Bidimensional Linear Recursive Sequences and Universality of Unambiguous Register Automata

Corentin Barloy
École Normale Supérieure de Paris, PSL, France

Lorenzo Clemente
University of Warsaw, Poland

Abstract

We study the universality and inclusion problems for register automata over equality data $(\mathbb{A}, =)$. We show that the universality $L(B) = (\Sigma \times \mathbb{A})^*$ and inclusion problems $L(A) \subseteq L(B)$ can be solved with 2-EXPTIME complexity when both automata are without guessing and B is unambiguous, improving on the currently best-known 2-EXPSPACE upper bound by Mottet and Quaas. When the number of registers of both automata is fixed, we obtain a lower EXPTIME complexity, also improving the EXPSPACE upper bound from Mottet and Quaas for fixed number of registers. We reduce inclusion to universality, and then we reduce universality to the problem of counting the number of orbits of runs of the automaton. We show that the orbit-counting function satisfies a system of bidimensional linear recursive equations with polynomial coefficients (linrec), which generalises analogous recurrences for the Stirling numbers of the second kind, and then we show that universality reduces to the zeroness problem for linrec sequences. While such a counting approach is classical and has successfully been applied to unambiguous finite automata and grammars over finite alphabets, its application to register automata over infinite alphabets is novel.

We provide two algorithms to decide the zeroness problem for bidimensional linear recursive sequences arising from orbit-counting functions. Both algorithms rely on techniques from linear non-commutative algebra. The first algorithm performs variable elimination and has elementary complexity. The second algorithm is a refined version of the first one and it relies on the computation of the Hermite normal form of matrices over a skew polynomial field. The second algorithm yields an EXPTIME decision procedure for the zeroness problem of linrec sequences, which in turn yields the claimed bounds for the universality and inclusion problems of register automata.

2012 ACM Subject Classification Theory of computation → Automata over infinite objects

Keywords and phrases unambiguous register automata, universality and inclusion problems, multi-dimensional linear recurrence sequences

Digital Object Identifier 10.4230/LIPIcs.STACS.2021.8

Related Version *Full Version*: `https://arxiv.org/abs/2101.01033` [2]

Funding *Corentin Barloy*: Partially supported by the Polish NCN grant 2017/26/D/ST6/00201.
Lorenzo Clemente: Partially supported by the Polish NCN grant 2017/26/D/ST6/00201.

Acknowledgements We would like to thank Daniel Robertz for kindly providing us with the `LDA` package for Maple 16.

1 Introduction

Register automata. *Register automata* extend finite automata with finitely many registers holding values from an infinite *data domain* $\mathbb{A}$ which can be compared against the data appearing in the input. The study of register automata arises naturally in automata theory as a conservative generalisation of finite automata over finite alphabets Σ to richer but well-behaved classes of infinite alphabets. The seminal work of Kaminski and Francez introduced *finite-memory automata* as the study of register automata over the data domain $(\mathbb{A}, =)$

38th International Symposium on Theoretical Aspects of Computer Science (STACS 2021).
Editors: Markus Bläser and Benjamin Monmege; Article No. 8; pp. 8:1–8:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

consisting of an infinite set $\mathbb{A}$ and the equality relation [21]. The recent book [3] studies automata theory over other data domains such as $(\mathbb{Q}, \leq)$, and more generally homogeneous [24] or even ω-categorical relational structures. Another motivation for the study of register automata comes from the area of database theory: XML documents can naturally be modelled as finite unranked trees where data values from an infinite alphabet are necessary to model the *attribute values* of the document (cf. [27] and the survey [33]).

The central verification question for register automata is the *inclusion problem*, which, for two given automata A, B, asks whether $L(A) \subseteq L(B)$. In full generality the problem is undecidable and this holds already in the special case of the *universality problem* $L(B) = (\Sigma \times \mathbb{A})^*$ [27, Theorem 5.1], when B has only two registers [3, Theorem 1.8] (or even just one register in the more powerful model *with guessing* [3, Exercise 9], i.e., non-deterministic reassignment in the terminology of [22]). One way to obtain decidability is to restrict the automaton B. One such restriction requires that B is *deterministic*: Since deterministic register automata are effectively closed under complementation, the inclusion problem reduces to non-emptiness of $L(A) \cap (\Sigma \times \mathbb{A})^* \setminus L(B)$, which can be checked in PSPACE. Another, incomparable, restriction demands that B has only one register: In this case the problem becomes decidable [21, Appendix A][1] and non-primitive recursive [18, Theorem 5.2].

Unambiguity. *Unambiguous automata* are a natural class of automata intermediate between deterministic and nondeterministic automata. An automaton is unambiguous if there is at most one accepting run on every input word. Unambiguity has often been used to generalise decidability results for deterministic automata at the price of a usually modest additional complexity. For instance, the universality problem for deterministic finite automata (which is PSPACE-complete in general [38]) is NL-complete, while for the unambiguous variant it is in PTIME [37, Corollary 4.7], and even in NC^2 [39]. An even more dramatic example is provided by universality of context-free grammars, which is undecidable in general [20, Theorem 9.22], PTIME-complete for deterministic context-free grammars, and decidable for unambiguous context-free grammars [31, Theorem 5.5] (even in PSPACE [12, Theorem 10]). (The more general equivalence problem is decidable for deterministic context-free grammars [34], but it is currently an open problem whether equivalence is decidable for unambiguous context-free grammars, as well as for the more general *multiplicity equivalence* of context-free grammars [23].) Other applications of unambiguity for universality and inclusion problems in automata theory include Büchi automata [5, 1], probabilistic automata [17], Parikh automata [7, 4], vector addition systems [16], and several others (cf. also [14, 15]).

Number sequences and the counting approach. The universality problem for a language over finite words $L \subseteq \Sigma^*$ is equivalent to whether its associated *word counting function* $f_L(n) := |L \cap \Sigma^n|$ equals $|\Sigma|^n$ for every n. The most classical way of exploiting unambiguity of a computation model A (finite automaton, context-free grammar, ...) is to use the fact that it yields a bijection between the recognised language $L(A)$ and the set of accepting runs. In this way, $f_L(n)$ is also the number of accepting runs of length n, and for the latter recursive descriptions usually exist. When the class of number sequences to which f_L belongs contains $|\Sigma|^n$ and is closed under difference, this is equivalent to the *zeroness* problem for $g(n) := |\Sigma|^n - f_L(n)$, which amounts to decide whether $g = 0$. This approach has been

[1] Decidability even holds for the so-called "two-window register automata", which combined with the restriction in [21] demanding that the last data value read must always be stored in some register boils down to a slightly more general class of "$1\frac{1}{2}$-register automata".

pioneered by Chomsky and Schützenberger [11] who have shown that the generating function $g_L(x) = \sum_{n=0}^{\infty} f_L(n) \cdot x^n$ associated to an unambiguous context-free language L is algebraic (cf. [6]). A similar observation by Stearns and Hunt [37] shows that $g_L(x)$ is rational [36, Chapter 4], when L is regular, and more recently by Bostan et al. [4] who have shown that $g_L(x)$ is holonomic [35] when L is recognised by an unambiguous Parikh automaton. Since the zeroness problem for rational, algebraic, and holonomic generating functions is decidable, one obtains decidability of the corresponding universality problems.

