Pumping-Like Results for Copyless Cost Register Automata and Polynomially Ambiguous Weighted Automata

Filip Mazowiecki
University of Warsaw, Poland

Antoni Puch
University of Warsaw, Poland

Daniel Smertnig
University of Ljubljana and Institute of Mathematics, Physics, and Mechanics (IMFM), Slovenia

Abstract

In this work we consider two rich subclasses of weighted automata over fields: polynomially ambiguous weighted automata and copyless cost register automata. Primarily we are interested in understanding their expressiveness power. Over the rational field and 1-letter alphabets, it is known that the two classes coincide; they are equivalent to linear recurrence sequences (LRS) whose exponential bases are roots of rationals. We develop two pumping-like results over arbitrary fields with unrestricted alphabets, one for each class. As a corollary of these results, we present examples proving that the two classes become incomparable over the rational field with unrestricted alphabets.

We complement the results by analysing the zeroness and equivalence problems. For weighted automata (even unrestricted) these problems are well understood: there are polynomial time, and even NC^2 algorithms. For copyless cost register automata we show that the two problems are PSPACE-complete, where the difficulty is to show the lower bound.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory; Theory of computation → Quantitative automata

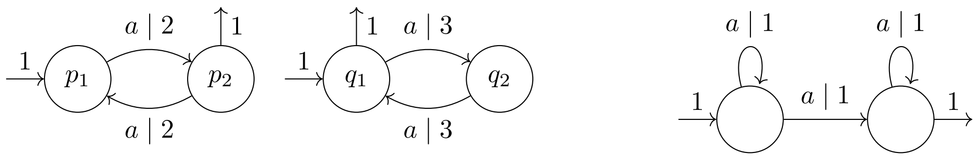
Keywords and phrases weighted automata, cost register automata, ambiguity, linear recurrence sequences, equivalence problem

Funding *Filip Mazowiecki*: Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00230
Antoni Puch: Supported by Polish National Science Centre SONATA BIS-12 grant number 2022/46/E/ST6/00230
Daniel Smertnig: Supported by the Slovenian Research and Innovation Agency (ARIS) program P1-0288 and grant J1-60025

1 Introduction

Weighted automata are a computational model assigning values from a fixed domain to words [11]. The domain can be anything with a semiring structure. Typical examples are: fields [25], where in particular probabilistic automata assign to every word the probability of its acceptance [22]; and tropical semirings, popular due to their connection with star height problems [15]. In this paper we focus on weighted automata over fields. These are finite automata with transitions, input and output edges additionally labeled by weights from the field. On an input word the value of a single run is the product of all weights, and the output of the weighted automaton is the sum of values over all runs. See Figure 1 for examples.

Unlike finite automata, nondeterminism makes weighted automata more expressive. This naturally leads to the decision problem of *determinisation*: given a weighted automaton does there exist an equivalent deterministic one? Over fields, it was recently shown that the problem is decidable [4], later improved to a 2-ExpTime upper bound on the running



(a) A four state unambiguous weighted automaton $\mathcal{A}$ over a 1-letter alphabet $\{a\}$. Nonzero initial labels for p_1 and q_1 have value 1. The nonzero final states are p_2 and q_1, both with weight 1. For every word a^n there are two runs: on the left of value $1 \cdot 2^n$; on the right of value $1 \cdot 3^n$. Depending on the parity of n only one of the runs has nonzero output, thus $\mathcal{A}(a^n) = 3^n$ for even n, and $\mathcal{A}(a^n) = 2^n$, otherwise.

(b) A 2-state polynomially ambiguous weighted automaton $\mathcal{B}$ over the alphabet $\{a\}$. Note that every word a^n has n runs with value 1, hence $\mathcal{B}(a^n) = n$.

Figure 1 Weighted automata over the rational field $\mathbb{Q}(+, \cdot)$. For clarity, we omit zero labels.

time [5]. Both papers rely on Bell and Smertnig's result [3] characterising an indeterminate class of *unambiguous weighted automata*: a subclass that allows nondeterminism, but for every word there is at most one run of nonzero value. The authors proved Reutenauer's conjecture, which we explain below.

Given a weighted automaton $\mathcal{W}$ over the alphabet Σ consider $\mathcal{W}(\Sigma^*)$, the set of all outputs over all words. For example, in Figure 1 we have: $\mathcal{A}(a^*) = \{\, 2^{2n+1},\ 3^{2n} \mid n \in \mathbb{N}\,\}$; and $\mathcal{B}(a^*) = \mathbb{N}$. Reutenauer's conjecture (now Bell and Smertnig's Theorem) states that for every weighted automaton $\mathcal{W}$ over a field K: there exists an equivalent unambiguous weighted automaton if and only if there exists a finitely generated subgroup $G \subseteq K$ such that $\mathcal{W}(\Sigma^*) \subseteq G \cup \{0\}$. For example $\mathcal{A}(a^*) \subseteq G_{\mathcal{A}}$, where $G_{\mathcal{A}}$ is generated by the transition weights $\{2, 3\}$. It is not hard to see that this construction generalises to every unambiguous weighted automaton, the crux is to prove the other implication. As an immediate nontrivial application, notice that there is no unambiguous weighted automaton equivalent to $\mathcal{B}$, as the set $\mathbb{N}$ is not contained in any finitely generated subgroup.

The equivalence problem for weighted automata over fields is famously decidable in polynomial time [25]. However, most natural problems are undecidable [22, 13, 10, 9]. This triggered the study of intermediate classes between deterministic and unrestricted weighted automata. One way to define such a class is based on *ambiguity*, generalising unambiguous weighted automata. A much broader class are *polynomially ambiguous weighted automata*, where the number of accepting runs is bounded by a polynomial in the size of the input word (see Figure 1b). Restricting the input automaton to polynomially ambiguous can significantly lower the complexity of a problem, for example, the discussed problem of determinisation is known to be in PSPACE over the rational field [16]. Another way to define such a class comes from *cost register automata* (CRA), a model which allows for a different characterisation of weighted automata [1]. In this context it is natural to consider its copyless restriction (CCRA), because every function recognisable by a CCRA is also recognisable by a weighted automaton [18] (for simplicity the definition of CCRA is postponed to Section 2).

As far as we know, these two classes, polynomially ambiguous weighted automata and copyless CRA, are the richest studied classes that are known to be strictly contained in the class of unrestricted weighted automata. Over the tropical semiring they are known to be incomparable in terms of expressiveness [19, 8], which suggests the same over fields. One attempt to prove this result was in [2], where the authors considered weighted automata over the rational field with 1-letter alphabets. By identifying a^n with $\mathbb{N}$, one can view such automata as sequences, and in fact they are equivalent to the well-known class of *linear* *recurrence sequences* (LRS) [20]. In [2], the authors prove that both classes coincide, and that they are also equivalent to the class of LRS whose exponential bases are roots of rationals. This means that in the exponential polynomial representation of LRS: $\sum_{i=1}^{n} p_i(x)\lambda_i^x$, for every i there is an n_i such that $\lambda_i^{n_i} \in \mathbb{Q}$. In particular this shows that the Fibonacci sequence does not belong to this class, as the golden ratio φ is not of this form.

Our Contribution

Our work can be seen as a follow-up to [2]. An immediate corollary of our results is that polynomially ambiguous weighted automata and copyless CRA over the rational field are incomparable classes in terms of expressiveness. To prove this, we developed two pumping-like results, one for each class. We believe they are both elegant, phrased in automata-free language, similar to the characterisation of unambiguous weighted automata. Surprisingly, we exploit the behaviour of these classes over 1-letter alphabets, where they coincide. Intuitively, given a weighted automaton $\mathcal{W}$ and three words u, v and w one can associate with them the following LRS: $(\mathcal{W}(uw^nv))_n$. One should think that w plays the role of the single letter in the alphabet. For fixed words one cannot infer anything due to the results in [2]. The idea is to reason collectively about the properties of $(\mathcal{W}(uw^nv))_n$ for all choices of words.

For a simple example illustrating why this approach makes sense, one can look at regular languages and context-free languages, known to be different in general. Due to Parikh's theorem, over 1-letter alphabets the two language classes are equivalent and semi-linear. However, their behaviour can be used to distinguish them in the general case. Given a language L, and words $u, v, w \in \Sigma^*$ consider the characteristic sequence $L_n(u, v, w) \coloneqq uw^nv \in_? L$. If L is regular or context-free, then $L_n(u, v, w)$ is ultimately periodic. Regular languages can be recognised by deterministic automata; thus it is easy to see that there exists a universal m not depending on u, w, v such that for all $n > m$ the language $L_n(u, v, w)$ is, moreover, periodic. Consider the context-free language $L = \{ a^kb^k \mid k \in \mathbb{N} \}$. We observe that it is not regular, since there is no universal m such that $L_n(a^r, \varepsilon, b)$ is periodic for all r and $n > m$.

In detail, for a copyless CRA $\mathcal{C}$ and given a sequence $(\mathcal{C}(uw^{n+1}v))_n$, we consider its exponential polynomial $\sum_{i=1}^{n} p_i(x)\lambda_i^x$. Let us write the polynomials p_i explicitly: $p_i(x) = \sum_{j=1}^{m_i} \alpha_{i,j}x^j$. Then, for every degree k, we define the sum of k-degree coefficients $S_k(u, v, w) = \sum_{i=1}^{n} \alpha_{i,k}$. In Theorem 10 we show that there exists an m such that for all k the set $\{ S_k(u, v, w^m) \mid u, v, w \in \Sigma^* \}$ is contained in a finitely generated subsemiring R. For intuition, if we consider the generators $\{\frac{1}{2}, \frac{1}{3}\}$, by adding, subtracting and multiplying they generate $R = \{ \frac{a}{6^k} \mid a \in \mathbb{Z}, k \in \mathbb{N} \}$. This allows us to give an example of a polynomially ambiguous automaton that is not definable by any copyless CRA (the proof is short but technical, see Example 13).

