A Robust Class of Linear Recurrence Sequences

Corentin Barloy
École Normale Supérieure de Paris, France

Nathanaël Fijalkow
CNRS, LaBRI, Bordeaux, France
The Alan Turing Institute of data science, London, United Kingdom

Nathan Lhote
University of Warsaw, Poland

Filip Mazowiecki
LaBRI, Université de Bordeaux, France

Abstract

We introduce a subclass of linear recurrence sequences which we call poly-rational sequences because they are denoted by rational expressions closed under sum and product. We show that this class is robust by giving several characterisations: polynomially ambiguous weighted automata, copyless cost-register automata, rational formal series, and linear recurrence sequences whose eigenvalues are roots of rational numbers.

2012 ACM Subject Classification Theory of computation → Models of computation

Keywords and phrases linear recurrence sequences, weighted automata, cost-register automata

Digital Object Identifier 10.4230/LIPIcs.CSL.2020.9

Related Version The full version is available on arXiv [4] at `https://arxiv.org/abs/1908.03890`.

Funding The second author was supported by the CODYS project ANR-18-CE40-0007.

Acknowledgements We thank Théodore Lopez for reporting a maths typo in Lemma 10, S. Akshay for fruitful discussions, and the anonymous reviewers for their useful suggestions.

1 Introduction

The study of sequences of numbers originated in mathematics and has deep connections with many fields. A prominent class of sequences is that of *linear recurrence sequences*, such as the Fibonacci sequence

$$0, 1, 1, 2, 3, 5, 8, 13, \dots$$

Despite the simplicity of linear recurrence sequences many problems related to them remain open, and are the object of active research. In theoretical computer science the two main questions are:

- How to finitely represent sequences?
- How to algorithmically analyse properties of sequences?

In this paper we focus on problems related to the first question. The question of representation has led to important insights in the structure of linear recurrence sequences by giving several equivalent characterisations, some of which we briefly review here. We refer to Section 2 and the next sections for technical definitions.

Linear recurrence sequences. A sequence of ratioanl numbers $\mathbf{u} = \langle u_n \rangle_{n \in \mathbb{N}} = \langle u_0, u_1, u_2, \ldots \rangle$ is a linear recurrence system (LRS) if there exist real numbers $a_1, \ldots, a_k$ such that for all $n \geq 0$

$$u_{n+k} \;=\; a_1 u_{n+k-1} + \ldots + a_k u_n. \tag{1}$$

In this paper we will consider only sequences of rational numbers, therefore, we additionally assume that a_i are rational numbers. The smallest k for which $\mathbf{u}$ satisfies an equation of the form (1) is called the order of $\mathbf{u}$. The Fibonacci sequence $\langle F_n \rangle_{n \in \mathbb{N}}$ is an LRS of order 2 satisfying the recurrence $F_{n+2} = F_{n+1} + F_n$.

Rational expressions. Studying the closure properties of linear recurrence sequences yields the following result, an instance of the Kleene-Schützenberger theorem [20]: linear recurrence sequences form the smallest class of sequences containing the sequences $\langle a, 0, 0, \ldots \rangle$ for a rational number a and closed under sum, Cauchy product, and Kleene star.

Weighted automata. The model of weighted automata is a well studied quantitative extension of classical automata. In general a weighted automaton recognises a function $f : \Sigma^* \to \mathbb{R}$, hence when considering a unary alphabet this becomes $f : \{a\}^* \to \mathbb{R}$, and identifying $\{a\}^*$ with $\mathbb{N}$ we can see f as a sequence of numbers. Whenever we write about sequences recognised by models like weighted automata, we implicitly assume that these are over a unary alphabet.

Cost-register automata. Several characterisations of weighted automata have been introduced [6, 12, 3]. We will be interested in the model of *cost-register automata* (CRA). These are deterministic models with registers whose contents are blindly updated (i.e., without transitions like zero tests). It was shown that considering linear updates yields a model equivalent to weighted automata.

We summarise in one theorem the equivalences above, which is the starting point of our work. Technical definitions are given in the paper.

▶ **Theorem 1** (Folklore, see for instance [5, 20, 7]). *The following classes of sequences are effectively equivalent.*

- *Linear recurrence sequences,*
- *Sequences recognised by weighted automata,*
- *Sequences recognised by linear cost-register automata,*
- *Sequences denoted by rational expressions,*
- *Sequences whose formal series are rational, i.e. of the form $\frac{P}{Q}$ where P, Q are polynomials.*

Algorithmic analysis of linear recurrence sequences

The questions regarding algorithmic analysis are far from being answered. A very simple and natural problem, the Skolem problem, is still unsolved [21, 18]: given a linear recurrence sequence, does it contain a zero? Recent breakthrough results sharpened our understanding of the Skolem problem [16, 17], but one of the outcomes is that the general problem for the whole class of linear recurrence sequences is beyond our reach at the moment, since it would impact notoriously difficult problems from number theory. We refer the reader to the recent survey about what is known to be decidable for linear recurrence sequences [18].

Our contributions

Since the full class of linear recurrence sequences is too hard to be algorithmically analysed (we only mentioned the Skolem problem but many related problems are also difficult), let us revise our ambitions, go back to the drawing board, and study tractable subclasses.

In this paper we introduce *poly-rational sequences* which is a strict fragment of linear recurrence sequences. We give several equivalent characterisations of this class following the equivalence results stated in Theorem 1. Our results are summarised in the following theorem.

▶ **Theorem 2.** *The following classes of sequences are effectively equivalent.*

- *Sequences denoted by poly-rational expressions (Section 2),*
- *Sequences recognised by polynomially ambiguous weighted automata (Section 3),*
- *Sequences recognised by copyless cost-register automata (Section 4),*
- *Sequences whose formal series are of the form $\frac{P}{Q}$ where P, Q are polynomials and the roots of Q are roots of rational numbers (Section 5),*
- *Linear recurrence sequences whose eigenvalues are roots of rational numbers (Section 5).*

We do not discuss the efficiency of reductions proving the equivalences. Our constructions are elementary, and in most cases they yield blow ups in the size of representation.

We note that the Skolem problem and its variants are known to be decidable, and NP-hard, for the subclass of poly-rational sequences. The decidability easily follows from the fact that our class is subsumed by other classes for which such results were obtained (see e.g. [19], for the case where all eigenvalues are roots of algebraic real numbers). The Skolem problem is known to be NP-hard already for the class of LRS whose eigenvalues are roots of unity [1]. This implies that the Skolem problem for the class of poly-rational sequences is also NP-hard, which is the best known lower bound even for the full class of linear recurrence sequences.

Related works

The intractability of the Skolem problem for linear recurrence sequences also impacts the other equivalent models, leading to the study of several restrictions. A classical approach to tame weighted automata is to bound the ambiguity of weighted automata, i.e. bounding the number of accepting runs with a function depending on the length of the word. Many positive results have been obtained in the past years following this approach [11, 10, 8].

Another restriction studied in the model of cost-register automata is the *copyless* restriction: registers are not allowed to be copied more than once. It was conjectured that the copyless restriction would result in good decidability properties [3], but this has been recently falsified [2].