Unambiguous register automata. Returning to register automata, Mottet and Quaas have recently shown that the inclusion problem in the case where B is an unambiguous register automaton over equality data (without guessing) can be decided in 2-EXPSPACE, and in EXPSPACE when the numbers of registers of B is fixed [25, Theorem 1]. Note that already decidability is interesting, since unambiguous register automata without guessing are not closed under complement in the class of nondeterministic register automata without guessing [22, Example 4], and thus the classical approach via complementing B fails for register automata[2]. (In fact, even for finite automata complementation of unambiguous finite automata cannot lead to a PTIME universality algorithm, thanks to Raskin's recent super-polynomial lower-bound for the complementation problem for unambiguous finite automata in the class of non-deterministic finite automata [30]). Mottet and Quaas obtain their result by showing that inclusion can be decided by checking a reachability property of a suitable graph of triply-exponential size obtained by taking the product of A and B, and then applying the standard NL algorithm for reachability in directed graphs.

Our contributions. In view of the widespread success of the counting approach to unambiguous models of computation, one may wonder whether it can be applied to register automata as well. This is the topic of our paper. A naïve counting approach for register automata immediately runs into trouble since there are infinitely many data words of length n. The natural remedy is to use the fact that $\mathbb{A}^n$, albeit infinite, is *orbit-finite* [3, Sec. 3.2], which is a crucial notion generalising finiteness to the realm of relational structures used to model data. In this way, we naturally count the number of *orbits* of words/runs of a given length, which in the context of model theory is sometimes known as the *Ryll-Nardzewski function* [32]. For example, in the case of equality data $(\mathbb{A}, =)$, the number of orbits of words of length n is the well-known *Bell number* $B(n)$, and for $(\mathbb{Q}, \leq)$ one obtains the *ordered Bell numbers* (a.k.a. *Fubini numbers*); cf. Cameron's book for more examples [9, Ch. 7].

When considering orbits of runs, the run length n seems insufficient to obtain recurrence equations. To this end, we also consider the number of distinct data values k that appear on the word labelling the run. For instance, in the case of equality data, the corresponding orbit-counting function is the well-known sequence of *Stirling numbers of the second kind* $S(n, k) : \mathbb{Q}^{\mathbb{N}^2}$, which satisfies $S(0, 0) = 1$, $S(m, 0) = S(0, m) = 0$ for $m \geq 1$, and

$$S(n, k) = S(n-1, k-1) + k \cdot S(n-1, k), \qquad \text{for } n, k \geq 1. \tag{1}$$

These intuitions lead us to define the class of *bidimensional linear recursive sequences with polynomial coefficients* (linrec; cf. (2)) which are a class of number sequences in $\mathbb{Q}^{\mathbb{N}^2}$ satisfying a system of shift equations with polynomial coefficients generalising (1). Linrec are sufficiently

[2] In the more general class of register automata with guessing, an unproved conjecture proposed by Colcombet states that unambiguous register automata with guessing are effectively closed under complement [15, Theorem 12], implying decidability of the universality and containment problems for unambiguous register automata with guessing and, a posteriori, unambiguous register automata without guessing as considered in this paper. No published proof of this conjecture has appeared as of yet.

general to model the orbit-counting functions of register automata and yet amenable to algorithmic analysis. Our first result is a complexity upper bound for the zeroness problem for a class of linrec sequences which suffices to model register automata.

▶ **Theorem 1.** *The zeroness problem for linrec sequences with univariate polynomial coefficients from $\mathbb{Q}[k]$ is in* EXPTIME*.*

This is obtained by modelling linrec equations as systems of linear equations with *skew polynomial coefficients* (introduced by Ore [29]) and then using complexity bounds on the computation of the Hermite normal form of skew polynomial matrices by Giesbrecht and Kim [19]. Our second result is a reduction of the universality and inclusion problems to the zeroness problem of a system of linrec equations of exponential size. Together with Theorem 1, this yields improved upper bounds on the former problems.

▶ **Theorem 2.** *The universality $L(B) = (\Sigma \times \mathbb{A})^*$ and the inclusion problem $L(A) \subseteq L(B)$ for register automata A, B without guessing with B unambiguous are in* 2-EXPTIME*, and in* EXPTIME *for a fixed number of registers of A, B. The same holds for the equivalence problem $L(A) = L(B)$ when both automata are unambiguous.*

The rest of the paper is organised as follows. In Section 2, we introduce linrec sequences (cf. [2, Appendix A.3] for a comparison with well known sequence families from the literature such as the C-recursive, P-recursive, and the more recent polyrec sequences [8]). In Section 3, we introduce unambiguous register automata and we present an efficient reduction of the inclusion (and thus equivalence) problem to the universality problem, which allows us to concentrate on the latter in the rest of the paper. In Section 4, we present a reduction of the universality problem to the zeroness problem for linrec. In Section 5, we show with a simple argument based on elimination that the zeroness problem for linrec is decidable, and in Section 6 we derive a complexity upper bound using non-commutative linear algebra. Finally, in Section 7 we conclude with further work and an intriguing conjecture. Full proofs, additional definitions, and examples are provided in the full version of the paper [2].

Notation. Let $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$ be the set of non-negative integers, resp., rationals. The *height* of an integer $k \in \mathbb{Z}$ is $|k|_\infty = |k|$, and for a rational number $a \in \mathbb{Q}$ uniquely written as $a = \frac{p}{q}$ with $p \in \mathbb{Z}, q \in \mathbb{N}$ co-prime we define $|a|_\infty = \max\{|p|_\infty, |q|_\infty\}$. Let $\mathbb{Q}[n, k]$ denote the ring of bivariate polynomials. The *(combined) degree* $\deg P$ of $P = \sum_{i,j} a_{ij} n^i k^j \in \mathbb{Q}[n, k]$ is the maximum $i + j$ s.t. $a_{ij} \neq 0$ and the *height* $|P|_\infty$ is $\max_{i,j} |a_{ij}|_\infty$. For a nonempty set A and $n \in \mathbb{N}$, let A^n be the set of sequences of elements from A of length n, In particular, $A^0 = \{\varepsilon\}$ contains only the empty sequence ε. Let $A^* = \bigcup_{n \in \mathbb{N}} A^n$ be the set of all finite sequences over A. We use the *soft-Oh* notation $\tilde{O}(f(n))$ to denote $\bigcup_{c \geq 0} O(f(n) \cdot \log^c f(n))$.

2 Bidimensional linear recursive sequences with polynomial coefficients

Let $f(n, k) : \mathbb{Q}^{\mathbb{N}^2}$ be a bidimensional sequence. For $L \in \mathbb{N}$, the *first L-section* of f is the one-dimensional sequence $f(L, k) : \mathbb{Q}^{\mathbb{N}}$ obtained by fixing its first component to L; the *second L-section* $f(n, L)$ is defined similarly. The two *shift operators* $\partial_1, \partial_2 : \mathbb{Q}^{\mathbb{N}^2} \to \mathbb{Q}^{\mathbb{N}^2}$ are

$$(\partial_1 f)(n, k) = f(n + 1, k) \quad \text{and} \quad (\partial_2 f)(n, k) = f(n, k + 1), \quad \text{for all } n, k \geq 0.$$

An *affine operator* is a formal expression of the form $A = p_{00} + p_{01} \cdot \partial_1 + p_{10} \cdot \partial_2$ where $p_{00}, p_{01}, p_{10} \in \mathbb{Q}[n, k]$ are bivariate polynomials over n, k with rational coefficients. Let

$\{f_1, \ldots, f_m\}$ be a set of variables denoting bidimensional sequences[3]. A *system of linear shift equations* over $f_1, \ldots, f_m$ consists of m equations of the form

$$\begin{cases} \partial_1 \partial_2 f_1 &= A_{1,1} \cdot f_1 + \cdots + A_{1,m} \cdot f_m, \\ &\vdots \\ \partial_1 \partial_2 f_m &= A_{m,1} \cdot f_1 + \cdots + A_{m,m} \cdot f_m, \end{cases} \tag{2}$$

where the $A_{i,j}$'s are affine operators. A bidimensional sequence $f : \mathbb{Q}^{\mathbb{N}^2}$ is *linear recursive of order m, degree d, and height h* (abbreviated, linrec) if the following two conditions hold:

1) there are auxiliary bidimensional sequences $f_2, \ldots, f_m : \mathbb{Q}^{\mathbb{N}^2}$ which together with $f = f_1$ satisfy a system of linear shift equations as in (2) where the polynomial coefficients have (combined) degree $\leq d$ and height $\leq h$.

