establish rationality and algebraicity of particular power series. In formal languages, this connection was (implicitly) used to give polynomial-time algorithms for the inclusion and universality

47th International Colloquium on Automata, Languages, and Programming (ICALP 2020).
Editors: Artur Czumaj, Anuj Dawar, and Emanuela Merelli; Article No. 114; pp. 114:1–114:16

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany


tests for unambiguous finite automata [35]. In [15], Flajolet uses the connection between unambiguous context-free grammars and algebraic series to prove the inherent ambiguity of certain context-free languages, solving several conjectures with this tool. Using analytic criteria on the series (for instance, the existence of infinitely many singularities), he establishes that the series of these context-free languages is not algebraic. Hence these languages cannot be described by unambiguous context-free grammars and are therefore inherently ambiguous.

In this article we propose to extend the connection to holonomic series. Holonomic series enjoy non-trivial closure properties whose algorithmic counterparts are actively studied in computer algebra. Our aim is to show that these advances can be leveraged to obtain non-trivial results in the formal languages and verification worlds. These results are particularly noteworthy as the notion of unambiguity in automata theory is not fully understood [11]. The work of extending the connection was already initiated by Massazza in [27], where he introduces two families of languages, named RCM[1] and linearly constrained languages (LCL), whose generating series are holonomic. These classes are, however, not captured by well-known models of automata, and this limits their appeal. Recently, Castiglione and Massazza addressed this issue and conjectured that RCM contains the languages accepted by deterministic one-way reversal bounded machines (RBCM for short) [8]; Massazza proved the result for RBCM for two subclasses of one-way deterministic RBCM [28, 29]. This conjecture hints that the class RCM is related to models of automata such as RBCM, which are used in program verification.

Our first contribution is to prove a stronger version of this conjecture. We show that RCM and LCL respectively correspond to the languages accepted by weakly-unambiguous[2] version of Parikh automata (PA, for short) [24] and pushdown Parikh automata. Intuitively, Parikh automata are[3] finite non-deterministic word automata enriched with the ability to test semilinear constraints between the number of occurrences of each transition in the run. In terms of RBCM, these classes correspond to unambiguous two-way RBCM and unambiguous one-way RBCM enriched with a stack. Parikh automata are also commonly used in program verification. In view of the literature, these results might seem expected but they still require a careful adaptation of the standard techniques in the absence of a stack and become even more involved when a stack is added.

After having established the relevance of the classes of languages under study, we provide two consequences of the holonomicity of their associated generating series.

The first consequence follows Flajolet's approach mentioned previously and gives criteria to establish the inherent weak-ambiguity for languages accepted by PA or pushdown PA, by proving that their generating series are not holonomic. These criteria are sufficient but not necessary; this is not surprising as the inherent weak-ambiguity is undecidable for languages accepted by PA. Yet, the resulting method captures non-trivial examples with quite short and elegant proofs. In contrast, we give an example of inherently weakly-ambiguous PA language having a holonomic series (and therefore not amenable to the analytic method) for which we prove inherent weak-ambiguity *by hand*. The proof is quite involved but shows the inherent ambiguity of this language for a much larger class of automata (i.e., PA whose semi-linear sets are replaced by arbitrary recursive sets).

---

[1] The name RCM comes from the fact that these languages are defined using a Regular language, a semilinear Constraint and a Morphism.

[2] We use the term weakly-unambiguous here to avoid a possible confusion with the class of unambiguous PA defined in [7] which is strictly contained in our class. Our notion of non-ambiguity is the standard one: every word has at most one accepting computation (this is detailed in Remark 11).

[3] In this article, we use an equivalent definition where the transitions of the automaton are additionally labeled by vectors of natural numbers. A run is accepting if the sum of all the vectors encountered in the run satisfies a semilinear constraint.

The second consequence is of an algorithmic nature. We focus on the inclusion problem for weakly-unambiguous PA, whose decidability can be deduced from Castiglione and Massazza's work [8]. Here our contribution is an effective decidability result: we derive a concrete bound B, depending on the size of the representation of the two PAs, such that the inclusion holds if and only if the languages are included when considering words up to the length B. This bound B is obtained by a careful analysis of the proofs establishing the closure properties of holonomic series (in several variables), notably under Hadamard product and specialization. We do this by controlling various parameters (order, size of the polynomial coefficients, ...) of the resulting partial differential equations.

2 Primer on holonomic power series in several variables

In this section, we introduce power series in several variables and the classes of rational, algebraic and holonomic power series. We recall the connection with regular and context-free languages via the notions of generating series in one or several variables.

Let $\mathbb{Q}[x_1,\dots,x_k]$ be the ring of polynomials in the variables $x_1,\dots,x_k$ with coefficients in $\mathbb{Q}$ and let $\mathbb{Q}(x_1,\dots,x_k)$ be the associated field of rational functions.

The *generating series of a sequence* $(f_n)_{n\in\mathbb{N}}$ is the (formal) power series in the variable x defined by $F(x)=\sum_{n\in\mathbb{N}} f_n x^n$. More generally, the generating series of a sequence $a(n_1,\dots,n_k)$ is a multivariate (formal) power series in the variables $x_1,\dots,x_k$ defined by $A(x_1,\dots,x_k)=\sum_{(n_1,\dots,n_k)\in\mathbb{N}^k} a(n_1,\dots,n_k)x_1^{n_1}\dots x_k^{n_k}$. In this article, we only consider power series whose coefficients belong to the field $\mathbb{Q}$. The set of such k-variate power series is denoted $\mathbb{Q}[[x_1,\dots,x_k]]$. Power series are naturally equipped with a sum and a product which generalize those of polynomials, for which $\mathbb{Q}[[x_1,\dots,x_k]]$ is a ring. We use the bracket notation for the coefficient extraction: $[x_1^{n_1}\dots x_k^{n_k}]A(x_1,\dots,x_k)=a(n_1,\dots,n_k)$. The *support* of $A\in\mathbb{Q}[[x_1,\dots,x_k]]$ is the set of $(n_1,\dots,n_k)$ such that $[x_1^{n_1}\dots x_k^{n_k}]A\neq 0$. The inverse $1/A$ of a series $A\in\mathbb{Q}[[x_1,\dots,x_k]]$ is well-defined when its constant term $[x_1^0\dots x_k^0]A$ is not zero. For instance, the inverse of $A(x_1,x_2)=1-x_1x_2^2$ is $\frac{1}{1-x_1x_2^2}=\sum_{n\geq 0} x_1^n x_2^{2n}$.

The *generating series of a language* $\mathcal{L}$ over the alphabet $\Sigma=\{a_1,\dots,a_k\}$ is the univariate power series $L(x)=\sum_{w\in\mathcal{L}} x^{|w|}=\sum_{n\in\mathbb{N}}\ell_n x^n$, where ℓ_n counts the number of words of length n in $\mathcal{L}$. Similarly the *multivariate generating series* of $\mathcal{L}$ defined by $L(x_{a_1},\dots,x_{a_k})=\sum_{(n_1,\dots,n_k)\in\mathbb{N}^k}\ell(n_1,\dots,n_k)x_{a_1}^{n_1}\dots x_{a_k}^{n_k}$ where $\ell(n_1,\dots,n_k)$ denotes the number of words w in $\mathcal{L}$ such that $|w|_{a_1}=n_1$, $|w|_{a_2}=n_2$, ..., and $|w|_{a_k}=n_k$, and $|w|_a$ denotes the number of occurrences of $a\in\Sigma$ in w. This way, we create one dimension per letter, so that each letter $a\in\Sigma$ has a corresponding variable x_a.

Observe that the univariate generating series of a language is exactly $L(x,\dots,x)$, obtained by setting each variable to x in its multivariate generating series.