2 Linear recurrence sequences and rational expressions

We let $\mathbf{u} = \langle u_n \rangle_{n \in \mathbb{N}} = \langle u_0, u_1, u_2 \dots \rangle$ denote a sequence of rational numbers.

Linear recurrence sequences

We will assume that an LRS $\mathbf{u}$ is given by the numbers $a_1, \dots, a_k$ and the values of the first k elements: $u_0, \dots, u_{k-1}$. The recurrence (1) induces the sequence $\mathbf{u}$. We let **LRS** denote the class of LRS. Given an LRS we define its characteristic polynomial as

$$Q(x) \;=\; x^k - a_1 x^{k-1} - \dots - a_{k-1} x - a_k.$$

The roots of the characteristic polynomial are called the *eigenvalues* of the LRS.

Formal series

Formal series are a different representation for sequences. The sequence $\langle u_n \rangle_{n\in\mathbb{N}}$ induces the formal series $S(x) = \sum_{n\in\mathbb{N}} u_n x^n$, with the interpretation that the coefficient of x^n is the value of the n-th element in the sequence. Note that a polynomial represents a sequence with a finite support.

▶ **Example 3.** A standard example of an LRS is the Fibonacci sequence $\langle F_n \rangle_{n\in\mathbb{N}}$ defined by the recurrence $F_{n+2} = F_{n+1} + F_n$ and initial values $F_0 = 0, F_1 = 1$. Its characteristic polynomial is $p(x) = x^2 - x - 1$, whose roots are $\frac{1+\sqrt{5}}{2}$ and $\frac{1-\sqrt{5}}{2}$. The corresponding formal series is $S(x) = \sum_{n=0}^{\infty} F_n x^n$. Using the definition of F we obtain $S(x) = x + xS(x) + x^2 S(x)$ and thus $S(x) = \frac{x}{1-x-x^2}$.

Rational expressions

We start by defining three classes of sequences.

- **Fin**: a sequence $\mathbf{u}$ is in **Fin**, or equivalently $\mathbf{u}$ has finite support, if the set $\{n \in \mathbb{N} : u_n \neq 0\}$ is finite;
- **Arith**: a sequence $\mathbf{u}$ is in **Arith**, or equivalently $\mathbf{u}$ is arithmetic, if $u_0 = a$, $u_{n+1} = u_n + b$ for some rational numbers a, b;
- **Geo**: a sequence $\mathbf{u}$ is in **Geo**, or equivalent $\mathbf{u}$ is geometric, if $u_0 = a$, $u_{n+1} = \lambda \cdot u_n$, for some rational numbers a, λ.

We let $\mathbf{Geo}_\lambda$ denote the class of geometric sequences with a fixed parameter λ.

We now define some classical operators. Here $\mathbf{u}, \mathbf{v}, \mathbf{u}^1, \ldots, \mathbf{u}^k$ are sequences.

- **Sum**: $\mathbf{u} + \mathbf{v}$ is the component wise sum of sequences;
- **Cauchy product**: $\mathbf{u} \cdot \mathbf{v} = \langle \sum_{p+q=n} u_p \cdot v_q \rangle_{n\in\mathbb{N}}$; inducing $(\mathbf{u})^n$ defined by $(\mathbf{u})^0 = \langle 1, 0, 0, 0, \ldots \rangle$ and $(\mathbf{u})^{n+1} = (\mathbf{u})^n \cdot \mathbf{u}$, in particular $(\mathbf{u})^1 = \mathbf{u}$;
- **Kleene star**: $(\mathbf{u})^* = \sum_{n\in\mathbb{N}} (\mathbf{u})^n$, it is only defined when $u_0 = 0$;
- **Hadamard product**: $\mathbf{u} \times \mathbf{v}$ is the component wise product of sequences;
- **Shift**: $\langle a, \mathbf{u} \rangle = \langle a, u_0, u_1, \ldots \rangle$, defined for any rational number a;
- **Shuffle**: $\mathrm{shuffle}(\mathbf{u}^1, \mathbf{u}^2, \ldots, \mathbf{u}^k) = \langle u_0^1, u_0^2, \ldots, u_0^k, u_1^1, u_1^2, \ldots, u_1^k, u_2^1, \ldots \rangle$.

We write $\mathbf{Rat}[\mathcal{C}, \mathrm{op}_1, \ldots, \mathrm{op}_k]$ for the smallest class of sequences containing $\mathcal{C}$ and closed under the operators $\mathrm{op}_1, \ldots, \mathrm{op}_k$. Rational expressions in Theorem 1 are classically defined as follows [20]:

$$\mathbf{Rat} = \mathbf{Rat}[\mathbf{Fin}, +, \cdot, *].$$

The class **Rat** contains all classes defined above, and is closed under all mentioned operators, i.e.

$$\mathbf{Rat} = \mathbf{Rat}[\mathbf{Fin} \cup \mathbf{Arith} \cup \mathbf{Geo}, +, \cdot, *, \times, \mathrm{shift}, \mathrm{shuffle}].$$

We now introduce a class of sequences denoted by a fragment of rational expressions, whose study is the purpose of this article. The class is called poly-rational sequences, because they are denoted by rational expressions using sum and product.

▶ **Definition 4** (Poly-rational sequences)**.**

$$\mathbf{PolyRat} = \mathbf{Rat}[\mathbf{Arith} \cup \mathbf{Geo}, +, \times, \mathit{shift}, \mathit{shuffle}].$$

In other words **PolyRat** is the smallest class of sequences containing arithmetic and geometric sequences that is closed under sum, Hadarmard product, shift, and shuffle. A trivial observation is that **Fin** $\subseteq$ **PolyRat** since using shift one can generate any sequence with finite support. One could try to simplify the definition of **PolyRat** replacing **Arith** $\cup$ **Geo** with **Fin**. Unfortunately, the operators $+, \times,$ shift, shuffle are too restricted, and geometric and arithmetic sequences could not be generated. In fact, the class would collapse to **Fin**.

Since **Rat** contains **Arith** and **Geo** and is closed under Hadamard product, shift, and shuffle, we have **PolyRat** $\subseteq$ **Rat**. We will show that the inclusion is indeed strict. As we will see in this paper, the class **PolyRat** has many equivalent and surprising characterisations.

3 Characterisation with polynomially ambiguous weighted automata

We refer to e.g. [7] for an excellent introduction to weighted automata. We consider weighted automata over the rational semiring $(\mathbb{Q}, +, \cdot)$, where $+$ and $\cdot$ are the standard sum and product. For an alphabet Σ, weighted automata recognise functions assigning rational numbers to finite words, i.e. $f : \Sigma^* \to \mathbb{Q}$. In this paper we will consider only one-letter alphabets so the set of words is $\{a\}^* = \{\varepsilon, a, a^2, \ldots\}$, which is identified with $\mathbb{N}$. Therefore, weighted automata recognise functions $f : \mathbb{N} \to \mathbb{Q}$, i.e. weighted automata recognise sequences of rational numbers.