2) for every $1 \leq i \leq m$ there are constants denoted $f_i(0, \geq 1), f_i(\geq 1, 0) \in \mathbb{Q}$ s.t. $f_i(0,k) = f_i(0, \geq 1)$ and $f_i(n,0) = f_i(\geq 1, 0)$ for every $n, k \geq 1$.

If we additionally fix the initial values $f_1(0,0), \ldots, f_m(0,0)$, then the system (2) has a unique solution, which is computable in PTIME.

▶ **Lemma 3.** *The values $f_i(n,k)$'s are computable in deterministic time $\tilde{O}(m \cdot n \cdot k)$.*

In the following we will use the following effective closure under section.

▶ **Lemma 4.** *If $f : \mathbb{Q}^{\mathbb{N}^2}$ is linrec of order $\leq m$, degree $\leq d$, and height $\leq h$, then its L-sections $f(L,k), f(n,L) : \mathbb{Q}^{\mathbb{N}}$ are linrec of order $\leq m \cdot (L+3)$, degree $\leq d$, and height $\leq h \cdot L^d$.*

We are interested in the following central algorithmic problem for linrec.

Zeroness problem.
Input: A system of linrec equations (2) together with all initial conditions.
Output: Is it the case that $f_1 = 0$?

In Section 4 we use linrec sequences to model the orbit-counting functions of register automata, which we introduce next.

3 Unambiguous register automata

We consider register automata over the relational structure $(\mathbb{A}, =)$ consisting of a countable set $\mathbb{A}$ equipped with equality as the only relational symbol. Let $\bar{a} = a_1 \cdots a_n \in \mathbb{A}^n$ be a finite sequence of n data values. An *$\bar{a}$-automorphism* of $\mathbb{A}$ is a bijection $\alpha : \mathbb{A} \to \mathbb{A}$ s.t. $\alpha(a_i) = a_i$ for every $1 \leq i \leq n$, which is extended pointwise to $\bar{a} \in \mathbb{A}^n$ and to $L \subseteq \mathbb{A}^*$. For $\bar{b}, \bar{c} \in \mathbb{A}^n$, we write $\bar{b} \sim_{\bar{a}} \bar{c}$ whenever there is an $\bar{a}$-automorphism α s.t. $\alpha(\bar{b}) = \bar{c}$. The *$\bar{a}$-orbit* of $\bar{b}$ is the equivalence class $[\bar{b}]_{\bar{a}} = \{\bar{c} \in \mathbb{A}^n \mid \bar{b} \sim_{\bar{a}} \bar{c}\}$, and the set of $\bar{a}$-orbits of sequences in $L \subseteq \mathbb{A}^*$ is $\mathsf{orbits}_{\bar{a}}(L) = \{[\bar{b}]_{\bar{a}} \mid \bar{b} \in L\}$. In the special case when $\bar{a} = \varepsilon$ is the empty tuple, we just speak about *automorphism* α and *orbit* $[\bar{b}]$. A set X is *orbit-finite* if $\mathsf{orbits}(X)$ is a finite set [3, Sec. 3.2]. All definitions above extend to $\mathbb{A}_\perp := \mathbb{A} \cup \{\perp\}$ with $\perp \notin \mathbb{A}$ in the expected way. A *constraint* φ is a quantifier-free[4] formula generated by $\varphi, \psi ::\equiv x = \perp \mid x = y \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \neg \varphi$, where x, y are variables and $\perp$ is a special constant denoting an undefined value. The semantics of a constraint

[3] We abuse notation and silently identify variables denoting sequences with the sequences they denote.
[4] Since $(\mathbb{A}, =)$ is a homogeneous relational structure, and thus it admits quantifier elimination, we would obtain the same expressive power if we would consider more general first-order formulas instead.

$\varphi(x_1, \ldots, x_n)$ with n free variables $x_1, \ldots, x_n$ is the set of tuples of n elements which satisfies: $[\![\varphi]\!] = \{a_1, \ldots, a_n \in \mathbb{A}_\bot^n \mid \mathbb{A}_\bot, x_1 : a_1, \ldots, x_n : a_n \models \varphi\}$. A *register automaton* of *dimension* $d \in \mathbb{N}$ is a tuple $A = (d, \Sigma, \mathsf{L}, \mathsf{L}_I, \mathsf{L}_F, \to)$ where d is the number of registers, Σ is a finite alphabet, L is a finite set of *control locations*, of which we distinguish those which are *initial* $\mathsf{L}_I \subseteq \mathsf{L}$, resp., *final* $\mathsf{L}_F \subseteq \mathsf{L}$, and "$\to$" is a set of rules of the form $p \xrightarrow{\sigma,\varphi} q$, where $p, q \in \mathsf{L}$ are control locations, $\sigma \in \Sigma$ is an input symbol from the finite alphabet, and $\varphi(x_1, \ldots, x_d, y, x'_1, \ldots, x'_d)$ is a constraint relating the current register values x_i's, the current input symbol (represented by the variable y), and the next register values of x'_i's.

▶ **Example 5.** Let A over $|\Sigma| = 1$ have one register x, and four control locations p, q, r, s, of which p is initial and s is final. The transitions are $p \xrightarrow{x=\bot \wedge x'=y} q$, $p \xrightarrow{x=\bot \wedge x'=y} r$, $q \xrightarrow{x\neq y \wedge x'=x} q$, $q \xrightarrow{x=y \wedge x'=x} s$, $r \xrightarrow{x=y \wedge x'=x} r$, and $r \xrightarrow{x\neq y \wedge x'=x} s$. The automaton accepts all words of the form $a(\mathbb{A} \setminus \{a\})^* a$ or $aa^*(\mathbb{A} \setminus \{a\})$ with $a \in \mathbb{A}$.

A register automaton is *orbitised* if every constraint φ appearing in some transition thereof denotes an orbit $[\![\varphi]\!] \in \mathsf{orbits}(\mathbb{A}_\bot^{2\cdot d+1})$. For example, when $d = 1$ the constraint $\varphi \equiv x = x'$ is not orbitised, however $[\![\varphi]\!] = [\![\varphi_0]\!] \cup [\![\varphi_1]\!]$ splits into two disjoint orbits for the orbitised constraints $\varphi_0 \equiv x = x' \wedge x = y$ and $\varphi_1 \equiv x = x' \wedge x \neq y$. The automaton from Example 5 is orbitised. Every register automaton can be transformed in orbitised form by replacing every transition $p \xrightarrow{\sigma,\varphi} q$ with exponentially many transitions $p \xrightarrow{\sigma,\varphi_1} q, \ldots, p \xrightarrow{\sigma,\varphi_n} q$, for each orbit $[\![\varphi_i]\!]$ of $[\![\varphi]\!] \subseteq \mathbb{A}_\bot^{2\cdot d+1}$.