For polynomially ambiguous weighted automata, our work is inspired by [23], where the authors attempt to characterise polynomially ambiguous weighted automata in a similar manner to Bell and Smertnig's Theorem. For a polynomially ambiguous weighted automaton $\mathcal{W}$ and given a sequence $(\mathcal{W}(uw^nv))_n$ consider again its exponential polynomial $\sum_{i=1}^{n} p_i(x)\lambda_i^x$ and let $E(u, v, w) = \{ \lambda_i \mid 1 \leq i \leq n \}$ be the set of exponential bases. In Theorem 24 we show that the set $\bigcup_{u,v,w\in\Sigma^*} E(u, v, w)$ is contained in a finitely generated subgroup G. We provide a self-contained proof, and we show that our property, which is simpler to work with when considering examples, is equivalent to the one in [23] (conjectured to characterise polynomially ambiguous automata). We obtain a corresponding, simple but technical, example of a copyless CRA that is not definable by any polynomially ambiguous weighted automaton (Example 25).

In this context a natural question is whether our property for copyless CRA can be a characterisation. We conjecture that it is not the case and that, in some sense, such a characterisation should not exist. We show examples of functions that satisfy the property we developed for CCRA, but we find it unlikely that there are CCRA that define them. More generally, in [19] the authors prove that the class of copyless CRA is not closed under reversal for the tropical semiring. More precisely, there is a copyless CRA $\mathcal{C}$ such that there is no copyless CRA $\mathcal{C}'(w) = \mathcal{C}(w^r)$, where w^r is w reversed. We conjecture that over fields copyless CRA are also not closed under reversal, which makes such characterisations unlikely.

Our final contribution is the analysis of the equivalence and zeroness problems for both classes. As already mentioned for weighted automata (even without restrictions) equivalence and zeroness are in polynomial time [25] and even in NC^2 [26]. For copyless CRA the translation to weighted automata [18] yields an exponential blow up in the size of the automaton (we provide a self-contained short translation). Since problems in NC^2 can be solved sequentially in polylogarithmic space [24], this yields a trivial PSpace algorithm. Our contribution is a matching PSpace lower-bound.

Organisation

We start with definitions in Section 2. In Section 3 and Section 4 we prove the properties of copyless CRA, and polynomially ambiguous weighted automata, respectively; and we present examples separating the classes. In Appendix B we discuss the decision problems.

2 Preliminaries

Let $\mathbb{N} := \{0, 1, 2, \dots\}$. For a field K, let $K^\times := K \setminus \{0\}$ denote the multiplicative group of nonzero elements. We sometimes write 1_K and 0_K for the elements 1 and 0 of the field, to emphasize which 1 and 0 we mean.

2.1 Automata and Sequences

A *weighted automaton* over a field K is a tuple $\mathcal{A} = (d, \Sigma, (M(a))_{a\in\Sigma}, I, F)$, where: $d \in \mathbb{N}$ is its dimension; Σ is a finite alphabet; $M(a) \in K^{d\times d}$ are transition matrices; $I, F \in K^d$ are the initial and final vectors, respectively. For simplicity, sometimes we will write $\mathcal{A} = (d, M, I, F)$, that is, we will omit Σ in the tuple.

Weighted automata can be defined more generally over semirings, but in this paper we only consider the case of fields. In this setting, informally, the field of rational numbers $\mathbb{Q}$ is essentially as difficult as the general case of all fields of characteristic 0. Thus, our examples will be for $\mathbb{Q}$ with the usual addition and product, unless stated otherwise.

Given a word $w = w_1 \dots w_n \in \Sigma^*$, we denote $M(w) := M(w_1) \cdot \ldots \cdot M(w_n)$. In particular $M(\epsilon)$ is the $d \times d$-identity matrix. A weighted automaton defines a function $\mathcal{A}\colon \Sigma^* \to K$, by $\mathcal{A}(w) := I^\intercal \cdot M(w) \cdot F$. We say that a weighted automaton $\mathcal{A}$ is a *linear recurrence sequence* (*LRS*) if $|\Sigma| = 1$. Then, by identifying Σ^* with $\mathbb{N}$, that is, identifying the word a^n with its length n, we write that $\mathcal{A}\colon \mathbb{N} \to K$. We will also denote such sequences $(a_n)_n$ instead of $\mathcal{A}$, where $a_n := \mathcal{A}(n)$.

▶ **Example 1.** Consider an LRS $\mathcal{A} = (2, M, I, F)$, where: $M = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$; $I = (1, 0)$ and $F = \left(\begin{smallmatrix} 0 \\ 1 \end{smallmatrix}\right)$. Then $\mathcal{A}(n) = a_n = n$.

For weighted automata, we need to recall a relevant definition of underlying automata. A weighted automaton $\mathcal{A} = (d, \Sigma, (M(a))_{a\in\Sigma}, I, F)$ can be interpreted as an automaton with states $\{1, \dots, d\}$ such that for every $a \in \Sigma$ a nonzero entry in $M_a[i, j]$ defines a transition from i to j labeled by a of weight $M_a[i,j]$ (thus we ignore transitions of weight 0). Similarly, initial and final states are i such that $I[i]$ and $F[i]$ are nonzero, respectively. Their weights are $I[i]$ and $F[i]$. By ignoring the weights of transitions, initial and final states, we obtain a finite automaton $\mathcal{B}$, which we call the *underlying automaton* of $\mathcal{A}$.

▶ **Example 2.** The LRS $\mathcal{A} = (2, M, I, F)$ in Example 1 is an equivalent presentation of the weighted automaton $\mathcal{B}$ in Figure 1b.

A weighted automaton $\mathcal{A}$ is *polynomially ambiguous* if there is a polynomial function $p\colon \mathbb{N} \to \mathbb{N}$ such that for every $w \in \Sigma^*$ the number of accepting runs of the underlying automaton on w is bounded by $p(|w|)$. For example, the automaton in Example 2 is polynomially ambiguous as it suffices to take $p(n) = n$. In general this is a strict subclass: there exist weighted automata that are exponentially ambiguous. More interestingly, there exist functions $f\colon \Sigma^* \to K$ that are recognisable by a weighted automaton, but not by a polynomially ambiguous one.

LRS can be characterised in another way. An LRS $(a_n)_n$ can be defined by a (homogeneous) recurrence relation of the form $a_{n+k} = \sum_{i=0}^{k-1} c_i \cdot a_{n+i}$ with $c_i \in K$ and k initial values a_0, $\dots$, a_{k-1}. Here k is the *order* of the recurrence. For instance, the LRS $(a_n)_n$ from Example 1 can be defined by $a_{n+2} = 2a_{n+1} - a_n$ and $a_0 = 0$, $a_1 = 1$. It is well-known that the two definitions coincide [14] [7, Proposition 2.1]. Moreover, the translation is effective in polynomial time, and under this translation, the dimension d of the weighted automaton equals the order k of the recurrence.

Any given LRS $(a_n)_n$ satisfies many different linear recurrences. However, it is well-known that there is a unique (homogeneous) recurrence of minimal order satisfied by $(a_n)_n$ [6, Ch. 6.1]. The corresponding order k is then the *order* of the LRS. This minimal recurrence gives rise to the *characteristic polynomial* $q = x^k - c_{k-1}x^{k-1} - \cdots - c_0$ of $(a_n)_n$ [6, Ch. 6.1] [12]. The roots of the characteristic polynomial (considered in the algebraic closure $\overline{K}$) are the *characteristic roots* of the LRS $(a_n)_n$.

▶ **Example 3.** Continuing from Example 1, the characteristic polynomial is $q = x^2 - 2x + 1 = (x-1)^2$. Hence, the only characteristic root is 1 (with multiplicity 2).

The characteristic roots of LRS definable by polynomially ambiguous weighted automata are always roots of elements of K [2, 17, 23]. So, for example, the Fibonacci sequence $F_0 = 0$, $F_1 = 1$, $F_{n+2} = F_{n+1} + F_n$, is not recognised by a polynomially ambiguous weighted automaton over $\mathbb{Q}$, as its characteristic roots are the golden ratio $\varphi = \frac{1+\sqrt{5}}{2}$ and $\psi = \frac{1-\sqrt{5}}{2}$.

We recall an additional characterisation of LRS, namely as coefficient sequences of rational functions, leading to exponential polynomials. See also [6, Chapter 6][12][14, Proposition 2.11] or Appendix A. A sequence $(a_n)_n$ is an LRS if and only if the (formal) generating series $F = \sum_{n=0}^{\infty} a_n x^n \in K[\![x]\!]$ is a rational function. For nonzero $\lambda \in \overline{K}$, the following are now equivalent:

- λ is a characteristic root of $(a_n)_n$;
- λ appears as an eigenvalue of $M(a)$ in a weighted automaton representation of $(a_n)_n$ of minimal dimension;
- $1/\lambda$ is a pole of F.

Further, the characteristic roots appear as eigenvalues of $M(a)$ in *every* representation of $(a_n)_n$ using a weighted automaton. But in a weighted automaton that is not of minimal dimension, the matrix $M(a)$ may have additional eigenvalues.

In characteristic 0, every LRS $(a_n)_n$, has, for large enough n, a representation as an *exponential polynomial sequence* (*EPS*):

$$a_n = \sum_{i=1}^{k} q_i(n)\lambda_i^n \quad \text{for sufficiently large n},$$

with q_i polynomials over $\overline{K}$ and $\lambda_i \in \overline{K}$ the nonzero characteristic roots of $(a_n)_n$. Furthermore, the exponential bases λ_i and the polynomials q_i are uniquely determined by $(a_n)_n$.

▶ **Example 4.** The Fibonacci numbers afford the representation $F_n = \frac{1}{\sqrt{5}}\varphi^n - \frac{1}{\sqrt{5}}\psi^n$.

In characteristic $p > 0$, the situation is more complicated (see Appendix A): an LRS may not have a representation by an exponential polynomial (even for large n). If it does have such a representation, it is however still unique as long as the polynomials q_i are chosen of minimal degree, that is, with $\deg(q_i) < p$. Further, every EPS is an LRS, and the exponential bases of the EPS are precisely the nonzero characteristic roots of the LRS.

2.2 Cost Register Automata

We will introduce one more formalism that generalises weighted automata to polynomial updates [1]. A *cost register automaton* (CRA) over a field K is a tuple $\mathcal{C} = (Q, q_0, d, \Sigma, \delta, \mu, \nu)$, where: Q is a finite set of states; $q_0 \in Q$ is the initial state; $d \in \mathbb{N}$ is its dimension; Σ is a finite alphabet; $\delta\colon Q \times \Sigma \to Q \times \mathrm{Poly}^d$ is a deterministic transition function, where Poly^d is the set of d-dimensional polynomial maps; $\mu\colon \{1, \ldots, d\} \to K$ is the initial function; and $\nu\colon Q \to \mathrm{Poly}^d$ is the final function. Here, a *polynomial map* $P \in \mathrm{Poly}^d$ is a tuple $P = (p^1, \ldots, p^d)$ with polynomials $p^i \in K[x_1, \ldots, x_d]$. Every polynomial map induces a function $K^d \to K^d$.