▶ **Example 1.** The generating series of the language $\mathcal{P}$ of well-nested parentheses defined by the grammar $S\to aSbS+\varepsilon$ is $P(x_a,x_b)=\frac{1-\sqrt{1-4x_ax_b}}{2x_ax_b}$ and its counting series is[4] $P(x)=\frac{1-\sqrt{1-4x^2}}{2x^2}$. Indeed the production of the grammar translates to the equation $P(x_a,x_b)=x_ax_bP(x_a,x_b)^2+1$. This equation admits only one power series solution, namely $\frac{1-\sqrt{1-4x_ax_b}}{2x_ax_b}$.

[4] As for the inverse, the square root of a power series with nonzero constant term can be defined using the usual Taylor formula $\sqrt{1-x}=\sum_{n\geq 0}\frac{1}{(1-2n)\,4^n}\binom{2n}{n}x^n$.

A power series $A(x_1,\dots,x_k) = \sum_{n_1,\dots,n_k} a(n_1,\dots,n_k)x_1^{n_1}\dots x_k^{n_k}$ is *rational* if it satisfies an equation of the form: $P(x_1,\dots,x_k)A(x_1,\dots,x_k) = Q(x_1,\dots,x_k)$, with $P,Q \in \mathbb{Q}[x_1,\dots,x_k]$ and $P \neq 0$. The generating series (both univariate and multivariate) of regular languages (i.e., languages accepted by a finite state automaton) are rational power series [4]. It is well-known that the generating series can be effectively computed from a deterministic automaton accepting the language (see for instance [17, §I.4.2] for a detailed proof). For example, the multivariate generating series of the regular language $(abc)^*$ is $\frac{1}{1-x_a x_b x_c} = \sum_{n\geq 0} x_a^n x_b^n x_c^n$. Its univariate generating series is $\frac{1}{1-x^3}$.

The connection between rational languages and rational power series is not tight. For instance, the generating series of the non-regular context-free language $\{a^n b^n \ : \ n \geq 0\}$ is $\frac{1}{1-x_a x_b}$, which is rational. In fact, it has the same generating series as the regular language $(ab)^*$. Also there exist rational power series with coefficients in $\mathbb{N}$ which are not the generating series of any rational language. This is the case for $\frac{x+5x^2}{1+x-5x^2-125x^3}$ as shown in [4].

A power series $A(x_1,\dots,x_k)$ is *algebraic* if there exists a non-zero polynomial $P \in \mathbb{Q}[x_1,\dots,x_k,Y]$ such that $P(x_1,\dots,x_k,A(x_1,\dots,x_k)) = 0$. All rational series are algebraic.

▶ **Example 2.** The series $\frac{1}{1-x_1x_2} = \sum_{n\geq 0} x_1^n x_2^n$ is rational, as it satisfies the equation $(1-x_1x_2)A(x_1,x_2) = 1$. The series $A(x_1,x_2) = \sqrt{1-x_1x_2}$ is algebraic but not rational, since $A(x_1,x_2)^2 + (x_1x_2 - 1) = 0$ and there is no similar algebraic equation of degree 1.

The reader is referred to [34, 17] for a detailed account on rational and algebraic series.

In the same manner that rational series satisfy linear equations and algebraic series satisfy polynomial equations, holonomic series satisfy linear differential equations with polynomial coefficients. To give a precise definition, we need to introduce the formal partial differentiation of power series. The differential operator ∂_{x_i} with respect to the variable x_i is defined by

$$\partial_{x_i} A(x_1,\dots,x_k) = \sum_{n_1,\dots,n_k} n_i\, a(n_1,\dots,n_k)x_1^{n_1}\dots x_{i-1}^{n_{i-1}}x_i^{n_i-1}x_{i+1}^{n_{i+1}}\dots x_k^{n_k}.$$

The composed operator $\partial_{x_i}^j$ is inductively defined for $j \geq 1$ by $\partial_{x_i}^1 = \partial_{x_i}$ and $\partial_{x_i}^{j+1} = \partial_{x_i} \circ \partial_{x_i}^j$.

▶ **Definition 3** (see [33, 26])**.** *A power series $A(x_1,\dots,x_k)$ is* holonomic *or* D-finite[5] *if the $\mathbb{Q}(x_1,\dots,x_k)$-vector space spanned by the family $\{\partial_{x_1}^{i_1}\dots\partial_{x_k}^{i_k}A(x_1,\dots,x_k) \ : \ (i_1,\dots,i_k) \in \mathbb{N}^k\}$ has a finite dimension. Equivalently, for every variable $z \in \{x_1,\dots,x_k\}$, $A(x_1,\dots,x_k)$ satisfies a linear differential equation of the form $P_r(x_1,\dots,x_k)\partial_z^r A(x_1,\dots,x_k) + \dots + P_0(x_1,\dots,x_k)A(x_1,\dots,x_k) = 0$, where the P_i's are polynomials of $\mathbb{Q}[x_1,\dots,x_k]$ with $P_r \neq 0$.*

In the sequel, except for Section 5, we rely on the closure properties of the holonomic series and will not need to go back to Definition 3.

▶ **Example 4.** A simple example of holonomic series is $A(x) = e^{x^2} = \sum_{n\geq 0} \frac{x^{2n}}{n!}$. It is holonomic (in one variable) since it satisfies $\partial_x A(x) - 2xA(x) = 0$.

For a more involved example, consider the language $\mathcal{L}_3 = \{w \in \{a,b,c\}^* \ : \ |w|_a = |w|_b = |w|_c\}$, containing the words having the same number of occurrences of a's, b's and c's. This language is classically not context-free. Moreover there are $\binom{3n}{n,n,n}$ words of length $3n$ in $\mathcal{L}_3$

[5] A priori, these notions differ: a function $A(x_1,\dots,x_k)$ is called *D-finite* if all its partial derivatives $\partial_{x_1}^{n_1}\cdots\partial_{x_k}^{n_k}\cdot A$ generate a finite dimensional space over $\mathbb{Q}(x_1,\dots,x_k)$, and *holonomic* if the functions $x_1^{\alpha_1}\cdots x_k^{\alpha_k}\partial_{x_1}^{\beta_1}\cdots\partial_{x_k}^{\beta_k}\cdot A$ subject to the constraint $\alpha_1+\dots+\alpha_k+\beta_1+\dots+\beta_k \leq N$ span a vector space whose dimension over $\mathbb{Q}$ grows like $O(N^k)$. The equivalence of these notions is proved by deep results of Bernšteĭn [2] and Kashiwara [21, 36].

and the power series $\sum_{n\geq 0}\binom{3n}{n,n,n}x^n$ is transcendental [15, §7]. Its multivariate generating series $L_3(x_a,x_b,x_c)$ is equal to $\sum_{n\geq 0}\frac{(3\,n)!\,(x_ax_bx_c)^n}{(n!)^3}$ and satisfies the partial differential equation:

$$(27x_a^2x_bx_c-x_a)\partial_{x_a}^2 f(x_a,x_b,x_c)+(54x_ax_bx_c-1)\partial_{x_a}f(x_a,x_b,x_c)+6x_bx_cf(x_a,x_b,x_c)=0,$$

and the symmetric ones for the other variables x_b and x_c.

Holonomic series are an extension of the hierarchy we presented, as stated in the following proposition (see [12] for a proof, and [3] for bounds, algorithms and historical remarks).

▶ **Proposition 5.** *Multivariate algebraic power series are holonomic.*

In the univariate case[6], a power series $A(x)=\sum_n a_nx^n$ is holonomic if and only if its coefficients satisfy a linear recurrence of the form $p_r(n)a_{n+r}+\ldots+p_0(n)a_n=0$, where every p_i is a polynomial with rational coefficients [33, Th. 1.2].

We now focus on these closure properties.

▶ **Proposition 6** ([33])**.** *Multivariate holonomic series are closed under sum and product.*

Holonomic series are also closed under substitution by algebraic series as long as the resulting series is well-defined[7].