Formally, a weighted automaton is a tuple $\mathcal{A} = (Q, M, I, F)$, where Q is a finite set of states, M is a $Q \times Q$ matrix over $\mathbb{Q}$ and I, F are the initial and final vectors, respectively, of dimension Q (for convenience we label the coordinates by elements of Q). The sequence recognised by the automaton $\mathcal{A}$ is $[\![\mathcal{A}]\!]$ defined by $[\![\mathcal{A}]\!](n) = I^t M^n F$, where I^t is the transpose of I.

We give an equivalent definition of $\mathcal{A}$ in terms of accepting runs. We say that a state $q \in Q$ is an initial state if $I(q) \neq 0$ and that it is a final state if $F(q) \neq 0$. If q is initial we say that its initial weight is $I(q)$, and if q is final then its final weight is $F(q)$. For two states $p, q \in Q$ we say that there is a transition from p to q if $M(p, q) \neq 0$. Such a transition is denoted $p \to q$ and its weights is $M(p, q)$. A run ρ is a sequence of consecutive transitions, and it is accepting if the first state is initial and the last state is final. The value of an accepting run $\rho = q_0 \to q_1 \to \cdots \to q_n$ is

$$|\rho| = I(q_0) \cdot \left(\prod_{i=0}^{n-1} M(q_i, q_{i+1}) \right) \cdot F(q_n).$$

Let $Runs_{\mathcal{A}}(n)$ denote the set of all accepting runs of length n. An alternative and equivalent definition of $[\![\mathcal{A}]\!]$ is

$$[\![\mathcal{A}]\!](n) = \sum_{\rho \in Runs_{\mathcal{A}}(n)} |\rho|.$$

▶ **Example 5.** Consider the automaton $\mathcal{A} = (Q, M, I, F)$ represented in Figure 1. We have $[\![\mathcal{A}]\!](n) = F_n$, where $\langle F_n \rangle_{n \in \mathbb{N}}$ is the Fibonacci sequence from Example 3.

The ambiguity of an automaton $\mathcal{A}$ is the function $a_{\mathcal{A}} : \mathbb{N} \to \mathbb{N}$ which associates to n the number of accepting runs $|Runs_{\mathcal{A}}(n)|$. We consider the following classes:

- **DetWA** – the class of deterministic weighted automata, *i.e.* such that for any p, there exists at most one q such that $M(p, q) \neq 0$;
- **kWA** for fixed $k \in \mathbb{N}$ – the class of k-ambiguous weighted automata, i.e. when $a_{\mathcal{A}}(n) \leq k$ for all n;

$$I = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad M = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \quad F = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

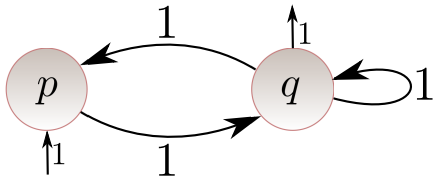

Figure 1 A weighted automaton recognising the Fibonacci sequence.

- **FinWA** $= \bigcup_{k \in \mathbb{N}}$ **kWA** – the class of finitely ambiguous weighted automata, i.e. when there exists k such that $a_{\mathcal{A}}(n) \leq k$ for all n;
- **PolyWA** – class of polynomially ambiguous automata, i.e. when there exists a polynomial $P : \mathbb{N} \to \mathbb{N}$ such that $a_{\mathcal{A}}(n) \leq P(n)$ for all n;
- **WA** – the full class of weighted automata.

For example, the automaton in Example 5 is not polynomially ambiguous because the number of accepting runs is exponential. We will see that this is no accident by proving in Section 5 that the Fibonacci sequence is not in **PolyWA**.

We present our first characterisation of **PolyRat**.

▶ **Theorem 6.** **PolyRat** = **PolyWA**

Proof of Theorem 6

This subsection is divided into two parts for both inclusions.

PolyRat ⊆ PolyWA

Figure 2 shows how to recognise the arithmetic and the geometric sequences. For each finitely

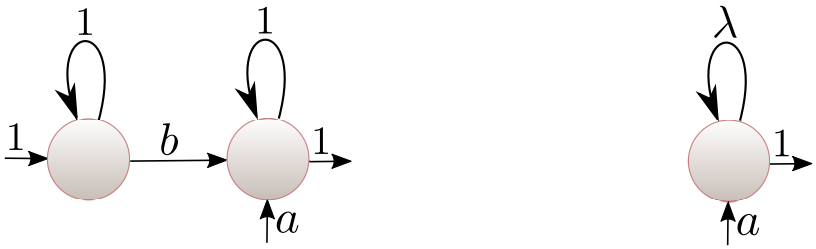

Figure 2 The weighted automaton on the left recognises the arithmetic sequence with parameters (a, b) and it is linearly ambiguous. The weighted automaton on the right recognises the geometric sequence with parameters a, λ and it is deterministic.

supported sequence a simple weighted automaton can be constructed. It remains to prove that the class **PolyWA** is closed under the operators. The sum and products correspond to union and product of automata, it is readily verified that these standard constructions preserve the polynomial ambiguity. Below we deal with shift and shuffle operators.

Suppose we have a polynomially ambiguous automaton $\mathcal{A}$ for $\mathbf{u}$ and we want to construct a new polynomially ambiguous automaton $\mathcal{A}'$ for $\langle a, \mathbf{u} \rangle$. We start with the case when $a = 0$. Then $\mathcal{A}'$ has the same set of states as $\mathcal{A}$ plus one new state q_0, which is the only initial state in $\mathcal{A}'$. All transitions from $\mathcal{A}$ are inherited. There are additionally only outgoing transitions from q_0 to all states that are initial in $\mathcal{A}$; the weight of each transition is the initial weight of the corresponding state in $\mathcal{A}$. It is readily verified that $\mathcal{A}'$ recognises $\langle 0, \mathbf{u} \rangle$ and that $\mathcal{A}'$ is

polynomially ambiguous. For $a \neq 0$ it suffices to add one more state that is both initial and final with initial weight 1 and final weight a.

To deal with shuffle we start with the following preliminary construction. Fix some $k > 0$ and a polynomially ambiguous automaton $\mathcal{A}$ recognising $\mathbf{u}$. We construct $\mathcal{A}[k]$ recognising $\mathbf{u}' = \langle \underbrace{u_0, 0, \dots, 0}_{k}, \underbrace{u_1, 0, \dots, 0}_{k}, u_2, \dots \rangle$, i.e. elements u_i are separated by $k-1$ elements with 0. The idea to construct $\mathcal{A}'$ is that the set of states have an additional component $\{0, \dots, k-1\}$, and they behave like $\mathcal{A}$ every k-th step; in the remaining steps they only wait. Formally, the set of states of $\mathcal{A}[k]$ is $Q \times \{0, \dots, k-1\}$, where Q is the set of states of $\mathcal{A}$. The initial (final) states are $(q, 0)$ such that q is initial (final) in $\mathcal{A}$ with the same weight. For every transition $p \to q$ in $\mathcal{A}$ there is a transition $(p, 0) \to (q, 1)$ in $\mathcal{A}[k]$ with the same weight. The remaining transitions are $(q, i) \to (q, (i+1) \mod k)$ with weight 1, defined for every $i > 0$ and every $q \in Q$. It is readily verified that $\mathcal{A}[k]$ recognises $\mathbf{u}'$.