Given $q \in Q$ and $a \in \Sigma$ we write $p_{q,a}$ for the polynomial such that $\delta(q, a) = (q', p_{q,a})$ for some $q' \in Q$. Note that if we ignore the polynomials in δ, then (Q, q_0, Σ, δ) is just a deterministic finite automaton without final states. Thus, given a word w, there is a unique state reachable from q_0 when reading w. We will denote it q_w. For words $w \in \Sigma^+$ we define polynomial maps p_w by induction: if $w = a \in \Sigma$ is a letter then $p_w = p_{q_0,a}$; otherwise if $w = w'a$ for a letter $a \in \Sigma$ then $p_w = p_{q_{w'},a} \circ p_{w'}$.

A CRA defines a function $\mathcal{C}\colon \Sigma^* \to K$, similarly to weighted automata. Formally, given a word $w = w_1 \ldots w_n \in \Sigma^*$ we define $\mathcal{C}(w) = (\nu_{q_w} \circ p_w)(\mu)$. As a simple example we define the automaton recognising the same function as in Example 1. There is only one state, which is also initial, and one letter so for simplicity we will ignore them, also in the transition function. The dimension is 2 and there is only one transition defined by the polynomial map (p^1, p^2) with $p^1(x, y) = x + y$ and $p^2(x, y) = y$. The input function is defined by $\mu(x) = 0$, $\mu(y) = 1$; and the output is the polynomial x.

One can think of the polynomial maps as generalising linear updates definable by matrices. When restricting the model to linear polynomials, the CRA formalism is equivalent to weighted automata [1], and it is called linear CRA. The resulting weighted automaton is of polynomial size in the size of the linear CRA. Note that CRA use separate notions of states (the set Q) and registers (i.e., the dimension d). In general, states are not needed, as one can easily encode the states by enlarging the dimension to $d \times |Q|$, even for linear CRA. However, such encodings do not preserve the copyless restriction on CRA, which we discuss next.

A *copyless* CRA (CCRA) is a CRA such that all polynomial maps in the transition function and the output function are copyless. A polynomial map $P \in \mathrm{Poly}^d$ is copyless if it can be written as an expression using each variable name only once. For example if $d = 3$ then P is defined by three polynomials $p^1(x, y, z)$, $p^2(x, y, z)$ and $p^3(x, y, z)$. If

$p^1 = (x+3)\cdot(y+z)$, $p^2 = 7$ and $p^3 = 1$, then P is copyless; but if $p^1 = y+1$, $p^2 = y$ and $p^3 = z$, then P is not copyless. It is easy to see that copyless polynomial maps are preserved under composition. Thus, in a CCRA all polynomial maps p_w are copyless.

▶ **Lemma 5.** *For every CCRA $\mathcal{C}$ there is a weighted automaton $\mathcal{W}$, of size exponential in the size of $\mathcal{C}$, such that $\mathcal{C}$ and $\mathcal{W}$ are equivalent.*

Proof. Fix a CCRA $\mathcal{C} = (Q, q_0, d, \Sigma, \delta, \mu, \nu)$. We use the fact that linear CRA are equivalent to weighted automata and define an equivalent linear CRA $\mathcal{C}' = (Q, q_0, d', \Sigma, \delta', \mu', \nu')$ (the states remain the same). We note that $\mathcal{C}'$ need not be copyless. The new dimension is $d' = 2^d$ with the following intuition. If the variables $x_1, \ldots, x_d$ represent the registers of $\mathcal{C}$, then the registers in $\mathcal{C}'$ correspond to all possible square-free monomials, i.e., monomials of the form $\prod_{i\in I} x_i$ for every $I \subseteq \{1, \ldots, d\}$ (note that there are 2^d square-free monomials).

We make a simple but useful observation that all monomials in each p_i are square-free. This means that we can maintain as an invariant in $\mathcal{C}'$ all values in $\prod_{i\in I} x_i$ using linear updates (essentially summing the monomials). It remains to define μ' so that the invariant also holds in the first step; and $\nu' = \nu$ (i.e., ignoring all variables except for $x_1, \ldots x_d$). ◀

3 Copyless Cost Register Automata

Throughout the section, fix a field K. In this section we prove a pumping-like result for CCRA. This result will be based on the observation that sequences of the form $(\mathcal{A}(uw^{m(n+1)}v))_n$, obtained from a CCRA $\mathcal{A}$, are always representable by very particular exponential polynomials. To this end, we first introduce the following class of functions.

▶ **Definition 6.** *A K-valued sequence $(a_n)_n$ is an* exponential polynomial sequence generatable from $A \subseteq K$ *(in short, an A-*generatable EPS*) if it can be obtained, using products and sums, from the following families:*

- *Constant sequences $(\alpha)_n$ for $\alpha \in A$,*
- *The linear sequence $(n \cdot 1_K)_n$,*
- *Exponential sequences (that is, geometric progressions) $(\alpha^n)_n$ for $\alpha \in A$,*
- *Sequences of the form $\left(\frac{1}{\alpha-1}\alpha^n - \frac{1}{\alpha-1}\right)_n$ for $1 \neq \alpha \in A$.*

The last family may be a bit unexpected at first glance. It arises from the geometric sum

$$\frac{1}{1-\alpha}\alpha^n - \frac{1}{1-\alpha} = \frac{\alpha^n - 1}{\alpha - 1} = \sum_{i=0}^{n-1} \alpha^i \qquad (\alpha \neq 1),$$

with the representation in Definition 6 corresponding to the normal form for exponential polynomials (with the two exponential bases α and 1_K). Because possibly $1/(1-\alpha) \notin A$, this last family cannot always be generated from the other three families.

▶ **Example 7.** Since $\sum_{i=0}^{n-1} \alpha^i = \alpha\big(\cdots(\alpha(\alpha+1)+1)\big)+1$, the geometric sum appears when iterating a copyless update rule of the form $x \mapsto \alpha x + 1$ from the starting value 1.

Taking $A = K$, the class of K-generatable EPS affords a more familiar description.

▶ **Lemma 8.** *A sequence $(a_n)_n$ is a K-generatable EPS if and only if it is an EPS with coefficients and exponential bases in K.*

Proof. We first check that every K-generatable EPS is indeed an EPS. Since EPS with coefficients and exponential bases in K are closed under products and sums, it suffices to

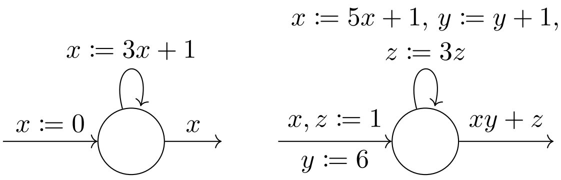


Figure 2 Two simple single-state CCRAs on a single-letter alphabet (Example 11).

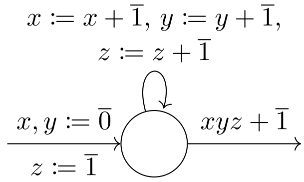


Figure 3 The CCRA over $\mathbb{F}_3$ illustrating coefficient sums in positive characteristic (Example 12).

verify the claimed property for the families in Definition 6. However, each of these families is obviously an EPS and the only exponential bases that appear are 1_K and $\alpha \in K$.

Conversely, suppose that $(a_n)_n$ has a representation $a_n = \sum_{\lambda \in K^\times} \sum_{i \geq 0} \alpha_{\lambda,i} n^i \lambda^n$ with $\alpha_{\lambda,i} \in K$ (only finitely many of which are nonzero). Each of $(\alpha_{\lambda,i})_n$, $(n^i)_n$ and $(\lambda^n)_n$ is clearly a K-generatable EPS, and so is therefore $(a_n)_n$. ◀

Recall that the exponential bases being contained in K is a nontrivial restriction on an EPS. In general, these will be contained in the algebraic closure $\overline{K}$. In particular, every A-generatable EPS is trivially a K-generatable EPS, and hence by Lemma 8 an EPS (in the sense discussed in Section 2), so that our terminology is consistent. Working with, possibly proper, subsets $A \subseteq K$ will be crucial to obtain a pumping-like criterion that is strong enough to differentiate between CCRA and polynomially ambiguous WFA.

We need a final definition before stating our main theorem of the section.

▶ **Definition 9.** *Let $R \subseteq K$ be a subsemiring. An R-CCRA is a CCRA with all of its starting register values, output expression and transition coefficients in R.*

▶ **Theorem 10.** *If $R \subseteq K$ is a subsemiring and $f\colon \Sigma^* \to K$ is recognised by an R-CCRA, then there exists $m \geq 1$ such that,*

- *for every $u, w, v \in \Sigma^*$, the sequence $g(n) = f(uw^{m(n+1)}v)$ is an R-generatable EPS, and*
- *if q is the exponential polynomial of minimal degree representing g, then for every $k \in \mathbb{N}$ the sum of k-degree coefficients $S_k(q)$ is in R.*

The sum of k-degree coefficients $S_k(q)$ is obtained by summing all the coefficients of x^k in q across all the exponential bases (see Appendix A for a detailed discussion).

It is obvious that, for any input, the output of an R-CCRA is in R. However, this is different from the property in Theorem 10 — we make a claim about the coefficients of the exponential polynomial, not the values that it takes. The individual coefficients do not need to always lie in R, as the following example illustrates.

▶ **Example 11.**

- Consider the left $\mathbb{Z}$-CCRA in Figure 2. On words of the form a^{n+1}, this CCRA outputs $q(n) = \mathcal{A}(n+1) = \frac{3^{n+1}-1}{2} = \frac{3}{2} \cdot 3^n - \frac{1}{2} \cdot 1^n$. Even though the automaton itself only uses integer coefficients, a denominator 2 appears in the coefficients of q. However, the sum of the coefficients is $S_0(q) = \frac{3}{2} - \frac{1}{2} = 1$, an integer.
- The second $\mathbb{Z}$-CCRA in Figure 2 outputs

$$\mathcal{B}(a^{n+1}) = \frac{5^{n+2}-1}{4}(n+7) + 3^{n+1} = \left(\tfrac{25}{4}n + \tfrac{175}{4}\right) \cdot 5^n + 3 \cdot 3^n + \left(-\tfrac{1}{4}n - \tfrac{7}{4}\right) \cdot 1^n =: q(n).$$

Here $S_1(q) = \frac{25}{4} - \frac{1}{4} = 6 \in \mathbb{Z}$ and $S_0(q) = \frac{175}{4} + 3 - \frac{7}{4} = 45 \in \mathbb{Z}$.