▶ **Proposition 7** ([26, Prop. 2.3])**.** *Let $A(x_1,\ldots,x_k)$ be a power series and let $G_i(y_1,\ldots,y_\ell)$ be algebraic power series such that $B(y_1,\ldots,y_\ell)=A(G_1(y_1,\ldots,y_\ell),\ldots,G_k(y_1,\ldots,y_\ell))$ is well-defined as a power series. If A is holonomic, then B is also holonomic.*

A sufficient condition for the substitution to be valid is that $G_i(0,\ldots,0)=0$ for all i (see [33, Th. 2.7]). For the case $G_1=\cdots=G_k=1$, called the *specialization to 1*, a sufficient condition is that for every index $(i_1,\ldots,i_k)$, $[x_1^{i_1}\ldots x_k^{i_k}]A$ is a polynomial in $y_1,\ldots,y_\ell$.

The Hadamard product is the coefficient-wise multiplication of power series. If the series $A(x_1,\ldots,x_k)$ and $B(x_1,\ldots,x_k)$ are the generating series of the sequences $a(n_1,\ldots,n_k)$ and $b(n_1,\ldots,n_k)$, the *Hadamard product* $A\odot B$ of A and B is the power series defined by

$$A\odot B(x_1,\ldots,x_k)=\sum_{(n_1,\ldots,n_k)\in\mathbb{N}^k}a(n_1,\ldots,n_k)b(n_1,\ldots,n_k)x_1^{n_1}\ldots x_k^{n_k}.$$

Observe that the support of $F\odot G$ is the intersection of the supports of F and G.

▶ **Theorem 8** ([25])**.** *Multivariate holonomic series are closed under Hadamard product.*

▶ **Example 9.** The generating series of the language $\mathcal{L}_3$ of Example 4, which is not context-free, can be expressed using the Hadamard product: since $\frac{1}{1-x_ax_bx_c}$ is the support series of the subset $\{(n,n,n):n\in\mathbb{N}\}$, and since $\frac{1}{1-(x_a+x_b+x_c)}$ is the multivariate series of all the words on $\{a,b,c\}$, we have $L_3(x_a,x_b,x_c)=\frac{1}{1-(x_a+x_b+x_c)}\odot\frac{1}{1-x_ax_bx_c}$, which is not algebraic.

One of our main technical contributions is to provide bounds on the sizes of the polynomials in the differential equations of the holonomic representation of the Hadamard product of two rational series $\frac{P_1}{Q_1}$ and $\frac{P_2}{Q_2}$: we prove that their maxdegree is at most $(kM)^{\mathcal{O}(k)}$ and that the logarithm of their largest coefficient is at most $(kM)^{\mathcal{O}(k^2)}(1+\log S_\infty)$, where M (resp. S_∞) is the maxdegree plus one (resp. largest coefficient) in P_1, Q_1, P_2 and Q_2.

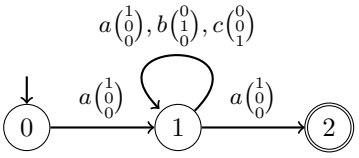
[6] The generalization of this equivalence to the multivariate case is not straightforward (see [26] for more details) and will not be used in this article.

[7] Note that the substitution of a power series into another power series might not yield a power series: for instance substituting x by $1+y$ in $\sum_{n\geq 0}x^n$ does not result in a power series as the constant term would be infinite.

3 Weakly-unambiguous Parikh automata

In this section, we introduce weakly-unambiguous Parikh automata and show that their multivariate generating series are holonomic. We establish that they accept the same languages as unambiguous two-way reversal bounded counter machines [19]. Finally, we prove that the class of accepted languages coincides with Massazza's RCM class [27, 8].

Parikh automata (PA for short) were introduced in [23, 24]. Informally, a PA is a finite automaton whose transitions are labeled by pairs $(a, \boldsymbol{v})$ where a is a letter of the input alphabet and $\boldsymbol{v}$ is a vector in $\mathbb{N}^d$. A run $q_0 \xrightarrow{a_1,\boldsymbol{v_1}} q_1 \xrightarrow{a_2,\boldsymbol{v_2}} q_2 \cdots q_{n-1} \xrightarrow{a_n,\boldsymbol{v_n}} q_n$ computes the word $a_1 \cdots a_n$ and the vector $\boldsymbol{v_1} + \cdots + \boldsymbol{v_n}$ where the sum is done component-wise. The acceptance condition is given by a set of final states and a semilinear set of vectors. A run is accepting if it reaches a final state and if its vector belongs to the semilinear set.



The PA depicted above, equipped with the semilinear constraint $\{(n_1, n_2, n_1 + n_2) \ : \ n_1, n_2 \geq 0\}$, accepts the set of words w over $\{a, b, c\}$ that start and end with a and that are such that $|w|_a + |w|_b = |w|_c$.

3.1 Semilinear sets and their characteristic series

A set $L \subseteq \mathbb{N}^d$ is *linear* if it is of the form $\boldsymbol{c} + P^* := \{\boldsymbol{c} + \lambda_1 \boldsymbol{p_1} + \cdots + \lambda_k \boldsymbol{p_k} : \lambda_1, \ldots, \lambda_k \in \mathbb{N}\}$, where $\boldsymbol{c} \in \mathbb{N}^d$ is the *constant* of the set and $P = \{\boldsymbol{p_1}, \ldots, \boldsymbol{p_k}\} \subset \mathbb{N}$ its set of *periods*. A set $C \subseteq \mathbb{N}^d$ is *semilinear* if it is a finite union of linear sets. For example, the semilinear set $\{(n_1, n_2, n_1 + n_2) \ : \ n_1, n_2 \geq 0\}$ is in fact a linear set $(0, 0, 0) + \{(1, 0, 1), (0, 1, 1)\}^*$.

In [13, 20], it is shown that every semilinear set admits an unambiguous presentation. A presentation $\boldsymbol{c} + P^*$ with $P = \{\boldsymbol{p_1}, \ldots, \boldsymbol{p_k}\}$ of a linear set L is unambiguous if for all $\boldsymbol{x} \in L$, the λ_i's such that $\boldsymbol{x} = \boldsymbol{c} + \lambda_1 \boldsymbol{p_1} + \cdots + \lambda_k \boldsymbol{p_k}$ are unique. An *unambiguous presentation of a semilinear set* is given by a disjoint union of unambiguous linear sets. A bound on the size of the equivalent unambiguous presentation is given in [9].

Semilinear sets are ubiquitous in theoretical computer science and admit numerous characterizations. They are the rational subsets of the commutative monoid $(\mathbb{N}^d, +)$, the unambiguous rational subset of $(\mathbb{N}^d, +)$ [13, 20], the Parikh images of context-free languages [30], the sets definable in Presburger arithmetic [31], the sets defined by boolean combinations of linear inequalities, equalities and equalities modulo constants.

For example, the semilinear set $\{(2n, 3n, 5n) \ : \ n \geq 0\}$ is equal to $(0, 0, 0) + \{(2, 3, 5)\}^*$. It is also the Parikh image of the regular (hence context-free) language $(aabbbccccc)^*$. In $(\mathbb{N}, +)$, it is defined by the Presburger formula $\phi(x, y, z) = \exists w, x = w + w \wedge y = w + w + w \wedge z = w + w + w + w + w$. Finally it is characterized in $\mathbb{N}^3$ by the equalities $3x = 2y$ and $5x = 2z$.

For a semilinear set $C \subseteq \mathbb{N}^d$, we consider its *characteristic generating series* $C(x_1, \ldots, x_d)$ defined by $\sum_{(i_1,\ldots,i_d)\in C} x_1^{i_1} \cdots x_d^{i_d}$. It is well-known [13, 20] that this power series is rational[8].

[8] The characteristic series of an unambiguous linear set $\boldsymbol{c} + P^* \subseteq \mathbb{N}$ with $P = \{\boldsymbol{p_1}, \ldots, \boldsymbol{p_k}\}$ is $x_1^{\boldsymbol{c}(1)} \ldots x_d^{\boldsymbol{c(d)}} \prod_{i=1}^{k} \left(1 - x_1^{\boldsymbol{p_i}(1)} \cdots x_d^{\boldsymbol{p_i}(d)}\right)^{-1}$ and hence is rational. As an unambiguous semilinear set is the disjoint union of unambiguous linear sets, its characteristic series is the sum of their series and it is therefore rational.