Let $\mathcal{A}_0, \dots, \mathcal{A}_{k-1}$ be polynomially ambiguous automata recognising $\mathbf{u}_0, \dots, \mathbf{u}_{k-1}$. For every $\mathcal{A}_i$ let $\mathcal{A}_i[k]$ be an automaton as above, additionally shifted i times with 0's. Then shuffle$(\mathbf{u}_0, \dots, \mathbf{u}_{k-1})$ is recognised by the disjoint union of $\mathcal{A}_i[k]$.

PolyWA $\subseteq$ PolyRat

The first step is to decompose polynomially ambiguous automata into a union of automata that we will call *chained loops*. We say that the states $p_0, p_1, \dots p_{k-1} \in Q$ form a *loop* if $M(p_i, p_j) \neq 0$ is equivalent to $j = i + 1 \mod k$ and a *path* if $M(p_i, p_j) \neq 0$ is equivalent to $j = i + 1$ (in particular p_{k-1} has no successor). A chained loop of size k is an automaton over the set states of $\{q_0, \dots, q_{k-1}\} \cup P$ such that

- q_0 is the unique initial state;
- $q_0, \dots, q_{k-1}$ form a path;
- each q_i is contained in at most one loop (the states in P are used only as intermediate states in the loops);
- q_{k-1} is the unique final state with $F(q_{k-1}) = 1$.

We define the concatenation of two chained loops $\mathcal{A}_1, \mathcal{A}_2$: this is the chained loop obtained by constructing the union of the two automata with the initial state being the initial state of $\mathcal{A}_1$, the final state being the final state of $\mathcal{A}_2$, and rewiring the output of $\mathcal{A}_1$ to the initial state of $\mathcal{A}_2$, see e.g. Figure 3.

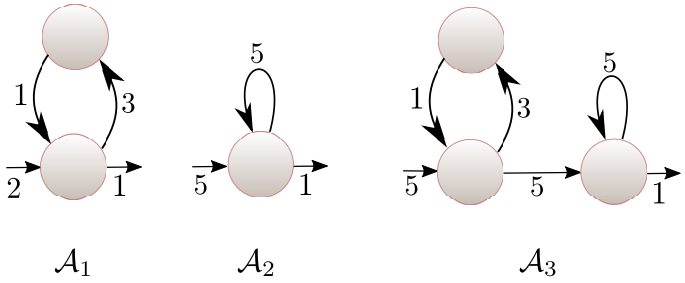

Figure 3 Three example chained loops. The initial and final weights are depicted by ingoing and outgoing edges. The chained loop $\mathcal{A}_1$ recognises the sequence defined by $f_1(2n) = 2 \cdot 3^n$, $f_1(2n+1) = 0$ whose power series is $\frac{2}{1-3x^2}$. The chained loop $\mathcal{A}_2$ recognises the sequence $f_2(n) = 5^{n+1}$ whose power series is $\frac{5}{1-5x}$. The chained loop $\mathcal{A}_3$ is the concatenation of $\mathcal{A}_1$ and $\mathcal{A}_2$ and it recognises the sequence $f_3(n) = \sum_{i=1}^{n} f_1(i-1) \cdot f_2(n-i)$ whose power series is $\frac{10x}{(1-3x^2)(1-5x)}$.

▶ **Lemma 7.** *Every polynomially ambiguous weighted automaton is equivalent to a union of chained loops.*

Proof. Let $\mathcal{A}$ be a polynomially ambiguous weighted automaton. Without loss of generality $\mathcal{A}$ is trimmed, i.e. every state occurs in at least one accepting run.

We first note that any state in $\mathcal{A}$ is contained in at most one loop. Indeed, a state contained in two loops induces a sequence of words with exponential ambiguity. This implies that a sequence $(q_0, q_1, \dots, q_k)$ with $q_i \neq q_j$ for $i \neq j$ induces at most one chained loop of which it is the path. There are finitely many such sequences because k is bounded by the number of states of $\mathcal{A}$.

We claim that $\mathcal{A}$ is equivalent to the union of all chained loops induced by such sequences. Indeed, there is a bijection between the runs of $\mathcal{A}$ and the runs of all the chained loops, respecting the values of runs. Consider a run ρ of $\mathcal{A}$, where a state q appears multiple times. Then between each occurence of q this is the same run, because they are loops over q and there can be only one loop containing q. So $\rho = uv^k w$, where v is the (only) loop containing q. Repeating this for u and w, we obtain a unique decomposition of ρ into

$$q_0 \cdot \ell_0^{m_0} \cdot q_0 \to q_1 \cdot \ell_1^{m_1} \cdot q_1 \to \cdots \to q_k \cdot \ell_n^{m_k} \cdot q_k,$$

where ℓ_i is a loop over q_i (we can have $m_i = 0$) and $q_i \neq q_j$ for $i \neq j$. ◀

Our aim is to use the decomposition result stated in Lemma 7 to prove the inclusion **PolyWA** $\subseteq$ **PolyRat**. It will be convenient for reasoning to use formal series.

▶ **Lemma 8.**

- *The formal series induced by a chained loop of size* 1 *with a loop is of the form* $\frac{\alpha}{1-\lambda x^\ell}$*, where* $\alpha = I(q_0)$*,* λ *is the product of the weights in the loop and* ℓ *is the length of the loop. If there is no loop this reduces to* α*.*
- *Let* S_1, S_2 *be the formal series induced by the chained loops* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*, then the formal series induced by the concatenation of* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *is* $x \cdot S_1 \cdot S_2$*.*
- *Let* S_1, S_2 *be the formal series induced by two automata* $\mathcal{A}_1$ *and* $\mathcal{A}_2$*, then the formal series induced by the union of* $\mathcal{A}_1$ *and* $\mathcal{A}_2$ *is* $S_1 + S_2$*.*

Proof. The first and the third item are immediate, we focus on the second. For convenience let us assume that $\mathcal{A}_1(-1) = 0$. By definition the concatenation of two chained loops recognises the sequence defined by

$$[\![\mathcal{A}]\!](n) = \sum_{i=0}^{n} [\![\mathcal{A}_1]\!](i-1) \cdot [\![\mathcal{A}_2]\!](n-i)$$

since an accepting run in the concatenation is the concatenation of an accepting run in $\mathcal{A}_1$ and an accepting run in $\mathcal{A}_2$. The only issue is that the output state of $\mathcal{A}_1$ was changed into a transition, and to include this step we write $\mathcal{A}_1(i-1)$ instead of $\mathcal{A}_1(i)$. Hence the formal series is indeed the Cauchy product of S_1 and S_2, shifted by one. ◀

We are now half-way through the proof of the inclusion **PolyWA** $\subseteq$ **PolyRat**: thanks to Lemma 7, we can restrict our attention to unions of chained loops, and thanks to Lemma 8, we know what are the formal series induced by the sequences computed by such automata. More specifically, they are obtained from formal series of the form $\frac{\alpha}{1-\lambda x^\ell}$ by taking sums and Cauchy products (with an additional shift).