A *register valuation* is a tuple of (possibly undefined) values $\bar{a} = (a_1, \ldots, a_d) \in \mathbb{A}_\bot^d$. A *configuration* is a pair $(p, \bar{a})$, where $p \in \mathsf{L}$ is a control location and $\bar{a} \in \mathbb{A}_\bot^d$ is a register valuation; it is *initial* if $p \in \mathsf{L}_I$ is initial and all registers are initially undefined $\bar{a} = (\bot, \ldots, \bot)$, and it is *final* whenever $p \in \mathsf{L}_F$ is so. The *semantics* of a register automaton A is the infinite transition system $[\![A]\!] = (C, C_I, C_F, \to)$ where C is the set of configurations, of which $C_I, C_F \subseteq C$ are the initial, resp., final ones, and $\to \subseteq C \times (\Sigma \times \mathbb{A}) \times C$ is the set of all transitions of the form

$$(p, \bar{a}) \xrightarrow{\sigma,a} (q, \bar{a}'), \qquad \text{with } \sigma \in \Sigma, a \in \mathbb{A}, \text{ and } \bar{a}, \bar{a}' \in \mathbb{A}_\bot^d,$$

s.t. there exists a rule $p \xrightarrow{\sigma,\varphi} q$ where satisfying the constraint $\mathbb{A}_\bot, \bar{x} : \bar{a}, y : a, \bar{x}' : \bar{a}' \models \varphi$. A *data word* is a sequence $w = (\sigma_1, a_1) \cdots (\sigma_n, a_n) \in (\Sigma \times \mathbb{A})^*$. A *run over* a data word w *starting at* $c_0 \in C$ and *ending at* $c_n \in C$ is a sequence π of transitions of $[\![A]\!]$ of the form $\pi = c_0 \xrightarrow{\sigma_1,a_1} c_1 \xrightarrow{\sigma_2,a_2} \cdots \xrightarrow{\sigma_n,a_n} c_n$. We denote with $\mathsf{Runs}(c_0; w; c_n)$ the set of runs over w starting at c_0 and ending in c_n, and with $\mathsf{Runs}(C_I; w; c_n)$ the set of *initial runs*, i.e., those runs over w starting at some initial configuration $c_0 \in C_I$ and ending in c_n. The run π is *accepting* if $c_n \in C_F$. The language $L(A, c)$ recognised from configuration $c \in C$ is the set of data words labelling some accepting run starting at c; the language recognised from a set of configurations $D \subseteq C$ is $L(A, D) = \bigcup_{c \in D} L(A, c)$, and the language recognised by the register automaton A is $L(A) = L(A, C_I)$. Similarly, the *backward language* $L^{\mathsf{R}}(A, c)$ is the set of words labelling some run starting at an initial configuration and ending at c. Thus, we also have $L(A) = L^{\mathsf{R}}(A, C_F)$. A register automaton is *deterministic* if for every input word there exists at most one initial run, and *unambiguous* if for every input word there is at most one initial and accepting run. A register automaton is *without guessing* if, for every initial run $(p, \bot^d) \xrightarrow{w} (q, \bar{a})$ every non-$\bot$ data value in $\bar{a}$ occurs in the input w, written $\bar{a} \subseteq w$. In the rest of the paper we will study exclusively automata without guessing. A deterministic automaton is unambiguous and without guessing. These semantic properties can be decided in PSPACE with simple reachability analyses (cf. [15]).

▶ **Example 6.** The automaton from Example 5 is unambiguous and without guessing. An example of language which can only be recognised by ambiguous register automata is the set of words where the same data value appears two times $L = \{u \cdot a \cdot v \cdot a \cdot w \mid a \in \mathbb{A}; u, v, w \in \mathbb{A}^*\}$.

▶ **Lemma 7.** *If A is an unambiguous register automaton, then there is a bijection between the language it recognises $L(A) = L(A, C_I) = L^{\mathsf{R}}(A, C_F)$ and the set of runs starting at some initial configuration in C_I and ending at some final configuration in C_F.*

We are interested in the following decision problem.

Inclusion problem.
Input: Two register automata A, B over the same input alphabet Σ.
Output: Is it the case that $L(A) \subseteq L(B)$?

The *universality problem* asks $L(A) = (\Sigma \times \mathbb{A})^*$, and the *equivalence problem* $L(A) = L(B)$. In general, universality reduces to equivalence, which in turn reduces to inclusion. In our context, inclusion reduces to universality and thus all three problems are equivalent.

▶ **Lemma 8.** *Let A and B be two register automata.*
1. *The inclusion problem $L(A) \subseteq L(B)$ with A orbitised and without guessing reduces in* PTIME *to the case where A is deterministic. The reduction preserves whether B is 1) unambiguous, 2) without guessing, and 3) orbitised.*
2. *The inclusion problem $L(A) \subseteq L(B)$ with A deterministic reduces in* PTIME *to the universality problem for some register automaton C. If B is unambiguous, then so is C. If B is without guessing, then so is C. If A and B are orbitised, then so is C.*

4 Universality of unambiguous register automata without guessing

We reduce universality of unambiguous register automata without guessing to zeroness of bidimensional linrec sequences with univariate polynomial coefficients. The *width* of a sequence of data values $\bar a = a_1 \cdots a_n \in \mathbb{A}^n$ is $\#\bar a = |\{a_1, \ldots, a_n\}|$, for a word $w = (\sigma_1, a_1) \cdots (\sigma_n, a_n) \in (\Sigma \times \mathbb{A})^*$ we set $\#w = \#(a_1 \cdots a_n)$, and for a run π over w we set $\#\pi = \#w$. Let the *Ryll-Nardzewski function* $G_{p,\bar a}(n, k)$ of a configuration $(p, \bar a) \in C = \mathsf{L} \times \mathbb{A}_\perp^d$ count the number of $\bar a$-orbits of initial runs of length n and width k ending in $(p, \bar a)$:

$$G_{p,\bar a}(n, k) = |\{[\pi]_{\bar a} \mid w \in (\Sigma \times \mathbb{A})^n, \pi \in \mathsf{Runs}(C_I; w; p, \bar a), \#w = k\}| . \tag{3}$$

▶ **Lemma 9.** *Let $\bar a, \bar b \in \mathbb{A}_\perp^d$. If $[\bar a] = [\bar b]$, then $G_{p,\bar a}(n, k) = G_{p,\bar b}(n, k)$ for every $n, k \geq 0$.*

We thus overload the notation and write $G_{p,[\bar a]}$ instead of $G_{p,\bar a}$. Since $\mathbb{A}_\perp^d$ is orbit-finite, this yields finitely many variables $G_{p,[\bar a]}$'s. By slightly abusing notation, let $G_{C_F}(n, k) = \sum_{[(p,\bar a)] \in \mathsf{orbits}(C_F)} G_{p,[\bar a]}(n, k)$ be the sum of the Ryll-Nardzewski function over all orbits of accepting configurations. When the automaton is unambiguous, thanks to Lemma 7, $G_{C_F}(n, k)$ is also the number of orbits of accepted words of length n and width k.