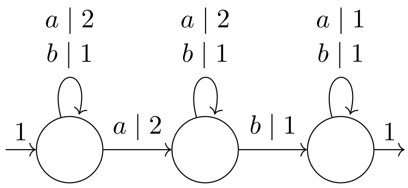


Figure 4 A polynomially ambiguous weighted automaton with no equivalent CCRA (Example 13).

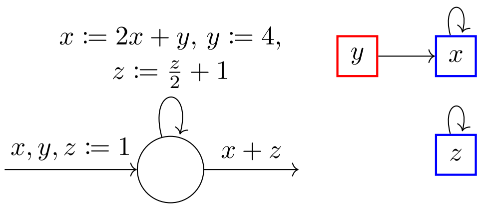


Figure 5 A simple CCRA and its register-dependency graph. Red nodes are constant registers; blue nodes are updating ones (Definition 15).

For fields of characteristic 0, there is a unique exponential polynomial q representing g, and the condition of q having minimal degree in Theorem 10 can be omitted in this case. But in characteristic $p > 0$ there will be multiple exponential polynomials representing g, and the claim is about the one of minimal degree (in which all the occurring polynomials have degree at most $p - 1$). This is explained in Appendix A, and illustrated in the next example.

▶ **Example 12.** Consider the single-letter CCRA $\mathcal{A}$ in Figure 3 over $\mathbb{F}_3 = \{\overline{0}, \overline{1}, \overline{2}\}$. After reading a^n, the registers hold the values $(\overline{n}, \overline{n}, \overline{n}{+}1)$. Hence, $\mathcal{A}(a^n) = \overline{n}^2(\overline{n}{+}\overline{1}){+}\overline{1} = \overline{n}^3{+}\overline{n}^2{+}\overline{1}$. However, the polynomial $x^3 + x^2 + 1$ is not the minimal degree polynomial representing $g(n) = \overline{n}^3 + \overline{n}^2 + 1$. Since $\overline{n}^3 + \overline{n}^2 + 1 = \overline{n}^2 + \overline{n} + 1$, instead $q = x^2 + x + 1$ is the minimal-degree representative. We have $S_0(q) = \overline{1}$, $S_1(q) = \overline{1}$, $S_2(q) = \overline{1}$ and $S_k(q) = \overline{0}$ for all other $k \in \mathbb{N}$.

Before proving Theorem 10, we demonstrate how it can be applied. We use it to show that not every function recognisable by a polynomially ambiguous WFA can be recognised by a CCRA.

▶ **Example 13.** The automaton in Figure 4 is polynomially ambiguous. For any k, m, $n \in \mathbb{N}$ and inputs of the form $(a^k b)^{m(n+1)}$ it outputs

$$g_{k,m}(n) = k2^k + 2k2^{2k} + \cdots + m(n+1)k2^{m(n+1)k} = \sum_{j=1}^{m(n+1)} jk2^{jk}.$$

Using the identity $\sum_{j=1}^{l} jx^j = \frac{lx^{l+2} - (l+1)x^{l+1} + x}{(x-1)^2}$, which can be derived from the geometric sum $\sum_{j=1}^{l} x^j = \frac{x^{l+1} - x}{x-1}$ by formal differentiation and some easy manipulations, one finds

$$g_{k,m}(n) = q_1(n) \cdot (2^{km})^n + q_2(n) \cdot 1^n,$$

with

$$q_1(n) = \frac{km2^{km+k}}{(2^k - 1)} n + \frac{k(m2^k - m - 1)2^{km+k}}{(2^k - 1)^2} \quad \text{and} \quad q_2(n) = \frac{k2^k}{(2^k - 1)^2}.$$

We claim that, for every finitely generated subring $R \subseteq \mathbb{Q}$ and any choice of m, there exists some choice of k such the first coefficient sum $S_1(g_{k,m}) \notin R$. Fix R a finitely generated subring of $\mathbb{Q}$ and $m \geq 1$.

We have $S_1(g_{k,m}) = km2^{km+k}/(2^k - 1)$. For any prime p, we can take $k = p - 1$ and get

$$S_1(g_{p-1,m}) = \frac{(p-1)m2^{(p-1)m+p-1}}{2^{p-1} - 1}.$$

For $p \nmid 2m$, the numerator is not divisible by p. However, by Fermat's Little Theorem, the denominator is. Only a finite set of prime numbers can appear among denominators of elements of R. This means that there exists some prime number p for which $S_1(g_{p-1,m}) \notin R$. As this contradicts the conclusion of Theorem 10, we see that f is not recognisable by a CCRA.

We record the conclusion as a theorem.

▶ **Theorem 14.** *If $|\Sigma| \geq 2$, then there exist functions $f : \Sigma^* \to \mathbb{Q}$ that are recognisable by a polynomially ambiguous weighted automaton, but not by a $\mathbb{Q}$-CCRA.*

Proof. By Example 13 and Theorem 10. Observe that we can assume that R in Theorem 10 is finitely generated. ◀

3.1 The Proof of Theorem 10

The proof of Theorem 10 proceeds in several steps. We start with a very simple case and then, in each step, use the previous result to show a slightly more general one.

▶ **Definition 15.** *A single letter, single state CCRA is* simple *if, in the transition, every register value is either set to a constant* (constant registers)*, or depends only on its old value and on the values of constant registers* (updating registers)*.*

Since there is only one state, there is also only one transition, so the definition makes sense. We can visualise constant and updating registers with a graph representing the dependency of register values on each other (Figure 5).

▶ **Lemma 16.** *If $\mathcal{A}$ is a single state, single letter simple R-CCRA, then $(\mathcal{A}(a^{n+1}))_n$ is an R-generatable EPS.*

Proof. In a simple CCRA, the register values change in very simple ways. For constant registers, after the first transition, they remain set to the same values. For updating registers, the first update is special. However, after that, the input they get from the constant registers stabilizes. Let us consider what happens from that point on. After the first step, the register values are of course still in R. Since the updating registers can only depend on themselves and constants, the update formulas can be reduced to the form $x \coloneqq \alpha x + \beta$ for constants α, $\beta \in R$. This gives us an LRS ($x_{n+1} = \alpha x_n + \beta$). Solving the LRS, we need to distinguish two cases. For $\alpha = 1$, the solution is

$$x_{n+1} = x_1 + \beta n, \qquad \text{and for } \alpha \neq 1 \text{ it is} \qquad x_{n+1} = \alpha^n \frac{\beta}{\alpha - 1} - \frac{\beta}{\alpha - 1} + \alpha^n x_1.$$

In both cases the sequence $(x_{n+1})_n$ is clearly an R-generatable EPS. Constant registers, leading to constant sequences, also clearly are R-generatable EPS.

The output expression combines these sequences using sums and products of the sequences and additional constants from R. These operations preserve the property of being an R-generatable EPS, and so the sequence $(\mathcal{A}(a^{n+1}))_n$ is an R-generatable EPS. ◀

In the next two lemmas we will reason about the behaviour of CCRA on cycles. Similar, but different, observations were made in [19, Proposition 1 and Lemma 4].

▶ **Lemma 17.** *If $\mathcal{A}$ is a single state, single letter R-CCRA (not necessarily simple) with r registers, then $(\mathcal{A}(a^{r!(n+1)}))_n$ is an R-generatable EPS.*

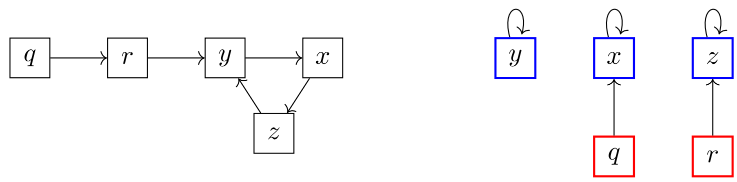


Figure 6 An example for register-dependency graphs of $\mathcal{A}$ and $\mathcal{B}$ in the proof of Lemma 17.

Proof. Consider the compound effect on registers of the letter a being applied $r!$ times. We can get the corresponding expressions simply by composing the substitution $r!$ times. They will still of course be copyless and polynomial, meaning we can create an auxiliary CCRA $\mathcal{B}$ with a 1-letter alphabet such that $\mathcal{B}(a^n) = \mathcal{A}(a^{nr!})$. It is also easy to see, from how substitutions compose, that $\mathcal{B}$ is still an R-CCRA.

We claim that the new CCRA $\mathcal{B}$ is simple: we will prove this by looking at the register-dependency graph of $\mathcal{A}$. Since $\mathcal{A}$ is copyless, there is at most one outgoing edge from any vertex. In $\mathcal{B}$ the expression for register v will use u if and only if in the dependency graph of $\mathcal{A}$ there is a path of length $r!$ from u to v. (This is visualised in Figure 6.)

Consider an arbitrary register t. It will either be in a cycle on the graph of $\mathcal{A}$ or not. Assume t is not in a cycle and there is a path of length $r!$ from some register u to t. Such a path would have to contain a cycle. However, that is impossible, since each vertex has at most one outgoing edge and t itself is not in a cycle. This means t will be a constant register in the auxiliary automaton.