To prove that **PolyRat** contains such sequences it is tempting to attempt showing that the sequences above are in **PolyRat** and the closure of **PolyRat** under sums and Cauchy

products. Unfortunately, the closure under Cauchy product is not clear (although it will follow from the final result that it indeed holds).

We sidestep this issue by observing that we only need to be able to do Cauchy products of formal series of a special form. Indeed, the formal series described above are of the form $\frac{P}{Q}$ where P, Q are rational polynomials and the roots of Q are roots of rational numbers: this is true of $\frac{\alpha}{1-\lambda x^\ell}$ and is clearly closed under sums and Cauchy products (with the additional shift).

Notice that every chained loop can be obtained as concatenations of chained loops of size 1. Thus Lemma 8 gives a characterisation of formal series corresponding to unions of chained loops: these are sums of products of $\frac{\alpha}{1-\lambda x^\ell}$ and polynomials. We further simplify this characterisation applying the following lemma.

▶ **Lemma 9.** *Consider the formal series $\frac{P}{Q}$ where P, Q are rational polynomials and the roots of Q are roots of rational numbers. Then $\frac{P}{Q}$ can be written as the sum of formal series of the form $\frac{R}{(1-\lambda x^\ell)^k}$ for rational polynomials R, rational numbers λ, and ℓ, k natural numbers.*

Proof. This is a direct consequence of the fact that $\mathbb{Q}[x]$ is a Euclidean ring. The exact statement following from this is that any product $\prod_{i=1}^n \frac{R_i}{P_i}$ where the polynomials P_i are mutually prime (meaning, for each i, the polynomials P_i and $\prod_{j\neq i} P_j$ are coprime) can be written as a sum of $\frac{Q_i}{P_i}$ for some rational polynomials Q_i.

To conclude, we observe that any polynomial whose roots are roots of rational numbers can be written as a product of mutually prime polynomials of the form $(1-\lambda x^\ell)^k$. ◀

By Lemma 8 and Lemma 9 it follows that for every finite union of chained loops its formal series is a sum of $\frac{R}{(1-\lambda x^\ell)^k}$ for rational polynomials R, rational numbers λ, and ℓ, k natural numbers. Combining this with Lemma 7 we get that the formal series computed by **PolyWA** are of the same form. Thus we have reduced proving the inclusion **PolyWA** $\subseteq$ **PolyRat** to proving that sequences whose formal series are sums of formal series of the form $\frac{R}{(1-\lambda x^\ell)^k}$ are in **PolyRat**.

Since **PolyRat** is closed under sum, it suffices to consider one such formal series. Moreover, due to the closure under shifts we can assume that the polynomial R is equal to 1; as stated in the lemma below.

▶ **Lemma 10.** *The sequence whose formal series is $\frac{1}{(1-\lambda x^\ell)^k}$ is in* **PolyRat**.

Proof. We know that

$$\frac{1}{(1-\lambda x^\ell)^k} = \sum_{n\in\mathbb{N}} \binom{n+k-1}{k-1} \lambda^n x^{\ell\cdot n}.$$

Note that $\binom{n+k-1}{k-1}$ is a polynomial in n of degree at most $k-1$, i.e. $\binom{n+k-1}{k-1} = \sum_{p=0}^{k-1} a_p n^p$. It follows that

$$\frac{1}{(1-\lambda x^\ell)^k} = \sum_{p=0}^{k-1} a_p \cdot \sum_{n\in\mathbb{N}} n^p \lambda^n x^{\ell\cdot n}$$

It is enough to prove that for each p the sequence whose formal series is

$$\sum_{n\in\mathbb{N}} a_p n^p \lambda^n x^{\ell\cdot n}$$

is in **PolyRat**. Using an arithmetic sequence and Hadamard products we construct $\langle a_p n^p\rangle_{n\in\mathbb{N}}$. Multiplying it using Hadamard product with the geometric sequence $\langle \lambda^n\rangle_{n\in\mathbb{N}}$ yields $\langle a_p n^p \lambda^n\rangle_{n\in\mathbb{N}}$. Shuffling the obtained sequence with $\ell-1$ null sequences yields the desired sequence. ◀

3.1 Application: the ambiguity hierarchy of weighted automata

We show that the natural classes of weighted automata defined by ambiguity can be described using subclasses of rational expressions.

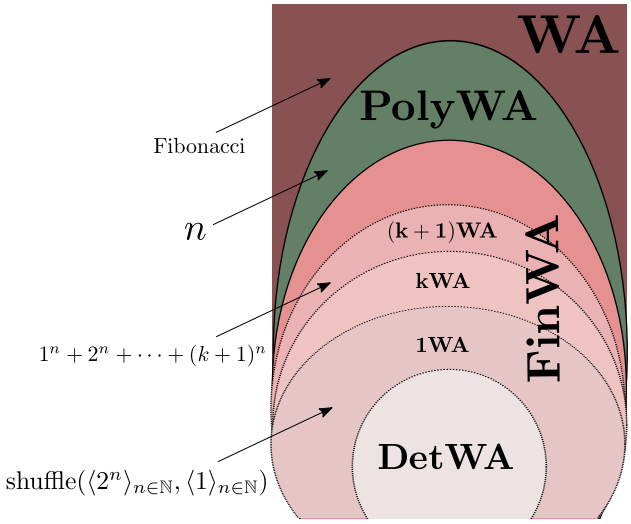

Figure 4 The strict ambiguous hierarchy of weighted automata.

▶ **Lemma 11.**

- $\mathbf{DetWA} = \bigcup_{\lambda\in\mathbb{Q}} \mathbf{Rat}[\mathbf{Geo}_\lambda, \mathit{shift}, \mathit{shuffle}]$;
- $\mathbf{FinWA} = \mathbf{Rat}[\mathbf{Geo}, +, \mathit{shift}, \mathit{shuffle}]$.

Proof. We start by proving $\mathbf{DetWA} = \bigcup_{\lambda\in\mathbb{Q}} \mathbf{Rat}[\mathbf{Geo}_\lambda, \mathrm{shift}, \mathrm{shuffle}]$.

($\subseteq$) Since the automaton is deterministic it has a shape of a lasso, i.e. the states can be partitioned into a path such that the last state on the path is in a loop. Let λ be the value obtained by multiplying all values on the loop, let l be the length of the loop and let m be the length of the path. Then it is easy to see that the sequence is obtained by first taking a shuffle of l sequences in $\mathbf{Geo}_\lambda$ and then shifting it m times.