▶ **Lemma 10.** *Let A be an unambiguous register automaton w/o guessing over Σ and let $S_\Sigma(n, k)$ be the number of orbits of all words of length n and width k. We have $L(A) = (\mathbb{A} \times A)^*$ if, and only if, $\forall n, k \in \mathbb{N} \cdot G_{C_F}(n, k) = S_\Sigma(n, k)$.*

In other words, universality of A reduces to zeroness of $G := S_\Sigma - G_{C_F}$. The sequence S_Σ is linrec since it satisfies the recurrence in Figure 2 with initial conditions $S_\Sigma(0, 0) = 1$ and $S_\Sigma(n + 1, 0) = S_\Sigma(0, k + 1) = 0$ for $n, k \geq 0$. We show that all the sequences of the form

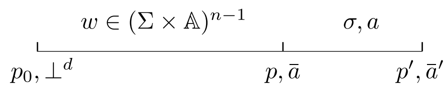

Figure 1 Last-step decomposition.

$$
\begin{aligned}
G_{p',[\bar{a}']}(n+1,k+1) &= \sum_{[p,\bar{a} \xrightarrow{\sigma,a} p',\bar{a}'] \colon a \in \bar{a}} \underbrace{G_{p,[\bar{a}]}(n,k+1)}_{\mathbf{I}} + \\
&\quad \sum_{[p,\bar{a} \xrightarrow{\sigma,a} p',\bar{a}'] \colon a \notin \bar{a}} \left(\underbrace{G_{p,[\bar{a}]}(n,k)}_{\mathbf{II}} + \underbrace{\max(k+1-\#[\bar{a}],0) \cdot G_{p,[\bar{a}]}(n,k+1)}_{\mathbf{III}} \right), \\
S_\Sigma(n+1,k+1) &= |\Sigma| \cdot S_\Sigma(n,k) + |\Sigma| \cdot (k+1) \cdot S_\Sigma(n,k+1), \\
G(n,k) &= S_\Sigma(n,k) - \sum_{[p,\bar{a}] \in \mathsf{orbits}(C_F)} G_{p,[\bar{a}]}(n,k).
\end{aligned}
$$

Figure 2 Linrec automata equations.

$G_{p,[\bar{a}]}$ are also linrec and thus also G will be linrec. We perform a last-step decomposition of an initial run; cf. Figure 1. Starting from some initial configuration $(p_0, \bot^d)$, the automaton has read a word w of length $n-1$ leading to $(p, \bar{a})$. Then, the automaton reads the last letter (σ, a) and goes to $(p', \bar{a}')$ via the transition $t = (p, \bar{a} \xrightarrow{\sigma,a} p', \bar{a}')$. The question is in how many distinct ways can an orbit of the run over w be extended into an orbit of the run over $w \cdot (\sigma, a)$. We distinguish three cases.

- **I:** Assume that a appears in register $\bar{a}_i = a$. Since the automaton is without guessing, $a \in w$ has appeared earlier in the input word and $\bar{a}' \subseteq \bar{a}$ (ignoring $\bot$'s). Thus, each $\bar{a}$-orbit of runs $[p_0, \bot^d \xrightarrow{w} p, \bar{a}]_{\bar{a}}$ yields, via the fixed t, an $\bar{a}'$-orbit of runs $[p_0, \bot^d \xrightarrow{w} p, \bar{a} \xrightarrow{\sigma,a} p', \bar{a}']_{\bar{a}'}$ of the same width in just one way.
- **II:** Assume that a is globally fresh $a \notin w$, and thus in particular $a \notin \bar{a}$ since the automaton is without guessing. Each $\bar{a}$-orbit of runs $[p_0, \bot^d \xrightarrow{w} p, \bar{a}]_{\bar{a}}$ of width $\#w$ yields, via the fixed t, a single $\bar{a}'$-orbit of runs $[p_0, \bot^d \xrightarrow{w} p, \bar{a} \xrightarrow{\sigma,a} p', \bar{a}']_{\bar{a}'}$ of width $\#(w \cdot a) = \#w + 1$.
- **III:** Assume that $a \in w$ is not globally fresh, but it does not appear in any register $a \notin \bar{a}$. Since the automaton is without guessing, every value in $\bar{a}$ appears in w. Consequently, a can be any of the $\#w$ distinct values in w, with the exception of $\#\bar{a}$ values. Each $\bar{a}$-orbit of runs $[p_0, \bot \xrightarrow{w} p, \bar{a}]_{\bar{a}}$ of width $\#w$ yields $\#w - \#\bar{a} \geq 0$ $\bar{a}'$-orbits of runs $[p_0, \bot^d \xrightarrow{w} p, \bar{a} \xrightarrow{\sigma,a} p', \bar{a}']_{\bar{a}'}$ of the same width.

(As expected, we do not need unambiguity at this point, since we are counting orbits of runs.) We obtain the equations in Figure 2, where the sums range over orbits of transitions. This set of equations is finite since there are finitely many orbits $[\bar{a}] \in \mathsf{orbits}(\mathbb{A}^d_\bot)$ of register valuations, and moreover we can effectively represent each orbit by a constraint [3, Ch. 4]. Strictly speaking, the equations are not linrec due to the "max" operator, however they can easily be transformed to linrec by considering $G_{p,[\bar{a}]}(n, K)$ separately for $1 \leq K < d$; in the interest of clarity, we omit the full linrec expansion. The initial condition is $G_{p,[\bar{a}]}(0,0) = 1$ if $p \in I$ initial, and $G_{p,[\bar{a}]}(0,0) = 0$ otherwise. The two 0-sections satisfy $G_{p,[\bar{a}]}(n+1,0) = 0$ for $n \geq 0$ (if the word is nonempty, then there is at least one data value) and $G_{p,[\bar{a}]}(0,k+1) = 0$ for $k \geq 0$ (an empty word does not have any data value).

▶ **Lemma 11.** *The sequences $G_{p,[\bar{a}]}$'s satisfy the system of equations in Figure 2.*

▶ **Example 12.** The equations corresponding to the automaton in Example 5 are as follows. (Since the automaton is orbitised, we can omit the orbit.) We have $G_p(0,0) = 1$, $G_q(0,0) = G_r(0,0) = G_s(0,0) = 0$ and for $n, k \geq 0$:

$$
\begin{aligned}
&G_p(n+1,k+1) = 0,\\
&G_q(n+1,k+1) = \underbrace{G_p(n,k)}_{\mathbf{II}} + \underbrace{(k+1)\cdot G_p(n,k+1)}_{\mathbf{III}} + \underbrace{G_q(n,k)}_{\mathbf{II}} + \underbrace{k\cdot G_q(n,k+1)}_{\mathbf{III}},\\
&G_r(n+1,k+1) = \underbrace{G_p(n,k)}_{\mathbf{II}} + \underbrace{(k+1)\cdot G_p(n,k+1)}_{\mathbf{III}} + \underbrace{G_r(n,k+1)}_{\mathbf{I}},\\
&G_s(n+1,k+1) = \underbrace{G_q(n,k+1)}_{\mathbf{I}} + \underbrace{G_r(n,k)}_{\mathbf{II}} + \underbrace{k\cdot G_r(n,k+1)}_{\mathbf{III}}.
\end{aligned}
$$

▶ **Lemma 13.** *Let A be an unambiguous register automaton over equality atoms without guessing with d registers and ℓ control locations. The universality problem for A reduces to the zeroness problem of the linrec sequence G defined by the system of equations in Figure 2 containing $O(\ell \cdot 2^{d\cdot \log d})$ variables and equations and constructible in* PSPACE*. If A is already orbitised, then the system of equations has size $O(\ell)$.*