3.2 Weakly-unambiguous PAs and their generating series

We now introduce PA and their weakly-unambiguous variant. We discuss the relationship with the class of unambiguous PA introduced by Cadilhac et al. in [7] and the closure properties of this class.

A *Parikh automaton* of dimension $d \geq 1$ is a tuple $\mathcal{A} = (\Sigma, Q, q_I, F, C, \Delta)$ where Σ is the alphabet, Q is the set of states, $q_I \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, $C \subseteq \mathbb{N}^d$ is the semilinear constraint and $\Delta \subseteq Q \times (\Sigma \times \mathbb{N}^d) \times Q$ is the transition relation.

A run of the automaton is a sequence $q_0 \xrightarrow{a_1,\boldsymbol{v_1}} q_1 \xrightarrow{a_2,\boldsymbol{v_2}} q_2 \cdots q_{n-1} \xrightarrow{a_n,\boldsymbol{v_n}} q_n$ where for all $i \in [1,n]$, $(q_{i-1},(a_i,\boldsymbol{v_i}),q_i)$ is a transition in Δ. The run is labeled by the pair $(a_1 \cdots a_n, \boldsymbol{v_1} + \cdots + \boldsymbol{v_n}) \in \Sigma^* \times \mathbb{N}^d$. It is accepting if $q_0 = q_I$, the state q_n is final and if the vector $\boldsymbol{v_1} + \cdots + \boldsymbol{v_n}$ belongs to C. The word w is then said to be accepted by $\mathcal{A}$. The language accepted by $\mathcal{A}$ is denoted by $\mathcal{L}(\mathcal{A})$.

To define a notion of size for a PA, we assume that the constraint set is given by an unambiguous presentation $\uplus_{i=1}^{p} \boldsymbol{c_i} + P_i^*$. We denote by $|\mathcal{A}| := |Q| + |\Delta| + p + \sum_i |P_i|$ and by $\|\mathcal{A}\|_\infty$ the maximum coordinate of a vector appearing in Δ, the $\boldsymbol{c_i}$'s and the P_i's.

▶ **Definition 10.** *A Parikh automaton is said to be* weakly-unambiguous *if for every word there is at most one accepting run.*

A language is *inherently weakly-ambiguous* if it cannot be accepted by any weakly-unambiguous PA. The language $\mathcal{S}$ (defined in Section 4.1) is an example[9] of a language accepted by a non-deterministic PA which is inherently weakly-ambiguous.

▶ Remark 11. We consider here the standard notion of unambiguity for finite state machines. However we decided to use the name *weakly-unambiguous* to avoid the confusion with the class of unambiguous PA which appears in the literature. This class was introduced by Cadilhac et al. in [7] for constraint automata, a model equivalent to PA and was latter defined directly on PAs. This notion of unambiguity is more restrictive than ours: they call a Parikh automaton *unambiguous* if the underlying automaton on letters, where the vectors have been erased, is unambiguous. Clearly such automata are weakly-unambiguous. However the converse is not true. Consider the language $\mathcal{L} = \{c^n w \ : \ w = x_1 x_2 \cdots x_m \in \{a,b\}^* \ \wedge \ m \geq n > 0 \wedge \ |x_1 x_2 \cdots x_n|_a < |x_1 x_2 \cdots x_n|_b\}$ over the alphabet $\{a,b,c\}$. Using results from [7], one can show that it is not recognized by any unambiguous Parikh automata. However, it is accepted by the weakly-unambiguous automaton depicted in Fig. 1 below with the semilinear $\{(n_1,n_2,n_3) \ : \ n_1 = n_2 + n_3 \text{ and } n_2 < n_3\}$.

The lack of expressivity of unambiguous PAs is counter-balanced by their closure under boolean operations, which is explained by their link with a class of deterministic PA [7, 14]. It was pointed out to us by a reviewer that the class of weakly-unambiguous PA is briefly considered, under the name OneCA, in Cadilhac's PhD thesis [5, p. 117], where only basic properties are established, in particular the strict inclusion of unambiguous PA in this class.

Using a standard product construction when the vectors are concatenated and using the concatenation of the constraints, it is easy to show that weakly-unambiguous PA are closed under intersection. In [8], the authors claim that the class[10] is closed under union. However their construction has an irrecoverable flaw and we do not know if weakly-unambiguous PA are closed under union or under complementation.

[9] Using the equivalences between weakly-unambiguous PA and RCM established in Proposition 13 and PA and RBCM [23, 6], it also gives an example of a language accepted by a RBCM with a non-holonomic generating series (strengthening Theorem 12 of [8]) and a witness for the strict inclusion of RCM in RBCM announced in Theorem 11 of [8]. Remark that their proof of this theorem only shows that there exists no recursive translation from RBCM to RCM.

[10] Actually, their claim is for the class RCM, which we will show to be equivalent in Section 3.4.

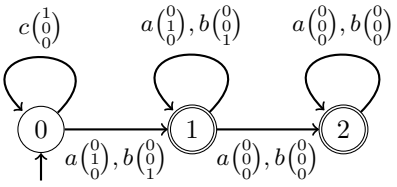


Figure 1 A weakly-unambiguous Parikh automaton accepting the language $\mathcal{L} = \{c^n w \ : \ w = x_1x_2\cdots x_m \in \{a,b\}^* \ \wedge \ m \geq n > 0 \wedge \ |x_1x_2\cdots x_n|_a < |x_1x_2\cdots x_n|_b\}$ over the alphabet $\{a,b,c\}$ with the semilinear constraint $\{(n_1,n_2,n_3) \ : \ n_1 = n_2 + n_3 \text{ and } n_2 < n_3\}$.

We now give a very short proof of the fact that weakly-unambiguous languages in PA have holonomic generating series. The idea of the proof can be traced back to [25]. A similar proof was given in [27] for languages in the class RCM but using the closure under algebraic substitutions instead of specialization (see Remark 25).

Our approach puts into light a different multivariate power series associated with a weakly-unambiguous PA $\mathcal{A}$ of dimension d. The multivariate *weighted generating series* $G(x, y_1, \ldots, y_d)$ of $\mathcal{A}$ is such that for all indices $(n, i_1, \ldots, i_d)$, $[x^n y_1^{i_1} \ldots y_d^{i_d}]G$ counts the number of words of length n accepted by $\mathcal{A}$ with a run labeled by the vector $(i_1, \ldots, i_d)$.

▶ **Proposition 12.** *The generating series of the language recognized by a weakly-unambiguous Parikh automaton is holonomic.*

Proof. Let $\mathcal{A}$ be a weakly-unambiguous PA with a constraint set $C \subseteq \mathbb{N}^d$. We first prove that its weighted series $G(x, y_1, \ldots, y_d)$ is holonomic. As holonomic series are closed under Hadamard product (see Theorem 8), it suffices to express G as the Hadamard product of two rational series $\overline{A}$ and $\overline{C}$ in the variables $x, y_1, \ldots, y_d$.

The first series $\overline{A}(x, y_1, \ldots, y_d)$ is such that for all $n, i_1, \ldots, i_d \geq 0$, $[x^n y_1^{i_1} \ldots y_d^{i_d}]\overline{A}$ counts the number of runs of $\mathcal{A}$ starting in q_I, ending in a final state and labeled with a word of length n and the vector $(i_1, \ldots, i_d)$. Note that we do not require that $(i_1, \ldots, i_d)$ belongs to C. As this series simply counts the number of runs in an automaton, its rationality is proved via the standard translation of the automaton into a linear system of equations.

For the second series, we take $\overline{C}(x, y_1, \ldots, y_d) := \frac{1}{1-x}\widetilde{C}(y_1, \ldots, y_d)$ where $\widetilde{C}$ is the support series of C, which is rational (see [13, 20]). A direct computation yields that for all $n, i_1, \ldots, i_d \geq 0$, $[x^n y_1^{i_1} \ldots y_d^{i_d}]\overline{C}$ is equal to 1 if $(i_1, \ldots, i_d)$ belongs to C and 0 otherwise.