($\supseteq$) We already know that $\mathbf{Geo}_\lambda$ are definable by deterministic weighted automata from Figure 2. Closure under shift follows from the construction in the proof of $\mathbf{PolyRat} \subseteq \mathbf{PolyWA}$ because it preserves the property of being deterministic. The shuffle construction preserves this property only up to a certain point. The construction of each automaton $\mathcal{A}_i[k]$ is deterministic but taking their sum does not yield explicitly a deterministic automaton. It suffices to observe that by construction $\mathcal{A}_i[k]$ are all lasso automata with loops of the same length. Moreover, every word is accepted by at most one $\mathcal{A}_i[k]$. To define the final automaton consider $\mathcal{A}_i[k]$ with the longest path. The final automaton will be $\mathcal{A}_i[k]$ with modified transitions and final outputs. Indeed we add the automata one by one, and for every accepting state we readjust the ingoing and outgoing transitions to give the correct value.

Proof of $\mathbf{FinWA} = \mathbf{Rat}[\mathbf{Geo}, +, \mathrm{shift}, \mathrm{shuffle}]$.

($\subseteq$) By Lemma 7 we know that each automaton in $\mathbf{FinWA}$ is a union of chained loops. It is easy to see that every such chained loop has to be a lasso otherwise it will contradict the assumption that the automaton is finitely ambiguous. Then the construction follows by doing the construction for every lasso as in the proof of $\mathbf{DetWA} = \bigcup_{\lambda\in\mathbb{Q}} \mathbf{Rat}[\mathbf{Geo}_\lambda, \mathrm{shift}, \mathrm{shuffle}]$ and using $+$ to deal with the union.

($\supseteq$) This follows the same steps as the proof of $\mathbf{DetWA} = \bigcup_{\lambda\in\mathbb{Q}} \mathbf{Rat}[\mathbf{Geo}_\lambda, \text{shift}, \text{shuffle}]$. It is even simpler because we can take a union of two automata and remain in the class of **FinWA**. ◀

We give examples witnessing the strict inclusions $\mathbf{DetWA} \subsetneq \mathbf{FinWA} \subsetneq \mathbf{PolyWA} \subsetneq \mathbf{WA}$ and $\mathbf{kWA} \subsetneq (\mathbf{k+1})\mathbf{WA}$.

▶ **Lemma 12.**

- $\boldsymbol{a} = \textit{shuffle}(\langle 2^n\rangle_{n\in\mathbb{N}}, \langle 1\rangle_{n\in\mathbb{N}})$ *is in* **1WA** *but not in* **DetWA**,
- $\boldsymbol{u_k}$ *defined by* $u_n = 1^n + 2^n + \cdots + (k+1)^n$ *is in* $(\mathbf{k+1})\mathbf{WA}$ *but not in* **kWA**,
- $\boldsymbol{v}$ *defined by* $v_n = n$ *is in* **PolyWA** *but not in* **FinWA**;
- *Fibonacci is in* **WA** *but not in* **PolyWA**.

We omit the simple but technical proofs of the first three items. Only the last item will be proved in Section 5, it follows from the fact that **PolyWA** = **PolyRat** is equal to the class of LRS whose eigenvalues are roots of rational numbers. As mentioned in Example 3 the characteristic polynomial of the Fibonacci sequence is $x^2 - x - 1$, so its eigenvalues are not roots of rationals.

4 Characterisation with copyless cost-register automata

Cost-register automata (CRA) [3] are deterministic automata with write-only registers, where each transition updates the registers using addition and multiplication. Like in Section 3 we will consider only the variant of the model over a one-letter alphabet recognising functions $f : \mathbb{N} \to \mathbb{Q}$.

Let $\mathcal{X}$ be a set of *variables (registers)*. The set of *expressions* $Expr(\mathcal{X})$ is generated by the following grammar

$$e ::= x \mid r \mid e + e \mid e \cdot e,$$

where $x \in \mathcal{X}$ and $r \in \mathbb{Q}$. A *substitution* is a mapping $\nu : \mathcal{X} \to Expr(\mathcal{X})$. We let $Subs(\mathcal{X})$ denote the set of all substitutions. A *valuation* is a function $\sigma : \mathcal{X} \to \mathbb{Q}$, it is a special case of substitutions, where expressions are limited to constants. We freely compose these objects: for instance let $\mathcal{X} = \{x\}$, define the valuation $\nu_0(x) = 0$, the substitution $\sigma(x) = x + 1$ and the expression $e = 2x$. Then $\nu_0 \circ \sigma^n \circ e = 2n$. Note that we use the non-standard order for functional composition. We see this computation as the output of a 1-register machine which initialises x with 0, increments its value at each step and outputs its double value.

Formally, a CRA is a tuple $\mathcal{A} = (Q, \mathcal{X}, \delta, q_0, \nu_0, \mu)$, where Q is the set of states, $\mathcal{X}$ is the set of registers, $\delta : Q \to Q \times Subs(\mathcal{X})$ is the transition function, q_0 is the initial state, $\nu_0 : \mathcal{X} \to \mathbb{Q}$ is the initial valuation and $\mu : Q \to \mathbb{Q}$ is the final output function. The output of $\mathcal{A}$ on n is defined by the unique run of length n: let $q_0 \to q_1 \to \cdots \to q_n$ such that $\delta(q_i) = (q_{i+1}, \sigma_{i+1})$

$$[\![\mathcal{A}]\!](n) = \nu_0 \circ \sigma_1 \circ \cdots \circ \sigma_n \circ \mu(q_n).$$

A CRA is said to be linear if its transitions and output function use only linear expressions, i.e. such that in the grammar $e \cdot e$ is restricted to $e \cdot r$. We let **LCRA** denote the class of sequences recognised by linear CRA, which is known to be equivalent to the class **WA** [3]. The linear CRA represented in Figure 5 recognises the Fibonacci sequence.

A substitution σ is called copyless if each register is used at most once in $\sigma(x)$ for every x. It is easy to observe that a composition of copyless substitutions is a copyless substitution.

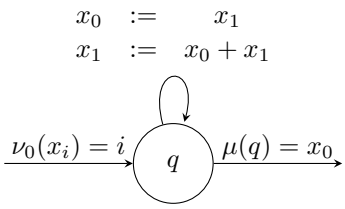

Figure 5 A linear CRA recognising the Fibonacci sequence. There is only one state and two variables $\mathcal{X} = \{x_0, x_1\}$. Since there is only one state the transitions are presented using only the expression that is applied every time.

A CRA is said to be copyless if in each transition, each substitution is copyless. For example in Figure 5 the register x_1 is used twice in the substitution so it is not a copyless automaton. We let **CCRA** denote the class of sequences recognised by copyless cost register automata (CCRA). In [13] it is shown that **CCRA** is a subclass of linear CRA. We show that this is another class characterising **PolyRat**.

▶ **Theorem 13.** **PolyRat** = **CCRA**

PolyRat ⊆ CCRA

This inclusion is easy to prove, it requires to perform the classical constructions as in Section 3 and to note that they respect the copyless restriction.

CCRA ⊆ PolyRat

We make use of a simple property in [15]. A substitution is in *normal form* if there exists an order on the registers $x_1 < \cdots < x_k$ such that the substitutions updating registers respect the order: $\sigma(x_i)$ can use only registers x_j such that $x_j \geq x_i$. A CCRA is in normal form if all substitutions used by it are in normal form, with the same order on the registers. It is known that every CCRA has an equivalent CCRA in normal form [15, Proposition 1]. We will use this fact only to prove Lemma 14, but in the construction we will assume that the CCRA is in normal form.