5 Decidability of the zeroness problem

In this section, we present an algorithm to solve the zeroness problem of bidimensional linrec sequences with univariate polynomial coefficients, which is sufficient for linrec sequences from Figure 2. We first give a general presentation on elimination for bivariate polynomial coefficients, and then we use the univariate assumption to obtain a decision procedure. We model the non-commutative operators appearing in the definition of linrec sequences (2) with Ore polynomials (a.k.a. skew polynomials) [29][5]. Let R be a (not necessarily commutative) ring and σ an automorphism of R. The ring of *(shift) skew polynomials* $R[\partial;\sigma]$ is defined as the ring of polynomials but where the multiplication operation satisfies the following commutation rule: For a coefficient $a \in R$ and the unknown ∂, we have

$$\partial \cdot a = \sigma(a) \cdot \partial.$$

(The usual ring of polynomials is recovered when σ is the identity.) The multiplication extends to monomials as $a\partial^k \cdot b\partial^l = a\sigma^k(b)\cdot\partial^{k+l}$ and to the whole ring by distributivity. The *degree* of a skew monomial $a \cdot \partial^k$ is k, and the degree $\deg P$ of a skew polynomial P is the maximum of the degrees of its monomials. The degree function satisfies the expected identities $\deg(P \cdot Q) = \deg P + \deg Q$ and $\deg(P + Q) \leq \max(\deg P, \deg Q)$. A skew polynomial is *monic* if the coefficient of its monomial of highest degree is 1. The crucial and only property that we need in this section is that skew polynomial rings admit a Euclidean pseudo-division algorithm, which in turns allows one to find common left multiples. A skew polynomial ring $R[\partial;\sigma]$ has *pseudo-division* if for any two skew polynomials $A, B \in R[\partial;\sigma]$ with $\deg A \geq \deg B$ there is a coefficient $a \in R$ and skew polynomials $Q, R \in R[\partial;\sigma]$ s.t. $a \cdot A = P \cdot B + Q$ and $\deg Q < \deg B$. We say that a ring R has the *common left multiple* (CLM) property if for every $a, b \neq 0$, there exists $c, d \neq 0$ such that $c \cdot a = d \cdot b$.

[5] The general definition of the Ore polynomial ring $R[\partial;\sigma,\delta]$ uses an additional component $\delta : R \to R$ in order to model differential operators. We present a simplified version which is enough for our purposes.

▶ **Theorem 14** (cf. [28, Sec. 1]). *If R has the* CLM *property, then*

1) *$R[\partial;\sigma]$ has a pseudo-division, and*

2) *$R[\partial;\sigma]$ also has the* CLM *property.*

The most important instances of skew polynomials are the *first* and *second Weyl algebras*:

$$W_1 = \mathbb{Q}[n,k][\partial_1;\sigma_1] \quad \text{and} \quad W_2 = W_1[\partial_2;\sigma_2] = \mathbb{Q}[n,k][\partial_1;\sigma_1][\partial_2;\sigma_2], \tag{4}$$

where $\mathbb{Q}[n,k]$ is the ring of bivariate polynomials, and the shifts satisfy $\sigma_1(p(n,k)) := p(n+1,k)$ and $\sigma_2\left(\sum_i p_i(n,k)\partial_1^i\right) := \sum_i p_i(n,k+1)\partial_1^i$. Skew polynomials in W_2 act on bidimensional sequences $f : \mathbb{Q}^{\mathbb{N}^2}$ by interpreting ∂_1 and ∂_2 as the two shifts. A linrec system of equations (2) can thus be interpreted as a system of linear equations with variables $f_1,\ldots,f_m$ and coefficients in W_2.

▶ **Example 15.** Continuing our running Example 12, we obtain the following linear system of equations with W_2 coefficients:

$$\begin{array}{rrrrl}
\partial_1\partial_2\cdot G_p & & & & = 0,\\
-(1+(k+1)\partial_2)\cdot G_p & +(\partial_1\partial_2 - k\partial_2 - 1)\cdot G_q & & & = 0,\\
-(1+(k+1)\partial_2)\cdot G_p & & +(\partial_1\partial_2-\partial_2)\cdot G_r & & = 0,\\
 & -\partial_2\cdot G_q & -(1+k\partial_2)\cdot G_r & +\partial_1\partial_2\cdot G_s & = 0,
\end{array}$$

$$(\partial_1\partial_2 - (k+1)\partial_2 - 1)\cdot S_1 = 0,$$
$$G_s - S_1 + G = 0.$$

Since $W_0 = \mathbb{N}[n,k]$ is commutative, it obviously has the CLM property. By two applications of Theorem 14, we have (see [2, Appendix D.1] for CLM examples):

▶ **Corollary 16.** *The two Weyl algebras W_1 and W_2 have the* CLM *property.*

A (linear) *cancelling relation* (CR) for a bidimensional sequence $f : \mathbb{Q}^{\mathbb{N}^2}$ is a linear equation of the form

$$p_{i^*,j^*}(n,k)\cdot\partial_1^{i^*}\partial_2^{j^*} f = \sum_{(i,j)<_{\text{lex}}(i^*,j^*)} p_{i,j}(n,k)\cdot\partial_1^i\partial_2^j f, \tag{CR-2}$$

where $p_{i^*,j^*}(n,k), p_{i,j}(n,k) \in \mathbb{Q}[n,k]$ are bivariate polynomial coefficients and $<_{\text{lex}}$ is the lexicographic ordering. Cancelling relations for a one-dimensional sequence $g : \mathbb{Q}^{\mathbb{N}}$ are defined analogously (we use the second variable k as the index for convenience):

$$q_{j^*}(k)\cdot\partial_2^{j^*} g = \sum_{0\le j<j^*} q_j(k)\cdot\partial_2^j g. \tag{CR-1}$$

We use cancelling relations as certificates of zeroness for f when the $p_{i,j}$'s are univariate. We do not need to construct any cancelling relation, just knowing that some exists with the required bounds suffices.

▶ **Lemma 17.** *The zeroness problem for a bidimensional linrec sequence $f : \mathbb{Q}^{\mathbb{N}^2}$ of order $\le m$ and univariate polynomial coefficients in $\mathbb{Q}[k]$ admitting some cancelling relation* (CR-2) *with leading coefficient $p_{i^*,j^*}(k) \in \mathbb{Q}[k]$ of degree $\le e$ and height $\le h$ s.t. each of the one-dimensional sections $f(M,k) \in \mathbb{Q}^{\mathbb{N}}$ for $1 \le M \le i^*$ also admits some cancelling relation* (CR-1) *of ∂_2-degree $\le d$ with leading polynomial coefficients of degrees $\le e$ and height $\le h$ is decidable in deterministic time $\tilde{O}(p(m,i^*,j^*,d,e,h))$ for some polynomial p.*

Elimination already yields decidability with elementary complexity for the zeroness problem and thus for the universality/equivalence/inclusion problems of unambiguous register automata without guessing.