The Hadamard product of $\overline{A}$ and $\overline{C}$ counts the number of runs accepting a word of length n with the vector $(i_1, \ldots, i_d)$. As $\mathcal{A}$ is weakly-unambiguous, this quantity is equal to the number of words of length n accepted with this vector. Hence $G = \overline{A} \odot \overline{C}$.

The univariate series $A(x)$ of $\mathcal{A}$ is equal to $G(x, 1, \ldots, 1)$. Indeed, for all $n \geq 0$, $[x^n]G(x, 1, \ldots, 1) = \sum_{\boldsymbol{i}=(i_1,\ldots,i_d)} [x^n y_1^{i_1} \ldots y_d^{i_d}]G(x, y_1, \ldots, y_d)$ is the sum over all vectors $\boldsymbol{i} \in \mathbb{N}^d$ of the number of words of length n accepted with the vector $\boldsymbol{i}$. As $\mathcal{A}$ is weakly-unambiguous, each word is accepted with at most one vector and this sum is therefore equal to the total number of accepted words of length n. Thanks to Proposition 7, $A(x) = G(x, 1, \ldots, 1)$ is holonomic. ◀

3.3 Equivalence with unambiguous reversal bounded counter machines

A *k-counter machine* [19] is informally a Turing machine with one read-only tape that contains the input word, and k counters. Reading a letter a on the input tape, in a state q, the machine can check which of its counters are zero, increment or decrement its counters, change

its state, and move its read head one step to the left or right, or stay on its current position. Note that the machine does not have access to the exact value of its counters. A k-counter machine is said (m, n)-*reversal bounded* if its reading head can change direction between left and right at most m times, and if every counter can alternate between incrementing and decrementing at most n times each. Finally, a *reversal bounded counter machine* (RBCM) is a k-counter machine which is (m, n)-reversal bounded for some m and n. A RBCM is *unambiguous* if for every word there is at most one accepting computation.

RBCM are known[11] to recognize the same languages as Parikh automata (see [24, 23, 6]). This equality does not hold anymore for their deterministic versions [6, Prop. 3.14]. However, the proof of the equivalence for the general case can be slightly modified to preserve unambiguity.

▶ **Proposition 13.** *The class of languages accepted by unambiguous* RBCM *and weakly-unambiguous* PA *coincide.*

Proof sketch. Unambiguous RBCMs are shown to be equivalent to one-way unambiguous RBCMs. In turn these are shown to be equivalent to weakly-unambiguous PA with ε-transitions, which in turn are equivalent to weakly-unambiguous PA. This ε-removal step needs to be adapted to preserve weak-unambiguity. ◀

3.4 Equivalence with RCM

If we fix an alphabet $\Gamma = \{a_1, \ldots, a_d\}$ with the ordering $a_1 < \cdots < a_d$ on the letters, we can associate with every semilinear set C of dimension d, the language $[C] = \{w \in \Gamma^* : (|w|_{a_1}, \ldots, |w|_{a_d}) \in C\}$ of words whose numbers of occurrences of each letter satisfy the constraint expressed by C. For instance, if we take the semilinear set $C_0 = \{(n, m, n, m) : n, m \geq 0\}$ and the alphabet $\{a, b, c, d\}$ ordered by $a < b < c < d$, $[C_0]$ consists of all words having as many a's as c's and as many b's as d's.

A language $\mathcal{L}$ over Σ belongs to RCM if there exist a regular language $\mathcal{R}$ over $\Gamma = \{a_1, \ldots, a_d\}$, a semilinear set[12] $C \subseteq \mathbb{N}^d$ and a length preserving morphism $\mu\colon \Gamma^* \longrightarrow \Sigma^*$, that is injective over $\mathcal{R} \cap [C]$, so that $\mathcal{L} = \mu(\mathcal{R} \cap [C])$. For example, $\mathcal{L}_{abab} = \{a^n b^m a^n b^m : n, m \in \mathbb{N}\}$ can be shown to be in RCM by taking $\Gamma = \{a, b, c, d\}, \Sigma = \{a, b\}$, $\mu(a) = \mu(c) = a, \mu(b) = \mu(d) = b$, $\mathcal{R} = a^* b^* c^* d^*$ and the semilinear set C_0 defined in the previous paragraph.

▶ **Theorem 14.** *$\mathcal{L} \in$ RCM iff $\mathcal{L}$ is recognized by a weakly-unambiguous Parikh automaton.*

Proof sketch. Every language in RCM can be accepted by a weakly-unambiguous PA that guesses the underlying word over Γ: the weak-unambiguity is guaranteed by the injectivity of the morphism. Conversely a language accepted by a weakly-unambiguous PA is in RCM by taking for $\mathcal{R}$ the set of runs of the PA and translating the constraint: the injectivity of the morphism is guaranteed by the weak-unambiguity of the PA. ◀

In [8], the authors conjectured that the class RCM contains the one-way deterministic RBCM. From Theorem 14 and Proposition 13 we get a stronger result:

▶ **Corollary 15.** *The languages in RCM are the languages accepted by unambiguous RBCM.*

[11] The proof in [23] contains a patchable error, that was corrected in [6].
[12] Our definition may seem a little more general than Massazza's, which only uses semilinear defined without modulo constraints, but it can be shown that the classes are equivalent.

3.5 Weakly-unambiguous pushdown Parikh automata

A *pushdown Parikh automaton* (ℙA for short) is a PA where the finite automaton is replaced by a pushdown automaton. A *weakly-unambiguous* ℙA has at most one accepting run for each word. Most results obtained previously can be adapted for weakly-unambiguous ℙA. However, unsurprisingly, the class of languages accepted by weakly-unambiguous ℙA is not closed under union and intersection. This can be shown using the inherent weak-ambiguity of the language $\mathcal{D}$ proved in Section 4.1. The closure under complementation is left open.

▶ **Proposition 16.** *The generating series of a weakly-unambiguous* ℙA *is holonomic.*

Proof. The proof is almost identical to the proof of Proposition 12. The only difference is that the series $\overline{A}$ is algebraic and not rational. Indeed it counts the number of runs in a pushdown automaton and the language of runs is a deterministic context-free language even if the pushdown automaton is not deterministic. ◀

Remark that using the same techniques, we can prove that the generating series of weakly-unambiguous Parikh tree automata are holonomic. As we proved that all these series are also generating series of ℙAs, we do not elaborate on this model in this extended abstract.

RBCMs can be extended with a pushdown storage to obtain a *RBCM with a stack* [19].

▶ **Theorem 17.** *Weakly-unambiguous* ℙA *are equivalent to unambiguous one-way RBCM with a stack.*

Proof sketch. We first establish that unambiguous one-way RBCM with a stack are equivalent to weakly-unambiguous ℙA with ε-transitions. Contrarily to the PA case, the removal of ε-transitions is quite involved and uses weighted context-free grammars. ◀

The class LCL of [27] is defined[13] as RCM is, except that the regular language is replaced by an unambiguous context-free[14] language. Similarly to the PA case, one can prove:

▶ **Proposition 18.** *LCL is the set of languages accepted by weakly-unambiguous* ℙA*.*

4 Examples of inherently weakly-ambiguous languages

There is a polynomial-time algorithm to decide whether a given PA is weakly-unambiguous. But inherent weak-unambiguity is undecidable, as a direct application of a general theorem from [18]. This emphasizes that inherent weak-ambiguity is a difficult problem in general.

4.1 Two examples using an analytic criterion

Following an idea from Flajolet [15] for context-free languages, the link between weakly-unambiguous PA and holonomic series yields sufficient criteria to establish inherent weak-ambiguity, of analytic flavor: the contraposition of Proposition 12 indicates that if $\mathcal{L}$ is recognized by a PA but its generating series is not holonomic, then $\mathcal{L}$ is inherently weakly-ambiguous. Hence, any criterion of non-holonomicity can be used to establish the inherent weak-ambiguity. Many such criteria can be obtained when considering the generating series as analytic functions (of complex variables). See [15, 16] for several examples. For the presentation of this method in this extended abstract, we only rely on the following property:

[13] In [27], LCL is defined without the injective morphism but we adapt it following [8].
[14] Using deterministic context-free languages instead of unambiguous ones in the definition of LCL would result in the same class.