Consider a CCRA $\mathcal{A}$, we prove that the sequence $\mathbf{u}$ it recognises is in **PolyRat**. We assume without loss of generality that $\mathcal{A}$ is in normal form. Since $\mathcal{A}$ is deterministic it has the shape of a lasso: a tail of length k and a loop of length ℓ. Let us fix $n \in \mathbb{N}$ and $\ell' \in \{0, \dots, \ell - 1\}$, the run is

$$q_0 \to \cdots \to q_k \to (p_0 \to \cdots \to p_{\ell-1})^n \to p_0 \to \cdots \to p_{\ell'}. \tag{2}$$

Let $\delta(q_i) = (q_{i+1 \mod \ell}, \beta_i)$ for $i \in \{0, \dots, k\}$, with the convention that $q_{k+1} = p_0$, and $\delta(p_i) = (p_{i+1 \mod \ell}, \sigma_i)$ for $i \in \{0, \dots, \ell - 1\}$. Define

$$\nu'_0 = \nu_0 \circ \beta_0 \circ \cdots \circ \beta_k \quad ; \quad \sigma = \sigma_0 \circ \cdots \circ \sigma_{\ell-1} \quad ; \quad e = \sigma_0 \circ \cdots \circ \sigma_{\ell'-1} \circ \mu(p_{\ell'}).$$

Notice that σ is a copyless substitution since it is a composition of copyless substitutions. We define the sequence $\mathbf{u}[\ell']$ by

$$u_n[\ell'] = \nu'_0 \circ \sigma^n \circ e.$$

We will prove in Lemma 14 that the sequence $\mathbf{u}[\ell']$ is in **PolyRat**. The decomposition of the runs into a lasso implies the following equality:

$$\mathbf{u} = \langle u_0, u_1, \dots, u_{k-1}, \text{shuffle}(\mathbf{u}[0], \dots, \mathbf{u}[\ell-1]) \rangle,$$

which implies that $\mathbf{u}$ is in **PolyRat**, provided the lemma below is true.

▶ **Lemma 14.** *For every copyless substitution σ in normal form, for all initial valuation ν and for all expression e, the sequence*

$$\langle \nu \circ \sigma^n \circ e \rangle_{n \in \mathbb{N}}$$

is in **PolyRat**.

Proof. We prove that the sequence $\mathbf{u}_x = \nu \circ \sigma^n(x)$ is in **PolyRat** for every register x, i.e. the lemma holds for $e = x$. The general case follows since **PolyRat** is closed under addition and product.

We consider two cases. Suppose x is not used in $\sigma(x)$. We prove that for n big enough the sequence stabilises, i.e. $\sigma^n(x) = \sigma^{n+1}(x) = c$ for some constant c. We show this by induction on the order $<$ from the assumed normal form. If x is the largest element in the order $<$ then $\sigma(x)$ is a constant and thus $\sigma^n(x) = \sigma^{n+1}(x)$. For the induction step suppose x is not the largest element. If $\sigma(x)$ is a constant then the claim is trivial. Otherwise let $x_1, \dots, x_m$ be registers used in $\sigma(x)$. Since σ is copyless then x_i is not used in $\sigma(x_i)$ for every i. Hence by the induction assumption for every i there exists n_i such that $\sigma^n(x_i) = \sigma^{n+1}(x_i)$ for all $n \geq n_i$. It suffices to take $n = \max_i\{n_i \mid 1 \leq i \leq m\} + 1$. Since constant sequences are geometric sequences with $\lambda = 1$ then $\mathbf{u}_x$ can be defined in **PolyRat** using shift.

Now suppose that x is used in $\sigma(x)$. The expression $\sigma(x)$ is equivalent to $\sum_{i=0}^{m} a_i \cdot x_i$ for some constants a_i, where $x_0 = x$ and x_i are pairwise different. Since σ is copyless then for all $i > 0$ we know that $\sigma(x_i)$ does not use x_i. By the previous paragraph there exists N such that $\sigma^N(x_i) = \sigma^{N+1}(x_i) = c_i$ for some constants c_i for all $i > 0$. Let $n \geq N$. Then

$$\nu \circ \sigma^{n+1}(x) \;=\; \nu \circ \sigma^n \circ \sigma(x) \;=\; \nu \circ \left(\sum_{i=0}^{m} a_i \cdot \sigma^n(x_i) \right) \;=\; a_0 \cdot (\nu \circ \sigma^n(x)) + \sum_{i=1}^{m} a_i \cdot c_i.$$

Let $a = a_0$ and $b = \sum_{i=1}^{m} a_i \cdot c_i$. We proved that for $n \geq N$ the sequence $\mathbf{u}_x$ satisfies $u_x(n+1) = a \cdot u_x(n) + b$. It remains to prove that this sequence is in **PolyRat**. It is enough to show that $\mathbf{u}'_x(n) = \mathbf{u}_x(n+N)$ is in **PolyRat** since to obtain $\mathbf{u}_x$ it suffices to use shift N times. There are two cases. If $a = 1$ then $\mathbf{u}'_x(n)$ is an arithmetic sequence, which concludes the proof. If $a \neq 1$ then

$$u'_x(n) \;=\; a^n \cdot u'_x(0) + \sum_{i=0}^{n-1} a^i \cdot b \;=\; a^n \cdot u'_x(0) + b \cdot \frac{a^n - 1}{a - 1}.$$

This is a sum of a geometric sequence $a^n \cdot (u'_x(0) + \frac{b}{a-1})$; and a constant sequence $-\frac{b}{a-1}$; which proves $\mathbf{u}'_x$ is in **PolyRat**. ◀

▶ Remark 15. One can extract from this proof the equivalence between linear **CCRA** and $\mathbf{Rat}[\mathbf{Arith} \cup \mathbf{Geo}, +, \text{shift}, \text{shuffle}]$.

It was recently shown that **CCRA** are strictly less expressive than weighted automata [15]. The proof goes by analysing the Fibonacci sequence. We will get as a corollary of our results a self-contained proof that **LCRA** and **CCRA** are different.

5 Characterisation with linear recurrence sequences and formal series

Our last two characterisations are as follows.

▶ **Theorem 16.** **PolyRat** *is the class of LRS whose eigenvalues are roots of rational numbers, and equivalently whose formal series are* $\frac{P}{Q}$ *with* P, Q *rational polynomials and the roots of* Q *are roots of rational numbers.*

Before proving the theorem, we note that we can now substantiate the claim that the Fibonacci sequence is not in **PolyRat** (hence not in **CCRA** and **PolyWA**), since its eigenvalues are not roots of rational numbers.

We rely on the following classical result about LRS, see e.g. [9].

▶ **Lemma 17.** *Let* $\boldsymbol{u}$ *be an LRS and* Q *its characteristic polynomial. The formal series induced by* $\boldsymbol{u}$ *is* $\frac{P}{Q}$ *for some rational polynomial* P*.*

For both inclusions we rely on Theorem 6 stating that **PolyRat** = **PolyWA** and the decompositions obtained in the subsequent lemmas.