▶ **Theorem 18.** *The zeroness problem for linrec sequences with univariate polynomial coefficients from $\mathbb{Q}[k]$ (or from $\mathbb{Q}[n]$) is decidable.*

▶ **Example 19.** Continuing our running Example 15, we subsequently eliminate G_p, G_s, G_r, G_q, S finally obtaining (cf. [2, Example 34 in Appendix D.2] for details)

$$\begin{aligned} G(n+4, k+4) = \quad & (k+3) \cdot G(n+3, k+4) + G(n+3, k+3) + \\ & -(k+2) \cdot G(n+2, k+4) - G(n+2, k+3). \end{aligned} \tag{5}$$

As expected, all coefficients are polynomials in $\mathbb{Q}[k]$ and in particular they do not involve the variable n. Moreover, we note that the relation above is *monic*, in the sense that the lexicographically leading term $G(n+4, k+4)$ has coefficient 1 (cf. Section 7). (Cf. [2, Example 35] for elimination in a two-register automaton and [2, Example 36] for a one-register automaton accepting all words of length ≥ 2.)

We omit a precise complexity analysis of elimination because better bounds can be obtained by resorting to linear non-commutative algebra, which is the topic of the next section.

6 Complexity of the zeroness problem

In this section we present an EXPTIME algorithm to solve the zeroness problem and we apply this result to register automata. We compute the *Hermite normal form* (HNF) of the matrix with skew polynomial coefficients associated to (2) in order to do elimination in a more efficient way. The complexity bounds provided by Giesbrecht and Kim [19] on the computation of the HNF lead to the following bounds for cancelling relations; cf. [2, Appendix E] for further details and full proofs.

▶ **Lemma 20.** *A linrec sequence $f \in \mathbb{Q}^{\mathbb{N}^2}$ of order $\leq m$, degree $\leq d$, and height $\leq h$ admits a cancelling relation* (CR-2) *with the orders i^*, j^* and the degree of p_{i^*,j^*} polynomially bounded, and with height $|p_{i^*,j^*}|_\infty$ exponentially bounded. Similarly, its one-dimensional sections $f(0,k), \ldots, f(i^*,k) \in \mathbb{Q}^{\mathbb{N}}$ also admit cancelling relations* (CR-1) *of polynomially bounded orders and degree, and exponentially bounded height.*

This allows us to prove below the EXPTIME upper-bound for zeroness of Theorem 1, and the 2-EXPTIME algorithm for inclusion of Theorem 2.

Proof of Theorem 1. Thanks to the bounds from Lemma 20, i^*, j^* are polynomially bounded; we can find a polynomial bound d on the ∂_2-degrees of the cancelling relations $R_0, \ldots, R_{i^*}$ for the sections $f(0,k), \ldots, f(i^*,k)$, respectively; we can find a polynomial bound e on the degrees of $p_{i^*,j^*}(k)$ and the leading polynomial coefficients of the R_i's; and an exponential bound h on $|p_{i^*,j^*}|_\infty$ and the heights of the leading polynomial coefficients of the R_i's. We thus obtain an EXPTIME algorithm by Lemma 17. ◀

This yields the announced upper-bounds for the inclusion problem for register automata.

Proof of Theorem 2. For the universality problem $L(B) = (\Sigma \times \mathbb{A})^*$, let d be the number of registers and ℓ the number of control locations of B. By Lemma 13, the universality problem reduces in PSPACE to zeroness of a linrec system with polynomial coefficients in $\mathbb{Q}[k]$

containing $O(\ell \cdot 2^{d \cdot \log d})$ variables $G_{p,[\bar{a}]}$ and the same number of equations. By Theorem 1, we get a 2-EXPTIME algorithm. When the numbers of registers d is fixed, we get an EXPTIME algorithm. For the inclusion problem $L(A) \subseteq L(B)$, we first orbitise A into an equivalent orbitised register automaton without guessing A'. A close inspection of the two constructions leading to C in the proof of Lemma 8 reveal that transitions in C are either transitions from A' (and thus already orbitised), or pairs of a transition in B together with a transition in A', the second of which is already orbitised. It follows that orbitising C incurs in an exponential blow-up w.r.t. the number of registers of B, but only polynomial w.r.t. the number of registers of A' (and thus of A), since the A'-part in C is already orbitised. Consequently, we can write (in PSPACE) a system of linrec equations for the universality problem of C of size exponential in the number of registers of A and of B. By reasoning as in the first part of the proof, we obtain a EXPTIME algorithm for the universality problem of C, and thus a 2-EXPTIME algorithm for the original inclusion problem $L(A) \subseteq L(B)$. If both the number of registers of A and of B is fixed, we get an EXPTIME algorithm. The equivalence problem $L(A) = L(B)$ with both automata A, B unambiguous reduces to two inclusion problems. ◀

7 Further remarks and conclusions

We say that $P = \sum_{i,j} p_{i,j}(n,k) \cdot \partial_1^i \partial_2^j$ is *monic* if $p_{i^*,j^*} = 1$ where (i^*, j^*) is the lexicographically largest pair (i, j) s.t. $p_{i,j} \neq 0$. The cancelling relation (CR-2) in our examples (5) and [2, (10), (11), (15)] happens to be monic in this sense.

▶ **Conjecture 21** (Monicity conjecture). *There always exists a* monic *cancelling relation* (CR-2) *for linrec systems obtained from automata equations in Figure 2, and similarly for their sections* (CR-1).

Conjecture 21 has important algorithmic consequences. The exponential complexity in Theorem 1 comes from the exponential growth of the rational number coefficients (heights) in the HNF. This is due to the use of Lemma 17, whose complexity depends on the maximal root of the leading polynomial $p_{i^*,j^*}(n,k)$ from (CR-2). If Conjecture 21 holds, then $p_{i^*,j^*}(n,k) = 1$, Lemma 17 would yield a PTIME algorithm for zeroness, and consequently all complexities in Theorem 2, would drop by one exponential. This provides ample motivation to investigate the monicity conjecture.

In order to obtain the lower EXPTIME complexity for $L(A) \subseteq L(B)$ in Theorem 2 we have to fix the number of registers in *both* automata A and B. The EXPSPACE upper bound of Mottet and Quaas [25] holds already when only the number of registers of B is fixed, while we only obtain a 2-EXPTIME upper bound in this case. It is left for future work whether the counting approach can yield better bounds without fixing the number of registers of A.

The fact that the automata are non-guessing is crucial in each of the cases **I**, **II**, and **III** of the equations in Figure 2 in order to correctly count the number of orbits of runs. For automata with guessing from the fact that the current input a is stored in a register we cannot deduce that a actually appeared previously in the input word w, and thus our current parametrisation in terms of length and width does not lead to a recursive characterisation.

Finally, it is also left for further work to extend the counting approach to other data domains such as total order atoms, random graph atoms, etc..., and, more generally, to arbitrary homogeneous and ω-categorical atoms under suitable computability assumptions (cf. [13]), and to other models of computation such as register pushdown automata [10, 26].