Now assume t is in a cycle in the dependency graph of $\mathcal{A}$, and let l be the length of the cycle. We want to prove that t is an updating register in $\mathcal{B}$. Assume there is a path of length $r!$ from some u to t. To show that t is an updating register, we need to show that either $u = t$ or u is a constant register in $\mathcal{B}$. If u is in the same cycle as t, we have $u = t$, since $l \mid r!$. If u is outside the cycle containing t, then u cannot be a part of any cycle, as any vertex can have at most one outgoing edge. This means that u is a constant register in $\mathcal{B}$. Thus, the auxiliary CCRA $\mathcal{B}$ is simple, and we can apply Lemma 16 to it, finishing the proof. ◀

▶ **Lemma 18.** *If $\mathcal{A}$ is a single letter R-CCRA (not necessarily single state) with s states and r registers, then $(\mathcal{A}(a^{(4r+2)!s!(n+1)}))_n$ is an R-generatable EPS.*

Proof. Consider the compound effect on registers of the letter a being applied $s!$ times. We can get the corresponding transitions between states by looking at paths of length $s!$, and corresponding update expressions by composing appropriate $s!$ substitutions. The updates will of course still be copyless and polynomial, and the transitions deterministic, meaning we can create an auxiliary CCRA $\mathcal{B}$ such that $\mathcal{B}(a^n) = \mathcal{A}(a^{ns!})$. Note that the transition expression coefficients will all still be in R, so $\mathcal{B}$ is still an R-CCRA. Since $\mathcal{A}$ is deterministic, after at most s steps it always reaches a cycle. This cycle has length at most s, and so its length divides $s!$. This means that, after trimming $\mathcal{B}$, we get an automaton of one of the forms in Figure 7.

We want to reduce $\mathcal{B}$ to only one state. The first possible form already has only one state. The second one can easily be simulated with one state, as shown in Figure 8. After this operation, the automaton $\mathcal{B}$ is a single-state R-CCRA with $4r + 2$ registers such that $\mathcal{B}(a^n) = \mathcal{A}(a^{ns!})$. This lets us use Lemma 17 and finishes the proof. ◀

▶ **Lemma 19.** *If $\mathcal{A}$ is an R-CCRA (not necessarily single letter) with r registers and s states, then, for all $w \in \Sigma^*$, the sequence $(\mathcal{A}(w^{(4r+2)!s!(n+1)}))_n$ is an R-generatable EPS.*

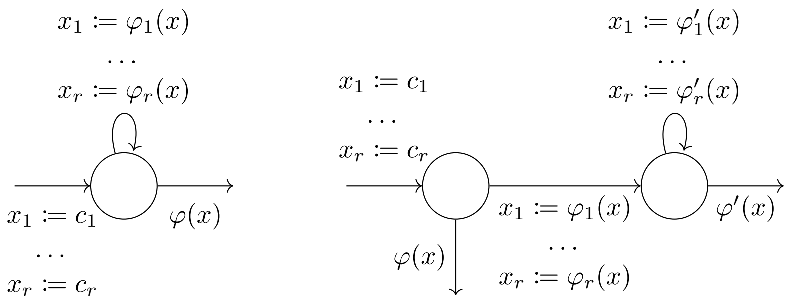


Figure 7 The two possible forms the auxiliary automaton $\mathcal{B}$ can take in the proof of Lemma 18.

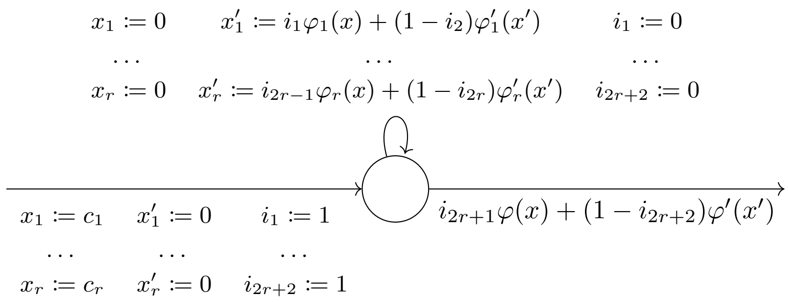


Figure 8 How to transform $\mathcal{B}$ into a single state automaton in the proof of Lemma 18.

Proof. Consider the composite effect of the word w on registers and state transitions. This effect is still copyless, polynomial, deterministic and all the transition coefficients are still in R. We can thus create an auxiliary R-CCRA $\mathcal{B}$ such that $\mathcal{A}(w^n) = \mathcal{B}(a^n)$. The CCRA $\mathcal{B}$ has a one letter alphabet, letting us use Lemma 18 and finishing the proof. ◀

▶ **Lemma 20.** *If $\mathcal{A}$ is an R-CCRA with r registers and s states, then, for all $u, w, v \in \Sigma^*$, the sequence $(\mathcal{A}(uw^{(4r+2)!s!(n+1)}v))_n$ is an R-generatable EPS.*

Proof. Adding some prefix u simply changes the initial register values. The register values are of course still in R. Adding a suffix v simply changes the output expression. Its coefficients are of course still in R. Thus, we obtain an R-CCRA $\mathcal{B}$ with $\mathcal{B}(x) = \mathcal{A}(uxv)$ for all words $x \in \Sigma^*$. By Lemma 19, the sequence $(\mathcal{A}(uw^n v))_n = (\mathcal{B}(w^n))_n$ is an R-generatable EPS. ◀

We also need the next lemma which is proven in Appendix C.

▶ **Lemma 21.** *If R is a subsemiring, $(a_n)_n$ is an R-generatable EPS and q is the exponential polynomial of minimal degree representing $(a_n)_n$, the sum of k-degree coefficients of q is in R.*

We can finally prove the main theorem of Section 3.

Proof of Theorem 10. Let $\mathcal{A}$ be an R-CCRA recognizing f. Let $m = (4r+2)!s!$, where r is the number of registers and s is the number of states of $\mathcal{A}$. By Lemma 20 the sequence $g(n) = f(uw^{m(n+1)}v)$ is an R-generatable EPS. Let q be the exponential polynomial of minimal degree representing g. By Lemma 21, for every k, the sum of k-degree coefficients of this representation is in R. ◀

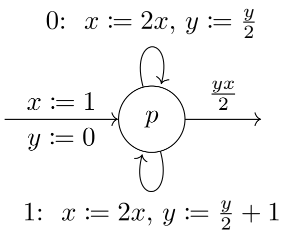


Figure 9 A CCRA recognising a function that, at first glance, may seem unrecognisable by a CCRA (Example 23).

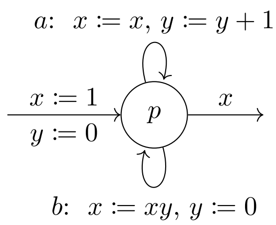


Figure 10 A two-register CRA recognising the function from Example 25.

3.2 CCRA versus R-generatable EPS

We have seen that, for functions $f\colon \Sigma^* \to K$ recognisable by a CCRA, there always exists a finitely generated subsemiring R and $m \geq 1$ such that all sequences of the form $(f(uw^{m(n+1)}v))_n$ are R-generatable EPS. We conjecture that this is not sufficient to characterise functions recognised by CCRA, even if it is already known that the function is recognised by a weighted automaton.

At present, we do not have a counterexample, but we outline a plausible candidate in this subsection. However, it appears difficult to prove that the given function is not recognised by a CCRA.

▶ **Example 22.** Consider the following function $f\colon \{0,1\}^* \to \mathbb{Q}$. Given $w \in \Sigma^*$ let $0 < k_1 < k_2 < \ldots < k_r$ be indices of all 1's, e.g. for $w = 0110$ we have: $r = 2$, $k_1 = 2$, $k_2 = 3$. Then $f(w) = \sum_{i=1}^{r} k_i$.

While it appears unlikely to us that it could be recognised by a CCRA, for any uw^nv, technical calculations show that the output function of this automaton is nevertheless an $\frac{1}{2}\mathbb{Z}$-generatable EPS (see Appendix C).

The reason why functions can or cannot be recognised by a CCRA can be subtle: while it appears that the function f in Example 22 cannot be recognised by a CCRA, the following example shows a function of similar nature that can be recognised by a CCRA.

▶ **Example 23.** We define the function $g\colon \{0,1\}^* \to \mathbb{Q}$. As before given $w \in \Sigma^*$ let $0 < k_1 < k_2 < \ldots < k_r$ be the indices of all 1's. Then $g(w) = \sum_{i=1}^{r} 2^{k_i}$ is recognised by the CCRA in Figure 9.

Another promising example is discussed in Appendix C.

4 Polynomially Ambiguous Weighted Automata

In this section we prove a pumping-like result for polynomially ambiguous weighted automata. Throughout the section, let $\overline{K}$ be an algebraic closure of K.

▶ **Theorem 24.** *If f is recognised by a polynomially ambiguous weighted automaton over K, then there exist a finitely generated multiplicative semigroup $G \subseteq \overline{K}$ and $N \geq 1$ such that*

- *for every $u, w, v \in \Sigma^*$, the characteristic roots of $(f(uw^nv))_n$ are contained in G,*
- *and $\alpha^N \in K$ for all $\alpha \in G$.*

With this theorem, we can give the following example.

▶ **Example 25.** The function $f\colon \{a,b\}^* \to \mathbb{Q}$ given by $f(a^{k_1}ba^{k_2}b\ldots ba^{k_n}) = k_1k_2\cdots k_n$ can be recognised by a CCRA, but not by a polynomially ambiguous weighted automaton.

Proof. The CCRA in Figure 10 recognises f. To see that f cannot be recognised by a polynomially ambiguous weighted automaton, let us consider inputs of the form $(a^kb)^n$. We have $f\big((a^kb)^n\big) = k^n$, meaning that every natural number k appears as characteristic root of an LRS arising from f. However, the monoid $\mathbb{Z}_{>0}$ generates $\mathbb{Q}_{>0}$ as a group, and $\mathbb{Q}_{>0}$ is a countably generated free abelian group (with primes as the generators). Since subgroups of a finitely generated abelian group are finitely generated, but there are infinitely many primes, the natural numbers cannot be a submonoid of a finitely generated abelian group. By Theorem 24, the function f cannot be recognised by a polynomially ambiguous weighted automaton. ◀

We again record this conclusion as a theorem.

▶ **Theorem 26.** *If $|\Sigma| \geq 2$, then there exist functions $f\colon \Sigma^* \to \mathbb{Q}$ that can be recognised by a $\mathbb{Q}$-CCRA but not by a polynomially ambiguous weighted automaton over $\mathbb{Q}$.*

Proof. By Example 25. ◀

The core of the proof of Theorem 24 will be the following lemma. The argument is similar to an argument in [16] and in [23, Prop. 9.3]. For the reader's convenience we put a self-contained proof in Appendix C.