▶ **Proposition 19** ([33]). *A holonomic function in one variable has finitely many singularities.*

Our first example is the language $\mathcal{D}$, defined over the alphabet $\{a, b\}$ as follows:

$$\mathcal{D} = \{\underline{n_1}\ \underline{n_2} \dots \underline{n_k}\ :\ k > 0,\ n_1 = 1 \text{ and } \exists j < k, n_{j+1} \neq 2n_j\},\ \text{where } \underline{n} = a^n b.$$

This language is recognized by a weakly-ambiguous Parikh automaton, which guesses the correct j, and then verifies that $n_{j+1} \neq 2n_j$. Let $\overline{\mathcal{D}} = ab(a^*b)^* \setminus \mathcal{D}$, and suppose by contradiction that $\mathcal{D}$ can be recognized by a weakly-unambiguous PA. Then its generating series should be holonomic by Proposition 12. Since the generating series of $\overline{\mathcal{D}}$ is $\overline{D}(x_a, x_b) = \frac{x_a x_b}{1 - \frac{x_b}{1-x_a}} - D(x_a, x_b)$, it should be holonomic too. Looking closely at the form of the words of $\overline{\mathcal{D}}$, we get that its generating series is $\sum_{k \geq 1} x_a^{2^k - 1} x_b^k$. It is not holonomic as $x\overline{D}(x, 1) + x$ has infinitely many singularities, see [15, p. 296–297].

Our second example is Shamir's language $\mathcal{S} = \{a^n b v_1 a^n v_2 : n \geq 1, v_1, v_2 \in \{a, b\}^*\}$. One can easily design a PA recognizing $\mathcal{S}$, where one coordinate stands for the length of the first run of a's and the other one for the second run of a's, the automaton guessing when the second run starts. Flajolet proved that $\mathcal{S}$ is inherently ambiguous as a context-free language, since its generating series $S(z) = \frac{z(1-z)}{1-2z} \sum_{n \geqslant 1} \frac{z^{2n}}{1-2z+z^{n+1}}$ has an infinite number of singularities [15, p. 296–297]. This also yields its inherent weak-ambiguity as a PA language.

4.2 Limit of the method: an example using pumping techniques

As already mentioned, the analytic method presented is not always sufficient to prove inherent ambiguity. In this section, we develop an example where it does not apply. We consider the following language $\mathcal{L}_{even}$, which is accepted both by a deterministic pushdown automaton and a non-deterministic PA (where $\underline{n} = a^n b$ as in Section 4.1):

$$\mathcal{L}_{even} = \left\{\underline{n_1}\ \underline{n_2} \dots \underline{n_{2k}}\ :\ k \in \mathbb{N},\ \forall i \leq 2k, n_i > 0,\ \text{and } \exists j \leq k, n_{2j} = n_{2j-1}\right\}.$$

In other words, $\mathcal{L}_{even}$ is the language of sequences of encoded numbers having two consecutive equal values, the first one being at an odd position. This language is accepted by a non-deterministic PA but is also deterministic context-free. This means that its generating series is algebraic and hence holonomic. This puts it out of the reach of our analytic method.

In this section we establish the following result:

▶ **Theorem 20.** *The language $\mathcal{L}_{even}$ is inherently weakly-ambiguous as a PA language.*

The remainder of this section is devoted to sketch the proof of this proposition. By contradiction, we suppose that $\mathcal{L}_{even}$ is recognized by a weakly-unambiguous PA $\mathcal{A}$.

An *a-piece* ω of $\mathcal{A}$ is a non-empty simple path of a-edges in $\mathcal{A}$, starting and ending at the same state: the states of the path are pairwise distinct, except for its extremities. The *origin* of w is its starting (and ending) state. Let $\Pi(\mathcal{A})$ be the (finite) set of a-pieces in $\mathcal{A}$.

We see a run in $\mathcal{A}$ as a sequence of transitions forming a path in $\mathcal{A}$. An *a-subpath* of a run R in $\mathcal{A}$ is a maximal consecutive subsequence of R whose transitions are all labeled by a's, that is, it cannot be extended further to the left nor to the right in R using a's.

Let R be an accepting run in $\mathcal{A}$. One can show that every a-subpath S of R can be decomposed as $S = w_1 \sigma_1^{s_1} w_2 \sigma_2^{s_2} \cdots w_f \sigma_f^{s_f} w_{f+1}$, where the σ_i's are a-pieces of $\Pi(\mathcal{A})$, the s_i's are positive integers, and the w_i's are paths not using twice the same state. Moreover, this decomposition is unique if we add the condition that if $w_i = \varepsilon$, then $\sigma_i \neq \sigma_{i-1}$ and the only state in common in w_i and σ_i is the origin of σ_i. This is done by repeatedly following the path until a state q is met twice, factorizing this segment of the form $w\sigma$, where σ is an a-piece of origin q. We call this decomposition the *canonical form* of S, and the *signature* of S is the tuple $(w_1, \sigma_1, w_2, \dots, w_f, \sigma_f, w_{f+1})$, i.e., we dropped the s_i's of the canonical form.

From the weak-unambiguity of $\mathcal{A}$ we can prove that there are at most c distinct possible signatures, where c only depends on $\mathcal{A}$, and that f is always at most $|Q_{\mathcal{A}}|$.

Ramsey's Theorem [32] guarantees that there exists an integer r such that any complete undirected graph with at least r vertices, whose edges are colored using c^2 different colors, admits a monochromatic triangle. We fix two positive integers n and k sufficiently large, which will be chosen later on, depending on $\mathcal{A}$ only. For $\ell \in \{1, \dots, r\}$, let w_ℓ be the word $w_\ell = \underline{n_1}\,\underline{n_2} \dots \underline{n_{2r}}$, where $n_i = n$ for odd i and $n_{2i} = n + k$ if $i \neq \ell$ and $n_{2\ell} = n$. Each w_ℓ is in $\mathcal{L}_{even}$, with a match at position 2ℓ only. By weak-unambiguity, each w_ℓ has a unique accepting run $\mathcal{R}_\ell$ in $\mathcal{A}$, each such run having $2r$ a-subpaths by construction. For $i \neq j$ in $\{1, \dots, r\}$, let λ_{ij} be the signature of the $2j$-th a-subpath of $\mathcal{R}_i$, which is a path of length $n + k$ by definition. The complete undirected graph of vertex set $\{1, \dots, r\}$ where each edge ij, with $i < j$, is colored by the pair $(\lambda_{ij}, \lambda_{ji})$ admits a monochromatic triangle of vertices $\alpha < \beta < \gamma$. In particular, $\lambda_{\alpha\beta} = \lambda_{\alpha\gamma}$ and $\lambda_{\gamma\alpha} = \lambda_{\gamma\beta}$.

We choose $k = \mathrm{lcm}(\{|\sigma| : \sigma \in \Pi(\mathcal{A})\})$, and n sufficiently large so that any a-subpath of an accepting run contains an a-piece σ repeated at least $k + 1$ times. This is possible as the w_i's have bounded length, and there are at most $|Q_{\mathcal{A}}| + 1$ of them. Hence, the 2γ-th a-subpath of w_α contains a a-piece σ that is repeated more than s times, where $s = k/|\sigma|$. As $\lambda_{\alpha\beta} = \lambda_{\alpha\gamma}$, the piece σ is also in the 2β-th a-subpath of w_α. If we alter the accepting path $\mathcal{R}_\alpha$ into $\mathcal{R}'_\alpha$ by looping s more times in σ in the a-subpath at position 2β and s less times in σ at position 2γ, we obtain a run for the word $w = \cdots \underbrace{\underline{n}\,\underline{n}}_{2\alpha} \cdots \underline{n}\,\underbrace{\underline{n+2k}}_{2\beta} \cdots \underbrace{\underline{n}\,\underline{n}}_{2\gamma} \cdots$. This run is accepting as the PA computes the same vector as for w_α, by commutativity of vectors addition. And the signatures remain unchanged, as there are sufficiently many repetitions of σ at position 2γ in w_α. Similarly, as $\lambda_{\gamma\alpha} = \lambda_{\gamma\beta}$, we can alter the accepting path $\mathcal{R}_\gamma$ into an accepting path $\mathcal{R}'_\gamma$ of same signatures as $\mathcal{R}_\gamma$ for the same word w, by removing $s' = k/|\sigma'|$ iterations of an a-piece σ' at position 2α and adding them at position 2β.