References

1. Christel Baier, Stefan Kiefer, Joachim Klein, Sascha Klüppelholz, David Müller, and James Worrell. Markov Chains and Unambiguous Büchi Automata. In Swarat Chaudhuri and Azadeh Farzan, editors, *Proc. of CAV'16*, pages 23–42, Cham, 2016. Springer International Publishing.
2. Corentin Barloy and Lorenzo Clemente. Bidimensional linear recursive sequences and universality of unambiguous register automata. *arXiv e-prints*, January 2021. `arXiv:2101.01033`.
3. Mikołaj Bojańczyk. Slightly infinite sets, 2019. URL: `https://www.mimuw.edu.pl/~bojan/paper/atom-book`.
4. Alin Bostan, Arnaud Carayol, Florent Koechlin, and Cyril Nicaud. Weakly-Unambiguous Parikh Automata and Their Link to Holonomic Series. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *Proc. of ICALP'20*, volume 168 of *LIPIcs*, pages 114:1–114:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
5. Nicolas Bousquet and Christof Löding. Equivalence and inclusion problem for strongly unambiguous büchi automata. In Adrian-Horia Dediu, Henning Fernau, and Carlos Martín-Vide, editors, *Proc. of LATA'10*, pages 118–129, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.
6. Mireille Bousquet-Mélou. Algebraic generating functions in enumerative combinatorics and context-free languages. In Volker Diekert and Bruno Durand, editors, *Proc. of STACS'05*, pages 18–35, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.
7. Michaël Cadilhac, Alain Finkel, and Pierre McKenzie. Unambiguous constrained automata. In Hsu-Chun Yen and Oscar H. Ibarra, editors, *Proc. of DLT'12*, volume 7410 of *LNCS*, pages 239–250. Springer Berlin Heidelberg, 2012.
8. Michaël Cadilhac, Filip Mazowiecki, Charles Paperman, Michał Pilipczuk, and Géraud Sénizergues. On Polynomial Recursive Sequences. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *Proc. of ICALP'20*, volume 168 of *LIPIcs*, pages 117:1–117:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.
9. Peter J. Cameron. *Notes on Counting: An Introduction to Enumerative Combinatorics*. Australian Mathematical Society Lecture Series. Cambridge University Press, 1 edition, 2017.
10. Edward Y. C. Cheng and Michael Kaminski. Context-free languages over infinite alphabets. *Acta Inf.*, 35(3):245–267, 1998.
11. N. Chomsky and M. P. Schützenberger. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, volume 35 of *Studies in Logic and the Foundations of Mathematics*, pages 118–161. Elsevier, 1963.
12. Lorenzo Clemente. On the complexity of the universality and inclusion problems for unambiguous context-free grammars. In Laurent Fribourg and Matthias Heizmann, editors, Proceedings 8th International Workshop on *Verification and Program Transformation* and 7th Workshop on *Horn Clauses for Verification and Synthesis,* Dublin, Ireland, 25-26th April 2020, volume 320 of *EPTCS*, pages 29–43. Open Publishing Association, 2020. `doi:10.4204/EPTCS.320.2`.
13. Lorenzo Clemente and Slawomir Lasota. Reachability analysis of first-order definable pushdown systems. In Stephan Kreutzer, editor, *Proc. of CSL'15*, volume 41 of *LIPIcs*, pages 244–259, Dagstuhl, 2015.
14. Thomas Colcombet. Forms of Determinism for Automata (Invited Talk). In Christoph Dürr and Thomas Wilke, editors, *Proc. of STACS'12*, volume 14 of *LIPIcs*, pages 1–23, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.
15. Thomas Colcombet. Unambiguity in automata theory. In Jeffrey Shallit and Alexander Okhotin, editors, *Descriptional Complexity of Formal Systems*, pages 3–18, Cham, 2015. Springer International Publishing.
16. Wojciech Czerwiński, Diego Figueira, and Piotr Hofman. Universality Problem for Unambiguous VASS. In Igor Konnov and Laura Kovács, editors, *Proc. of CONCUR'20*, volume 171 of *LIPIcs*, pages 36:1–36:15, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

17 Laure Daviaud, Marcin Jurdzinski, Ranko Lazic, Filip Mazowiecki, Guillermo A. Pérez, and James Worrell. When is Containment Decidable for Probabilistic Automatal. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *Proc. of ICALP'18*, volume 107 of *LIPIcs*, pages 121:1–121:14, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

18 Stéphane Demri and Ranko Lazić. LTL with the freeze quantifier and register automata. *ACM Trans. Comput. Logic*, 10(3):16:1–16:30, April 2009.

19 Mark Giesbrecht and Myung Sub Kim. Computing the Hermite form of a matrix of Ore polynomials. *Journal of Algebra*, 376:341–362, 2013.

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

21 Michael Kaminski and Nissim Francez. Finite-memory automata. *Theoretical Computer Science*, 134(2):329–363, 1994.

22 Michael Kaminski and Daniel Zeitlin. Finite-memory automata with non-deterministic reassignment. *International Journal of Foundations of Computer Science*, 21(05):741–760, 2010.

23 Werner Kuich. On the multiplicity equivalence problem for context-free grammars. In *Proceedings of the Colloquium in Honor of Arto Salomaa on Results and Trends in Theoretical Computer Science*, pages 232—-250, Berlin, Heidelberg, 1994. Springer-Verlag.

24 Dugald Macpherson. A survey of homogeneous structures. *Discrete Math.*, 311(15):1599–1634, August 2011.

25 Antoine Mottet and Karin Quaas. The containment problem for unambiguous register automata and unambiguous timed automata. *Theory of Computing Systems*, 2020. `doi:10.1007/s00224-020-09997-2`.

26 A.S. Murawski, S.J. Ramsay, and N. Tzevelekos. Reachability in pushdown register automata. *Journal of Computer and System Sciences*, 87:58–83, 2017.

27 Frank Neven, Thomas Schwentick, and Victor Vianu. Finite state machines for strings over infinite alphabets. *ACM Trans. Comput. Logic*, 5(3):403—-435, July 2004.

28 Oystein Ore. Linear equations in non-commutative fields. *Annals of Mathematics*, 32(3):463–477, 1931. URL: `http://www.jstor.org/stable/1968245`.

29 Oystein Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34(3):480–508, 1933. URL: `http://www.jstor.org/stable/1968173`.

30 Mikhail Raskin. A Superpolynomial Lower Bound for the Size of Non-Deterministic Complement of an Unambiguous Automaton. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *Proc. of ICALP'18*, volume 107 of *LIPIcs*, pages 138:1–138:11, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

31 Arto Salomaa and Marti Soittola. *Automata-theoretic aspects of formal power series.* Texts and Monographs in Computer Science. Springer, 1978.

32 James Schmerl. A decidable $\aleph_0$-categorical theory with a non-recursive Ryll-Nardzewski function. *Fundamenta Mathematicae*, 98(2):121–125, 1978.

33 Luc Segoufin. Automata and logics for words and trees over an infinite alphabet. In Zoltán Ésik, editor, *Computer Science Logic*, volume 4207 of *LNCS*, pages 41–57. Springer Berlin Heidelberg, 2006.

34 Géraud Sénizergues. The equivalence problem for deterministic pushdown automata is decidable. In Pierpaolo Degano, Roberto Gorrieri, and Alberto Marchetti-Spaccamela, editors, *Proc. of ICALP'97*, pages 671–681, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.

35 Richard P. Stanley. Differentiably finite power series. *European Journal of Combinatorics*, 1(2):175–188, 1980.

36 Richard P. Stanley. *Enumerative Combinatorics.* The Wadsworth & Brooks/Cole Mathematics Series 1. Springer, 1 edition, 1986.

37 R. Stearns and H. Hunt. On the equivalence and containment problems for unambiguous regular expressions, grammars, and automata. In *Proc. of SFCS'81*, pages 74–81, Washington, DC, USA, 1981. IEEE Computer Society. doi:10.1109/SFCS.1981.29.

38 L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time (preliminary report). In *Proc. of STOC'73*, pages 1–9, New York, NY, USA, 1973. ACM.

39 Tzeng Wen-Guey. On path equivalence of nondeterministic finite automata. *Information Processing Letters*, 58(1):43–46, 1996.