▶ **Lemma 27.** *Let $\mathcal{A} = (d, I, M, F)$ be a trim polynomially ambiguous weighted automaton. Then, for every word $w \in \Sigma^*$, there exists a permutation matrix P such that $P \cdot M(w^{d!}) \cdot P^{-1}$ is upper triangular. Furthermore, all nonzero eigenvalues of $M(w^{d!})$ are products of transition weights (that is, of entries of the matrices $M(a)$ for letters $a \in \Sigma$).*

Before proving Theorem 24, we need a final small observation.

▶ **Lemma 28.** *If $H \subseteq K$ is a finitely generated semigroup and $N \geq 1$, then the semigroup $G = \{\, \alpha \in \overline{K} \mid \alpha^N \in H \cup \{1\} \,\}$ is also finitely generated.*

Proof. Suppose $\beta_1, \ldots, \beta_n$ generate H. For each β_i let $\alpha_i \in \overline{K}$ be a root of $X^N - \beta_i$. Let G' be the subsemigroup of $\overline{K}$ generated by $\alpha_1, \ldots, \alpha_n$ together with the N-th roots of unity in $\overline{K}$ (of which there are at most N, since they are the roots of $X^N - 1$).

We claim $G = G'$. The inclusion $G' \subseteq G$ holds by definition. Suppose $\gamma \in G$. Then $\gamma^N = \beta_1^{k_1} \cdots \beta_n^{k_n}$ for some $k_i \geq 0$. Define $\gamma' \coloneqq \alpha_1^{k_1} \cdots \alpha_n^{k_n} \in G'$. Then $(\gamma')^N = \gamma^N$. It follows that $\gamma = \gamma'\zeta$ with ζ an N-th root of unity (whether or not $\gamma = 0$). So $\gamma \in G'$. ◀

Proof of Theorem 24. Let $\mathcal{A} = (d, I, M, F)$ be a polynomially ambiguous weighted automaton recognising f. Let $N \coloneqq d!$. Without restriction, we can take $\mathcal{A}$ to be trim. Let $H \subseteq K$ be the subsemigroup of K generated by all the finitely many transition weights of $\mathcal{A}$, and let $G = \{\, \alpha \in \overline{K} \mid \alpha^N \in H \cup \{1\} \,\}$. By Lemma 28, the semigroup G is finitely generated.

The characteristic roots of the LRS $(\mathcal{A}(uw^nv))_n$ are eigenvalues of $M(w)$. By Lemma 27, the eigenvalues of $M(w^N)$ are products of transition weights. We have $M(w^N) = M(w)^N$, and so the eigenvalues of $M(w)$ are roots of degree N of products of transition weights of $\mathcal{A}$. This means they belong to G. ◀

We (ambitiously) conjecture the following converse of Theorem 24.

▶ **Conjecture 29.** *Let $f\colon \Sigma^* \to K$ be recognised by a weighted automaton. If there exists a finitely generated multiplicative subsemigroup $G \subseteq \overline{K}$ and $N \in \mathbb{Z}_{\geq 1}$ such that*

- *for every u, w, $v \in \Sigma^*$, the characteristic roots of $(f(uw^n v))_n$ are contained in G,*
- *and $\alpha^N \in K$ for all $\alpha \in G$,*

then f is recognised by a polynomially ambiguous weighted automaton.

Conjecture 29 postulates a pumping-style characterisation. The following conjecture postulates a "global" characterisation, with a similar restriction as in Conjecture 29 imposed on the eigenvalues of the matrix semigroup. Here it is important that the condition is imposed on *all* matrices, not just on the generators.

▶ **Conjecture 30.** *Let $f\colon \Sigma^* \to K$ be recognised by a weighted automaton. If there exists a finitely generated multiplicative subsemigroup $G \subseteq \overline{K}$ and $N \geq 1$ such that*

- *all eigenvalues of matrices $M(w)$ for $w \in \Sigma^*$ are contained in G,*
- *and $\alpha^N \in K$ for all $\alpha \in G$,*

then f is recognised by a polynomially ambiguous weighted automaton.

A positive resolution of the conjectures would extend a characterisation in similar spirit of functions that can be recognised by unambiguous weighted automata [4]. While the conjectures seem ambitious, in the preprint [23], Conjecture 30 was already proved in the case that all transition matrices are invertible.

The following lemma shows that Conjectures 29 and 30 are in fact equivalent.

▶ **Lemma 31.** *Let $\mathcal{A} = (d, I, M, F)$ be a minimal weighted automaton and let $w \in \Sigma^*$. Then the set of nonzero eigenvalues of $M(w)$ is precisely the set of all nonzero characteristic roots of the LRS $(\mathcal{A}(uw^n v))_n$ as u, $v \in \Sigma^*$ range through all words.*

Proof. One direction is obvious — characteristic roots come from eigenvalues of the matrix $M(w)$. We only have to show that every nonzero eigenvalue $\lambda \in \overline{K}$ of $M(w)$ shows up as characteristic root of some LRS.

Working over $\overline{K}$ we can assume that $K = \overline{K}$ is algebraically closed. This allows us to change to a basis in which $M(w)$ is in the Jordan normal form. In particular, we can assume that $M(w)$ is upper triangular and $M(w)[1,1] = \lambda$. Let $e_1 = (1, 0, \ldots, 0) \in K^{d\times 1}$ and let $e_1^\intercal$ be its transpose. Then $\lambda^n = e_1^T M(w^n) e_1$.

Because $\mathcal{A}$ is minimal, the reachability set $\{\, I^\intercal M(w) \mid w \in \Sigma^* \,\}$ spans $K^{1\times d}$ as a vector space, and analogously the coreachability set spans $K^{d\times 1}$. Therefore, there exist α_i, $\beta_j \in K$ and u_i, $v_j \in \Sigma^*$ such that $e_1^\intercal = \sum_{i=1}^d \alpha_i I^\intercal M(u_i)$ and $e_1 = \sum_{j=1}^d M(v_j) F \beta_j$. Now

$$\lambda^n = \sum_{i=1}^d \sum_{j=1}^d \alpha_i \beta_j \, I^\intercal M(u_i w^n v_j) F,$$

expresses the LRS $(\lambda^n)_n$ as linear combination of LRS $(I^\intercal M(u_i w^n v_j)F)_n$. Since the former has a characteristic root λ, a summand must have λ as a characteristic root as well: this is easily seen by considering the LRS as rational functions, and recalling that nonzero characteristic roots correspond to reciprocals of poles, or by the uniqueness result in Theorem 33. ◀

While the main theorem of this section provides a necessary pumping criterion for polynomially ambigualisable automata (that may eventually lead to a full characterisation), another related open problem is to relate the minimal degree of the polynomial bounding the ambiguity to arithmetic properties of the output (in other words, to characterise linearly ambiguous, quadratically ambiguous, etc.). At least in characteristic zero, a tempting idea is to look at the degrees of polynomials arising in the generated exponential polynomials. Indeed, it is easy to see that if the ambiguity of the automaton is bounded by a polynomial of degree d, then no polynomial of higher degree can appear in the generated exponential polynomials. The converse however does not hold, as the following example shows.

▶ **Example 32.** The function $f\colon \{0,1\}^* \to \mathbb{Q}$, mapping a binary word to the natural number it represents (say, LSB on the left), is easily seen to be recognisable by a weighted automaton. We check that, for any u, w, $v \in \Sigma^*$, the exponential polynomial representation of $(\mathcal{A}(uw^nv))_n$ only contains constant polynomials. Indeed, let $u = u_1u_2 \dots u_r$, $w = w_1w_2 \dots w_t$, $v = v_1v_2 \dots v_l$. We have

$$\begin{aligned} f(uw^nv) &= 2^0u_1 + 2^1u_2 + \cdots + 2^{r-1}u_r + (2^rw_1 + 2^{r+1}w_2 + \cdots + 2^{r+t-1}w_t)(1 + 2^t + \cdots + 2^{t(n-1)}) \\ &+ 2^{nt+r}v_1 + \cdots + 2^{nt+r+l-1}v_l = \alpha + \beta\sum_{i=0}^{n-1} 2^{ti} = \alpha + \beta\frac{2^{tn}-1}{2^t-1} \quad (\alpha, \beta \in \mathbb{Q}). \end{aligned}$$

This gives us an exponential polynomial with only constant polynomials.

A set of the form $\{\, g_1 + \cdots + g_m \mid m \le M, g_i \in G \,\}$ for some $M \ge 0$ and a finitely generated subgroup $G \le \mathbb{Q}^\times$ is called a Bézivin set [23]. One can show that $\mathbb{N}$ is not a Bézivin set.[1] It is also easy to see that the output set of a finitely ambiguous weighted automaton is a Bézivin set. Since $f(\Sigma^*) = \mathbb{N}$, the function f cannot be recognised by a finitely ambiguous weighted automaton.

References

1 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. IEEE Computer Society, 2013. doi:10.1109/LICS.2013.65.

2 Corentin Barloy, Nathanaël Fijalkow, Nathan Lhote, and Filip Mazowiecki. A robust class of linear recurrence sequences. *Inf. Comput.*, 289(Part):104964, 2022. doi:10.1016/J.IC.2022.104964.

3 Jason Bell and Daniel Smertnig. Noncommutative rational Pólya series. *Selecta Math. (N.S.)*, 27(3):Paper No. 34, 34, 2021. doi:10.1007/s00029-021-00629-2.

4 Jason P. Bell and Daniel Smertnig. Computing the linear hull: Deciding deterministic? and unambiguous? for weighted automata over fields. In *38th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2023, Boston, MA, USA, June 26-29, 2023*, pages 1–13. IEEE, 2023. doi:10.1109/LICS56636.2023.10175691.

5 Yahia Idriss Benalioua, Nathan Lhote, and Pierre-Alain Reynier. Minimizing Cost Register Automata over a Field. In Rastislav Královic and Antonín Kucera, editors, *49th International Symposium on Mathematical Foundations of Computer Science, MFCS 2024, August 26-30, 2024, Bratislava, Slovakia*, volume 306 of *LIPIcs*, pages 23:1–23:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPICS.MFCS.2024.23.

6 Jean Berstel and Christophe Reutenauer. *Noncommutative Rational Series with Applications*, volume 137 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2011.

7 Michaël Cadilhac, Filip Mazowiecki, Charles Paperman, Michal Pilipczuk, and Géraud Sénizergues. On Polynomial Recursive Sequences. *Theory Comput. Syst.*, 68(4):593–614, 2024. doi:10.1007/S00224-021-10046-9.