PolyRat ⊆ LRS whose eigenvalues are roots of rational numbers

By Lemma 7 and Lemma 8 the formal series of sequences in **PolyWA** are sums and Cauchy products of formal series of the form $\frac{R}{1-\lambda x^\ell}$, where R is a rational polynomial, $\ell \in \mathbb{N}$ and $\lambda \in \mathbb{Q}$. The roots of $1 - \lambda x^\ell$ are roots of $\frac{1}{\lambda}$, so the roots of the characteristic polynomial are roots of rational numbers.

LRS whose eigenvalues are roots of rational numbers ⊆ PolyRat

Consider an LRS whose eigenvalues are roots of rational numbers. Thanks to Lemma 17 the formal series it induces is $\frac{P}{Q}$ with P, Q rational polynomials and the roots of Q are roots of rational numbers. By Lemma 9 the formal series can be written as a sum of formal series of the form $\frac{R}{(1-\lambda x^\ell)^k}$ for rational polynomials R, rational number λ, and ℓ, k natural numbers. It follows from Lemma 10 and the closure of **PolyRat** under sum and shift that such sequences belong to **PolyRat**.

6 Conclusion

We introduced a class of linear recurrence sequences and obtained several characterisations. The most surprising equivalence is **CCRA** = **PolyWA**. This equality is very particular to our setting: for instance the two classes are incomparable, i.e. neither of the inclusions hold, for tropical semirings [15, 14]. We also conjecture that these classes are incomparable over the rational semiring for general alphabets (of size bigger than 1).

We leave open the precise complexity of the Skolem problem for **PolyRat**. Recent progress has been made for a subclass of **PolyRat** [1]: the Skolem problem for LRS whose eigenvalues are roots of unity is NP-complete. Our class is more general since we consider LRS whose eigenvalues are roots of rational numbers, so the NP-hardness also applies. However the algorithm constructed in [1] does not extend to our class.

References

1 S. Akshay, Nikhil Balaji, and Nikhil Vyas. Complexity of Restricted Variants of Skolem and Related Problems. In *42nd International Symposium on Mathematical Foundations of Computer Science, MFCS 2017, August 21-25, 2017 - Aalborg, Denmark*, pages 78:1–78:14, 2017. doi:10.4230/LIPIcs.MFCS.2017.78.

2 Shaull Almagor, Michaël Cadilhac, Filip Mazowiecki, and Guillermo A. Pérez. Weak Cost Register Automata Are Still Powerful. In *Developments in Language Theory - 22nd International Conference, DLT 2018, Tokyo, Japan, September 10-14, 2018, Proceedings*, pages 83–95, 2018. doi:10.1007/978-3-319-98654-8_7.

3 Rajeev Alur, Loris D'Antoni, Jyotirmoy V. Deshmukh, Mukund Raghothaman, and Yifei Yuan. Regular Functions and Cost Register Automata. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 13–22, 2013. doi:10.1109/LICS.2013.65.

4 Corentin Barloy, Nathanaël Fijalkow, Nathan Lhote, and Filip Mazowiecki. A Robust Class of Linear Recurrence Sequences. *CoRR*, abs/1908.03890, 2019. arXiv:1908.03890.

5 Mireille Bousquet-Mélou. Algebraic Generating Functions in Enumerative Combinatorics and Context-Free Languages. In *STACS 2005, 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005, Proceedings*, pages 18–35, 2005. doi:10.1007/978-3-540-31856-9_2.

6 Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380(1-2):69–86, 2007. doi:10.1016/j.tcs.2007.02.055.

7 Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. Springer, 1st edition, 2009.

8 Nathanaël Fijalkow, Cristian Riveros, and James Worrell. Probabilistic Automata of Bounded Ambiguity. In Roland Meyer and Uwe Nestmann, editors, *28th International Conference on Concurrency Theory, CONCUR 2017, September 5-8, 2017, Berlin, Germany*, volume 85 of *LIPIcs*, pages 19:1–19:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.CONCUR.2017.19.

9 Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete mathematics - a foundation for computer science (2. ed.)*. Addison-Wesley, 1994.

10 Daniel Kirsten and Sylvain Lombardy. Deciding Unambiguity and Sequentiality of Polynomially Ambiguous Min-Plus Automata. In *26th International Symposium on Theoretical Aspects of Computer Science, STACS 2009, February 26-28, 2009, Freiburg, Germany, Proceedings*, pages 589–600, 2009. doi:10.4230/LIPIcs.STACS.2009.1850.

11 Ines Klimann, Sylvain Lombardy, Jean Mairesse, and Christophe Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327(3):349–373, 2004. doi:10.1016/j.tcs.2004.02.049.

12 Stephan Kreutzer and Cristian Riveros. Quantitative Monadic Second-Order Logic. In *28th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2013, New Orleans, LA, USA, June 25-28, 2013*, pages 113–122, 2013. doi:10.1109/LICS.2013.16.

13 Filip Mazowiecki and Cristian Riveros. Maximal Partition Logic: Towards a Logical Characterization of Copyless Cost Register Automata. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, pages 144–159, 2015. doi:10.4230/LIPIcs.CSL.2015.144.

14 Filip Mazowiecki and Cristian Riveros. Pumping Lemmas for Weighted Automata. In *35th Symposium on Theoretical Aspects of Computer Science, STACS 2018, February 28 to March 3, 2018, Caen, France*, pages 50:1–50:14, 2018. doi:10.4230/LIPIcs.STACS.2018.50.

15 Filip Mazowiecki and Cristian Riveros. Copyless cost-register automata: Structure, expressiveness, and closure properties. *Journal of Computer and System Sciences*, 100:1–29, 2019. doi:10.1016/j.jcss.2018.07.002.

16 Joël Ouaknine and James Worrell. On the Positivity Problem for Simple Linear Recurrence Sequences,. In *Automata, Languages, and Programming - 41st International Colloquium,*

ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II, pages 318–329, 2014. doi:10.1007/978-3-662-43951-7_27.

17 Joël Ouaknine and James Worrell. Ultimate Positivity is Decidable for Simple Linear Recurrence Sequences. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, pages 330–341, 2014. doi:10.1007/978-3-662-43951-7_28.

18 Joël Ouaknine and James Worrell. On linear recurrence sequences and loop termination. *SIGLOG News*, 2(2):4–13, 2015. doi:10.1145/2766189.2766191.

19 Rachid Rebiha, Arnaldo Vieira Moura, and Nadir Matringe. On the Termination of Linear and Affine Programs over the Integers. *CoRR*, abs/1409.4230, 2014. arXiv:1409.4230.

20 Marcel Paul Schützenberger. On the Definition of a Family of Automata. *Information and Control*, 4(2-3):245–270, 1961. doi:10.1016/S0019-9958(61)80020-X.

21 Terence Tao. *Structure and randomness: pages from year one of a mathematical blog*. American Mathematical Society Providence, RI, 2008.