We have built two paths $\mathcal{R}'_\alpha$ and $\mathcal{R}'_\gamma$ that both accept the same word w. Therefore, as $\mathcal{A}$ is weakly-unambiguous, they are equal. As the signatures have not changed, this implies that the signature at position 2α in $\mathcal{R}'_\alpha$ is $\lambda_{\gamma\alpha}$, which is equal to $\lambda_{\gamma\beta}$ (monochromaticity), which is equal to $\lambda_{\alpha\beta}$ ($\mathcal{R}'_\alpha$ and $\mathcal{R}'_\gamma$ have same signatures). This is a contradiction as we could remove one a-piece at position 2α in w_α and add it at position 2β, while computing the same vector with the same starting and ending states: but this word is not in $\mathcal{L}_{even}$.

▶ Remark 21. The proof relies on manipulations of paths in the automaton, and we only use the commutativity of the addition for the vector part. Thus, it still holds if we consider automata where we use a recursively enumerable set instead of a semilinear set for acceptance.

5 Algorithmic consequence of holonomicity

Generating series of languages have already been used to obtain efficient algorithms on unambiguous models of automata. For instance, they were used by Stearns and Hunt as a basic tool to obtain bounds on the length of a word witnessing the non-inclusion between two unambiguous word automata [35]. More precisely, the proof in [35] relies on the recurrence equation satisfied by the coefficients of the generating series (which is guaranteed to exist by holonomicity in one variable). In the rational case, this recurrence relation can be derived from the automaton and does not require advanced results on holonomic series. In this section, our aim is to obtain a similar bound for the inclusion problem for weakly-unambiguous Parikh automata. The inclusion problem for RCM (and hence for weakly-unambiguous PA) is shown to be decidable in [8] but no complexity bound is provided. Note that this problem

is known to be undecidable for non-deterministic PA [19]. We follow the same approach as for the rational case [35] and for RCM [8]. In stark contrast with the rational case, it is necessary to closely inspect holonomic closure properties in order to give concrete bounds.

Fix $\mathcal{A}$ and $\mathcal{B}$ two weakly-unambiguous PA. We can construct a weakly-unambiguous PA $\mathcal{C}$ accepting $L(\mathcal{A}) \cap L(\mathcal{B})$. We rely on the key fact that the series $D(x) = A(x) - C(x)$ counts the number of words of length n in $L(\mathcal{A}) \setminus L(\mathcal{B})$. In particular, $L(\mathcal{A}) \subseteq L(\mathcal{B})$ if and only if $D(x) = 0$.

As $D(x)$ is the difference of two holonomic series, it is holonomic. As equality between holonomic series is decidable, [8] concludes that the problem is decidable. But without further analysis, no complexity upper-bound can be derived. The coefficients of $D(x) = \sum_{n\geq 0} d_n x^n$ satisfy a recurrence equation of the form $p_0(n)d_n = \sum_{k=1}^{r} p_k(n)d_{n-k}$ for $n \geq r$ with $p_0(x) \neq 0$. This equation fully determines d_n in terms of its r previous values $d_{n-1}, \ldots, d_{n-r}$, *provided that* $p_0(n) \neq 0$. In particular, if the r previous values are all equal to 0, then $d_n = 0$. Consequently, if d_n is equal to 0 for all $n \leq r + R$ where R denotes the largest positive root of p_0 (which is bounded from above by its ∞-norm, as a polynomial on $\mathbb{Z}$) then[15] $D(x) = 0$.

Taking $W := r + R$, we have that if $L(\mathcal{A}) \not\subseteq L(\mathcal{B})$ then there exists a word witnessing this non-inclusion of length at most W. We now aim at computing an upper-bound on W on the size of the inputs $\mathcal{A}$ and $\mathcal{B}$.

For this, we first bound the order of the linear recurrence satisfied by $A(x)$ and $C(x)$, as well as the degrees and norm of the polynomials involved. This is stated in Proposition 22, whose proof follows the one of Proposition 12, while establishing such bounds for the multivariate Hadamard product and the specialization to 1.

▶ **Proposition 22.** *The generating series $A(x)$ of a weakly-unambiguous PA $\mathcal{A}$ of dimension $d \geq 1$ satisfies a non-trivial linear differential equation $q_s(x)\partial_x^s A(x) + \cdots + q_0(x)A(x) = 0$, with $s \leq ((d+1)|\mathcal{A}|\,\|\mathcal{A}\|_\infty)^{O(d)}$ and for all $i \in [0, s]$, $\deg(q_i) \leq ((d+1)|\mathcal{A}|\,\|\mathcal{A}\|_\infty)^{O(d)}$ and $\log \|q_i\|_\infty \leq ((d+1)|\mathcal{A}|\,\|\mathcal{A}\|_\infty)^{O(d^2)}$ using the notations of Section 3.2.*

Finally, we transfer these bounds to the series $D(x)$ of $L(\mathcal{A}) \setminus L(\mathcal{B})$ using the analysis of [22] for the sum of holonomic series in one variable.

▶ **Theorem 23.** *Given two weakly-unambiguous PA $\mathcal{A}$ and $\mathcal{B}$ of respective dimensions $d_\mathcal{A}$ and $d_\mathcal{B}$, if $L(\mathcal{A})$ is not included in $L(\mathcal{B})$ then there exists a word in $L(\mathcal{A}) \setminus L(\mathcal{B})$ of length at most $2^{2^{O(d^2 \log(dM))}}$ where $d = d_\mathcal{A} + d_\mathcal{B}$ and $M = |\mathcal{A}|\,|\mathcal{B}|\,\|\mathcal{A}\|_\infty\,\|\mathcal{B}\|_\infty$.*

Using the bound of Theorem 23, the inclusion problem can be solved in triply exponential time by a naive counting of all words up to the bound. Using dynamic programming to compute the number of accepted words, we can decide inclusion in doubly exponential time.

▶ **Corollary 24.** *Given two weakly-unambiguous PA $\mathcal{A}$ and $\mathcal{B}$ of dimensions $d_\mathcal{A}$ and $d_\mathcal{B}$, we can decide if $L(\mathcal{A})$ is included in $L(\mathcal{B})$ in time $2^{2^{O(d^2 \log(dM))}}$ where $d = d_\mathcal{A} + d_\mathcal{B}$ and $M = |\mathcal{A}|\,|\mathcal{B}|\,\|\mathcal{A}\|_\infty\,\|\mathcal{B}\|_\infty$.*

▶ Remark 25. In [8], the authors propose a different construction to prove the holonomicity of the generating series of languages in RCM. This proof uses the closure of holonomic series under Hadamard product and algebraic substitution $x_1 = x_2 = \cdots = x_n = x$. It is natural to wonder if this approach would lead to better bounds in Proposition 22 (using the

[15] The proof of Theorem 7 in [8] wrongly suggests that we can take the order of the differential equation for D as a bound on the length of a witness for non-inclusion. In general, this is not the case. For instance consider $D(x) = x^{1000}$ which satisfies the first-order differential equation $1000D(x) - x\partial_x D(x) = 0$. It is clear that the coefficients $D_0 = 0$ and $D_1 = 0$ are not enough to decide that D is not zero.

equivalence between weakly-unambiguous PA and RCM). It turns out that the operation $x_1 = x_2 = \cdots = x_n = x$ is more complicated than it seems at a first glance. Indeed, to our knowledge, no proof of the closure under algebraic substitution explains what happens if, during the substitution process, the equations become trivial. This issue can be overcome by doing the substitution step by step: $x_2 = x_1$, then $x_3 = x_1$, etc. However, this naive approach would produce worse bounds.