8 Agnishom Chattopadhyay, Filip Mazowiecki, Anca Muscholl, and Cristian Riveros. Pumping lemmas for weighted automata. *Log. Methods Comput. Sci.*, 17(3), 2021. doi:10.46298/LMCS-17(3:7)2021.

[1] Among all the denominators of elements of G only finitely many primes appear. The same is then true for the elements of a Bézivin set.

9 Wojciech Czerwinski, Engel Lefaucheux, Filip Mazowiecki, David Purser, and Markus A. Whiteland. The boundedness and zero isolation problems for weighted automata over non-negative rationals. In Christel Baier and Dana Fisman, editors, *LICS '22: 37th Annual ACM/IEEE Symposium on Logic in Computer Science, Haifa, Israel, August 2 - 5, 2022*, pages 15:1–15:13. ACM, 2022. doi:10.1145/3531130.3533336.

10 Laure Daviaud, Marcin Jurdzinski, Ranko Lazic, Filip Mazowiecki, Guillermo A. Pérez, and James Worrell. When are emptiness and containment decidable for probabilistic automata? *J. Comput. Syst. Sci.*, 119:78–96, 2021. doi:10.1016/J.JCSS.2021.01.006.

11 Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. Springer Science & Business Media, 2009.

12 Graham Everest, Alfred Jacobus Van Der Poorten, Igor Shparlinski, Thomas Ward, et al. *Recurrence Sequences*, volume 104. American Mathematical Society Providence, RI, 2003.

13 Nathanaël Fijalkow, Cristian Riveros, and James Worrell. Probabilistic automata of bounded ambiguity. *Inf. Comput.*, 282:104648, 2022. doi:10.1016/J.IC.2020.104648.

14 Vesa Halava, Tero Harju, Mika Hirvensalo, and Juhani Karhumäki. Skolem's Problem — On the Border between Decidability and Undecidability. *TUCS Technical Reports*, 683, 2005.

15 Kosaburo Hashiguchi. Algorithms for Determining Relative Star Height and Star Height. *Inf. Comput.*, 78(2):124–169, 1988. doi:10.1016/0890-5401(88)90033-8.

16 Ismaël Jecker, Filip Mazowiecki, and David Purser. Determinisation and Unambiguisation of Polynomially-Ambiguous Rational Weighted Automata. In Pawel Sobocinski, Ugo Dal Lago, and Javier Esparza, editors, *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2024, Tallinn, Estonia, July 8-11, 2024*, pages 46:1–46:13. ACM, 2024. doi:10.1145/3661814.3662073.

17 Peter Kostolányi. Polynomially Ambiguous Unary Weighted Automata over Fields. *Theory Comput. Syst.*, 67(2):291–309, 2023. doi:10.1007/S00224-022-10107-7.

18 Filip Mazowiecki and Cristian Riveros. Maximal Partition Logic: Towards a Logical Characterization of Copyless Cost Register Automata. In Stephan Kreutzer, editor, *24th EACSL Annual Conference on Computer Science Logic, CSL 2015, September 7-10, 2015, Berlin, Germany*, volume 41 of *LIPIcs*, pages 144–159. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICS.CSL.2015.144.

19 Filip Mazowiecki and Cristian Riveros. Copyless cost-register automata: Structure, expressiveness, and closure properties. *J. Comput. Syst. Sci.*, 100:1–29, 2019. doi:10.1016/J.JCSS.2018.07.002.

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

21 Christos H. Papadimitriou. Computational complexity. In *Encyclopedia of Computer Science*, pages 260–265. 2003.

22 Azaria Paz. *Introduction to Probabilistic Automata*. Academic Press, 1971.

23 Antoni Puch and Daniel Smertnig. Factoring through monomial representations: Arithmetic characterizations and ambiguity of weighted automata. 2024. Preprint. arXiv:2410.03444.

24 Walter L. Ruzzo. On Uniform Circuit Complexity. *J. Comput. Syst. Sci.*, 22(3):365–383, 1981. doi:10.1016/0022-0000(81)90038-6.

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

26 Wen-Guey Tzeng. On Path Equivalence of Nondeterministic Finite Automata. *Inf. Process. Lett.*, 58(1):43–46, 1996. doi:10.1016/0020-0190(96)00039-7.

A Exponential Polynomials and LRS

In this appendix we discuss exponential polynomials, exponential polynomial sequences and their relation to rational functions and LRS in somewhat more detail. The material and treatment are essentially standard, see also [6, Chapter 6][12][14, Proposition 2.11], but a few points are somewhat subtle in positive characteristic.

Every EPS satisfies a linear recurrence and is hence an LRS. In the converse direction, in characteristic 0, if $(a_n)_n$ is an LRS, then there exists some n_0 such that a_n for $n \geq n_0$ coincides with a (unique) EPS. So, in characteristic 0, there is a close relation between LRS and EPS.

In characteristic $p > 0$, it is no longer true that every LRS coincides with an EPS (even for large enough n), because the map $n \mapsto \overline{\binom{n+j-1}{j-1}}$ for $j > p$ does not factor through a function $\mathbb{F}_p \to \mathbb{F}_p$, and hence cannot be represented by a polynomial $q \in \mathbb{F}_p[x]$. However, if an LRS can be expressed as an EPS, the EPS is still unique, with the caveat that the EPS has multiple representations by exponential polynomials, leading us to consider the minimal degree representation, that is, that of degree at most $p-1$ (which is unique). We now discuss this in detail.

Throughout the section, let K be a field and $\overline{K}$ its algebraic closure. If $K = \mathbb{Q}$, this is the field of all algebraic numbers $\overline{\mathbb{Q}}$, a subfield of the complex numbers. The characteristic of K is either 0 or a prime number p. If it is 0, then $\mathbb{Q}$ embeds uniquely into K as prime field, and we can assume $\mathbb{Q} \subseteq K$. Similarly, if $p > 0$, then $\mathbb{F}_p \subseteq K$.

A sequence $(a_n)_n$ is an LRS if and only if the (formal) generating series $F = \sum_{n=0}^{\infty} a_n x^n \in K[\![x]\!]$ is a rational function. Using the existence and uniqueness of partial fraction decompositions of rational functions over the algebraically closed field $\overline{K}$, one obtains the following well-known result.

▶ **Theorem 33.** *Let $(a_n)_n$ be an LRS over K. Then there exist $l \geq 0$, pairwise distinct $\lambda_1, \ldots, \lambda_l \in \overline{K}^{\times}$, and for each $1 \leq i \leq l$ natural numbers $k_i \geq 0$ and coefficients $\alpha_{i,1}, \ldots, \alpha_{i,k_i} \in \overline{K}$ with $\alpha_{i,k_i} \neq 0$ such that*

$$a_n = \sum_{i=1}^{l} \sum_{j=1}^{k_i} \alpha_{i,j} \binom{n+j-1}{j-1} \lambda_i^n \qquad \text{for all sufficiently large } n. \tag{1}$$

The elements λ_i and $\alpha_{i,j}$ are uniquely determined by the sequence $(a_n)_n$. The set $\{\lambda_1, \ldots, \lambda_l\}$ is the set of nonzero characteristic roots of $(a_n)_n$.

We also recall (but do not need) that Equation (1) holds for *all* $n \geq 0$, that is, not only for sufficiently large n, if and only if all characteristic roots of $(a_n)_n$ are nonzero. A converse to Theorem 33 also holds: every sequence expressed as in Equation (1) is an LRS over the algebraically closed $\overline{K}$.

▶ **Example 34.** The sequence $(a_n)_n = n$, whose only characteristic root is 1, can be represented as $a_n = n \cdot 1^n = \left(\binom{n+1}{1} - \binom{n}{0}\right) \cdot 1^n$. For the Fibonacci sequence one obtains the well-known representation $F_n = \frac{1}{\sqrt{5}}\varphi^n - \frac{1}{\sqrt{5}}\psi^n$. Note that $\varphi, \psi \in \overline{\mathbb{Q}} \setminus \mathbb{Q}$. While the Fibonacci sequence cannot be recognised by a polynomially ambiguous weighted automaton with weights in $\mathbb{Q}$, this formula shows that there is such an automaton with weights in the quadratic field $\mathbb{Q}(\varphi)$.

Before proving Theorem 33, we recall one more lemma (and in particular, that it also holds in positive characteristic).

▶ **Lemma 35.** *For all $\alpha \in K$,*

$$\frac{1}{(1-\alpha x)^k} = \sum_{n=0}^{\infty} \binom{n+k-1}{k-1} \alpha^n x^n \in K[\![x]\!].$$

Proof. First suppose $\alpha = 1$. For $K = \mathbb{Z}$, this is well-known and easily derived from the geometric series $(1-x)^{-1} = \sum_{n=0}^{\infty} x^n$ by formal differentiation (or a combinatorial argument). The ring homomorphism $\mathbb{Z} \to K$, $1 \mapsto 1_K$ extends coefficient-wise to a ring homomorphism $\mathbb{Z}[\![x]\!] \to K[\![x]\!]$. Applying the homomorphism to the identity

$$(1-x)^k \sum_{n=0}^{\infty} \binom{n+k-1}{k-1} x^n = 1,$$

and dividing by $(1_K - x)^k$ in $K[\![x]\!]$ proves the claim for $\alpha = 1$. For arbitrary $\alpha \in K$, it follows by substituting αx for x. ◀

Proof of Theorem 33. We may without restriction assume that K is algebraically closed. Because $(a_n)_n$ satisfies an LRS, its generating function $F(x) = \sum_{n=0}^{\infty} a_n x^n \in K[\![x]\!]$ is rational. Thus, there exist coprime polynomials p, $q \in K[x]$ such that $F = p/q$ with $q = (x - \lambda_1^{-1})^{k_1} \cdots (x - \lambda_l^{-1})^{k_l}$, where $\lambda_1, \ldots, \lambda_r \in \overline{K}^\times$ are pairwise distinct and $k_i \geq 1$. By partial fraction decomposition, there are uniquely determined $\alpha_{i,j} \in \overline{K}$ and a uniquely determined polynomial $r \in K[x]$ such that

$$F = r + \sum_{i=1}^{l} \sum_{j=1}^{k_i} \frac{\alpha_{i,j}}{(1-\lambda_i x)^j} = r + \sum_{i=1}^{l} \sum_{j=1}^{k_i} \alpha_{i,j} \sum_{n=0}^{\infty} \binom{n+j-1}{j-1} \lambda_i^n x^n,$$

which shows existence of the claimed representation for all $n > \deg(r)$. The uniqueness follows from the uniqueness of the partial fraction decomposition. ◀

To go further than Theorem 33, we need to distinguish according to the characteristic of K. If $\operatorname{char} K = 0$, then $\binom{n+j-1}{j-1} = \frac{(n+1)\cdots(n+j-1)}{j!}$ allows us to view the binomial coefficients in Equation (1) as polynomial functions in n. Expanding shows that, for sufficiently large n, the sequence $(a_n)_n$ can be represented by a (uniquely determined) exponential polynomial [6, Ch. 6.2].