6 Perspectives

The bounds obtained in Section 5 are derived directly from constructions given in the proofs of the closure properties. In particular, we did not use any information on the special form of our series. The bounds are certainly perfectible using more advanced tools from computer algebra. Also it seems that the complexity of the closure under the algebraic substitution deserves more investigation, as discussed in Remark 25.

A more ambitious perspective is to find larger classes of automata whose generating series are holonomic. This would certainly require new ideas, as for instance any holonomic power series with coefficients in $\{0, 1\}$ is known to be the characteristic series of some semilinear set [1].

References

1 Jason P. Bell and Shaoshi Chen. Power series with coefficients from a finite set. *J. Comb. Theory, Ser. A*, 151:241–253, 2017. doi:10.1016/j.jcta.2017.05.002.

2 Joseph N. Bernstein. Analytic continuation of generalized functions with respect to a parameter. *Funct. Anal. Appl.*, 6(4):273–28, 1972. doi:10.1007/BF01077645.

3 Alin Bostan, Frédéric Chyzak, Grégoire Lecerf, Bruno Salvy, and Éric Schost. Differential equations for algebraic functions. In *International Symposium on Symbolic and Algebraic Computation, ISSAC 2007*, pages 25–32. ACM, 2007. doi:10.1145/1277548.1277553.

4 Mireille Bousquet-Mélou. Rational and algebraic series in combinatorial enumeration. In *International Congress of Mathematicians (ICM 2006)*, volume 3, pages 789–826. Eur. Math. Soc., Zürich, 2006. doi:10.4171/022-3/40.

5 Michaël Cadilhac. *Automata with a semilinear constraint.* PhD thesis, Université de Montréal, 2013.

6 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Affine Parikh automata. *RAIRO – Theor. Inf. and Applic.*, 46(4):511–545, 2012. doi:10.1051/ita/2012013.

7 Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Unambiguous constrained automata. *Int. J. Found. Comput. Sci.*, 24(7):1099–1116, 2013. doi:10.1142/S0129054113400339.

8 Giusi Castiglione and Paolo Massazza. On a class of languages with holonomic generating functions. *Theor. Comput. Sci.*, 658:74–84, 2017. doi:10.1016/j.tcs.2016.07.022.

9 Dmitry Chistikov and Christoph Haase. The taming of the semi-linear set. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016*, volume 55 of *LIPIcs*, pages 128:1–128:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.128.

10 Noam Chomsky and Marcel-Paul Schützenberger. *The Algebraic Theory of Context-Free Languages*, volume 35 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 1963. doi:10.1016/S0049-237X(08)72023-8.

11 Thomas Colcombet. Unambiguity in automata theory. In Jeffrey Shallit and Alexander Okhotin, editors, *Descriptional Complexity of Formal Systems - 17th International Workshop, DCFS 2015, Waterloo, ON, Canada, June 25-27, 2015. Proceedings*, volume 9118 of *Lecture Notes in Computer Science*, pages 3–18. Springer, 2015. doi:10.1007/978-3-319-19225-3_1.

12 Louis Comtet. Calcul pratique des coefficients de Taylor d'une fonction algébrique. *Enseignement Math. (2)*, 10:267–270, 1964.

13 Samuel Eilenberg and Marcel-Paul Schützenberger. Rational sets in commutative monoids. *J. Algebra*, 13(2):173–191, 1969. doi:10.1016/0021-8693(69)90070-2.

14 Emmanuel Filiot, Shibashis Guha, and Nicolas Mazzocchi. Two-Way Parikh Automata. In Arkadev Chattopadhyay and Paul Gastin, editors, *39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019)*, volume 150 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2019.40.

15 Philippe Flajolet. Analytic models and ambiguity of context-free languages. *Theor. Comput. Sci.*, 49(2):283–309, 1987. doi:10.1016/0304-3975(87)90011-9.

16 Philippe Flajolet, Stefan Gerhold, and Bruno Salvy. On the non-holonomic character of logarithms, powers, and the nth prime function. *Electr. J. Comb.*, 11(2), 2005. doi:10.37236/1894.

17 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, first edition, 2009.

18 Sheila Greibach. A note on undecidable properties of formal languages. *Mathematical Systems Theory*, 2(1):1–6, 1968. doi:10.1007/BF01691341.

19 Oscar H. Ibarra. Reversal-bounded multicounter machines and their decision problems. *J. ACM*, 25(1):116–133, 1978. doi:10.1145/322047.322058.

20 Ryuichi Ito. Every semilinear set is a finite union of disjoint linear sets. *J. Comput. Syst. Sci.*, 3(2):221–231, 1969. doi:10.1016/S0022-0000(69)80014-0.

21 Masaki Kashiwara. On the holonomic systems of linear differential equations. II. *Invent. Math.*, 49(2):121–135, 1978. doi:10.1007/BF01403082.

22 Manuel Kauers. Bounds for D-finite closure properties. In *Proceedings of the 39th International Symposium on Symbolic and Algebraic Computation, (ISSAC 2014)*, pages 288–295. ACM, New York, 2014. doi:10.1145/2608628.2608634.

23 Felix Klaedtke and Harald Rueß. Parikh automata and Monadic Second-Order logics with linear cardinality constraints. Technical Report 177, Freiburg University, 2002.

24 Felix Klaedtke and Harald Rueß. Monadic second-order logics with cardinalities. In *Automata, Languages and Programming, 30th International Colloquium, ICALP 2003*, volume 2719 of *Lecture Notes in Computer Science*, pages 681–696. Springer, 2003. doi:10.1007/3-540-45061-0_54.

25 Leonard Lipshitz. The diagonal of a D-finite power series is D-finite. *J. Algebra*, 113(2):373–378, 1988. doi:10.1016/0021-8693(88)90166-4.

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

27 Paolo Massazza. Holonomic functions and their relation to linearly constrained languages. *ITA*, 27(2):149–161, 1993. doi:10.1051/ita/1993270201491.

28 Paolo Massazza. On the conjecture $\mathcal{L}_{\text{dfcm}} \subsetneq$ RCM. In *Implementation and Application of Automata - 22nd International Conference, CIAA 2017*, volume 10329 of *Lecture Notes in Computer Science*, pages 175–187. Springer, 2017. doi:10.1007/978-3-319-60134-2_15.

29 Paolo Massazza. On the generating functions of languages accepted by deterministic one-reversal counter machines. In *Proceedings of the 19th Italian Conference on Theoretical Computer Science, (ICTCS 2018)*, volume 2243 of *CEUR Workshop Proceedings*, pages 191–202. CEUR-WS.org, 2018.

30 Rohit J. Parikh. On context-free languages. *J. ACM*, 13(4):570–581, 1966. doi:10.1145/321356.321364.

31 Mojżesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. *C.R. 1er Congrès des Mathématiciens des pays slaves*, pages 92–101, 1929.

32 Frank P. Ramsey. On a Problem of Formal Logic. *Proc. London Math. Soc. (2)*, 30(4):264–286, 1929. doi:10.1112/plms/s2-30.1.264.

33 Richard P. Stanley. Differentiably finite power series. *Eur. J. Comb.*, 1(2):175–188, 1980. doi:10.1016/S0195-6698(80)80051-5.

34 Richard P. Stanley. *Enumerative combinatorics. Vol. 2*, volume 62 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 1999. doi:10.1017/CBO9780511609589.

35 Richard Edwin Stearns and Harry B. Hunt III. On the equivalence and containment problems for unambiguous regular expressions, regular grammars and finite automata. *SIAM J. Comput.*, 14(3):598–611, 1985. doi:10.1137/0214044.

36 Nobuki Takayama. An approach to the zero recognition problem by Buchberger algorithm. *J. Symbolic Comput.*, 14(2-3):265–282, 1992. doi:10.1016/0747-7171(92)90039-7.