If $\operatorname{char} K = p > 0$, then there neither needs to exist a representation by an exponential polynomial, nor need this representation be unique if it exists.

▶ **Example 36.** The triangular numbers $T_n \coloneqq \binom{n+1}{2} = \sum_{k=1}^{n} k$ satisfy the linear recurrence relation $T_{n+3} = 3T_{n+2} - 3T_{n+1} + T_n$ with $T_0 = 0$, $T_1 = 1$, $T_2 = 3$. The characteristic polynomial is $x^3 - 3x^2 + 3x - 1 = (x-1)^3$. In the normal form of Theorem 33,

$$T_n = \binom{n+1}{2} \cdot 1^n = \left(\binom{n+2}{2} - \binom{n+1}{1} \right) \cdot 1^n$$

In particular, in characteristic 0, the sequence T_n, whose elements are 0, 1, 3, 6, 10, 15, 21, 28, …, can be expressed using a polynomial, as $T_n = \frac{n^2+n}{2}$.

Now consider $K = \mathbb{F}_2$. Reducing modulo 2, the sequence $\overline{T_n} \in \mathbb{F}_2$ still satisfies the same linear recurrence relation and $\overline{T_n} = \overline{\frac{n^2+n}{2}} = \overline{0}, \overline{1}, \overline{1}, \overline{0}, \overline{0}, \overline{1}, \overline{1}, \overline{0}$, etc. However, this sequence is not induced from a polynomial function $\mathbb{F}_2 \to \mathbb{F}_2$: indeed $(\overline{T_n})_n$ is not 2-periodic. But, if there were a function $f \colon \mathbb{F}_2 \to \mathbb{F}_2$ such that $\overline{T_n} = f(\overline{n})$, then $\overline{T_n}$ would have to be 2-periodic. We see that, in positive characteristic, not every LRS can be represented by an exponential polynomial (even for large enough n).

As the issues of uniqueness of exponential polynomial representations in positive characteristic are relevant in the present paper, we now discuss them in more detail.

A *polynomial* $q = q(x) \in K[x]$ is a formal expression $q = \sum_{i=0}^{k} \alpha_i x^k$ with $\alpha_i \in K$. Polynomials are multiplied and added according to the usual rules, by K-linear extension of

$x^k \cdot x^l = x^{k+l}$. Every polynomial q induces a function $\overline{q}\colon K \to K$, $\lambda \mapsto q(\lambda) = \sum_{i=0}^{k} \alpha_i \lambda^k$, and every function of such a form is called a *polynomial function*. If the field K is infinite (in particular if $\operatorname{char} K = 0$), then $q = q'$ for polynomials q, $q' \in K[x]$ if and only if $\overline{q} = \overline{q'}$, that is, if and only if $q(\lambda) = q'(\lambda)$ for all $\lambda \in K$. In this case, there is no need to carefully distinguish polynomials from polynomial functions.

▶ **Example 37.** If K is finite, then different polynomials may induce the same polynomial functions. Indeed, there are infinitely many polynomials but only finitely many functions $K \to K$. For instance, if $K = \mathbb{F}_p$, then $x^p + 1 \neq x + 1$ as polynomials, but $\lambda^p + 1 = \lambda + 1$ for all $\lambda \in \mathbb{F}_p$, so these two polynomials induce the same function $\mathbb{F}_p \to \mathbb{F}_p$.

As the previous example reminds us, in positive characteristic, we need to carefully distinguish between polynomials and polynomial functions. In particular, for polynomial functions there is no canonical notion of the i-th coefficient, as the example shows.

In any characteristic, since there is a unique ring homomorphism $\mathbb{Z} \to K$, $n \mapsto n \cdot 1_K$, it also makes sense to evaluate polynomials at integers, and we can think of $q \in K[x]$ as inducing a sequence $\overline{q}\colon \mathbb{N} \to K$, $n \mapsto q(n \cdot 1_K)$ (by slight abuse of notation, overloading the notation $\overline{q}$).

We now extend these considerations from polynomials to exponential polynomials. An *exponential polynomial* is a formal expression $q(x) = \sum_{i=1}^{k} q_i(x)\lambda_i^x$ with $q_i \in K[x]$ and pairwise distinct $\lambda_i \in K^\times$. The λ_i for which $q_i \neq 0$ are the *exponential bases* of q. The *degree* of q is $\deg(q) \coloneqq \max\{\, \deg(q_i) \mid 1 \le i \le k \,\}$. Exponential polynomials are again added and multiplied in the usual way, by K-linearly extending $(x^m \lambda_i^x) \cdot (x^n \lambda_j^x) = x^{m+n}(\lambda_i\lambda_j)^x$.[2] An exponential polynomial q induces a sequence $\overline{q}\colon \mathbb{N} \to K$, defined by $\overline{q}(n) = \sum_{i=1}^{k} q_i(n)\lambda_i^n$. We call any sequence arising in such a way an *exponential polynomial sequence* (*EPS*).

The question to what degree the induced sequence determines the polynomials q_i and the exponential bases λ_i is answered by the following lemma.

▶ **Lemma 38.** *Let $p_1, \dots, p_k, q_1, \dots, q_l \in K[x]$. Let $\lambda_1, \dots, \lambda_k \in K^\times$ be pairwise distinct, and let similarly $\mu_1, \dots, \mu_l \in K^\times$ be pairwise distinct. Suppose*

$$\sum_{i=1}^{k} p_i(n)\lambda_i^n = \sum_{j=1}^{l} q_j(n)\mu_j^n \qquad \text{for all } n \in \mathbb{N}.$$

Assume also that, for each i and j we have $p_i(\mathbb{N}) \neq \{0\}$ and $q_j(\mathbb{N}) \neq \{0\}$. Then, up to re-indexing, we have $k = l$, $\lambda_i = \mu_i$ for all $1 \le i \le k$, and $p_i(n) = q_i(n)$ for all $n \in \mathbb{N}$.

Proof. Let $q \in K[x]$. It suffices to show that there exist $\alpha_1, \dots, \alpha_m \in K$ such that $q(n) = \sum_{j=1}^{m} \alpha_j \binom{n+j-1}{j-1}$ for all $n \in \mathbb{Z}$. Then the claim follows from the (stronger) uniqueness statement of Theorem 33.

To show the desired expression for $q(n)$, it in turn suffices to show that $x^j \in \mathbb{Z}[x]$ is a $\mathbb{Z}$-linear combination of polynomials of the form $\binom{x+i-1}{i-1}$. For $j = 0$, this is true because $x^0 = 1 = \binom{x}{0}$. For $j \ge 1$, note that $x^j - j!\binom{x+j}{j} = x^j - (x+1)\cdots(x+j)$ is a polynomial of degree strictly less than j, so the claim follows by induction. ◀

If $\operatorname{char} K = 0$, the uniqueness in the previous lemma implies that $p_i = q_i$ and so an exponential polynomial is uniquely determined by its induced EPS.

[2] This can be made rigorous by considering exponential polynomials as elements of the group algebra $K[x][\Lambda]$ of the group $\Lambda = K^\times$ over the polynomial ring $K[x]$. See [6, Ch. 6].

If char $K = p > 0$, then this is not the case. In this case $p_i(n) = q_i(n)$ for all $n \in \mathbb{N}$ if and only if $p_i(n) = q_i(n)$ for $n \in \{0, \dots, p-1\}$. This is the case if and only if the polynomial $p_i - q_i$ is divisible by $x(x-1)\cdots(x-p+1) = x^p - x$. It follows that, when representing an EPS $(a_n)_n$ using an exponential polynomial $\sum_{i=1}^{k} p_i(x)\lambda_i^x$, we can always find a representation with $\deg(p_i) < p$. In this case, the polynomials p_i are uniquely determined by $(a_n)_n$, and we call the resulting exponential polynomial the *exponential polynomial of minimal degree* representing $(a_n)_n$.

It is important to note that, independent of the characteristic of the field, the exponential bases appearing in a representation as in Theorem 33 or in an exponential polynomial representation (if it exists) are unique. Further, every EPS is an LRS, and the exponential bases of the EPS are precisely the nonzero characteristic roots of the LRS.

▶ **Definition 39.** *If $q = \sum_{i=1}^{m} \sum_{j=0}^{l} \alpha_{i,j} x^j \lambda_i^x$ is an exponential polynomial, then $S_k(q) := \sum_{i=1}^{m} \alpha_{i,k}$ is its* sum of k-degree coefficients.

Thus, in $S_k(q)$, we are summing all coefficients next to some x^k, across all exponential bases. The following straightforward observation on the behaviour of $S_k(q)$ on products will be useful in one of the proofs below.

▶ **Lemma 40.** *If q and q' are exponential polynomials, then $S_k(qq') = \sum_{j=0}^{k} S_j(q) S_{k-j}(q')$.*

Proof. Suppose

$$q = \sum_{\lambda \in K^\times} \sum_{i \geq 0} \alpha_{\lambda,i} x^i \lambda^x \quad \text{and} \quad q' = \sum_{\lambda \in K^\times} \sum_{i \geq 0} \alpha'_{\lambda,i} x^i \lambda^x.$$

(It is notationally convenient to allow formally infinite sums; but in each case there are only finitely many nonzero terms.) Then $S_j(q) = \sum_{\lambda \in K^\times} \alpha_{\lambda,j}$, and analogously for q'. Now

$$qq' = \sum_{\lambda,\mu \in K^\times} \sum_{k \geq 0} \Big(\sum_{j=0}^{k} \alpha_{\mu,j} \alpha'_{\lambda\mu^{-1},k-j} \Big) x^k \lambda^x.$$

This shows

$$\begin{aligned} S_k(qq') &= \sum_{\lambda,\mu \in K^\times} \sum_{j=0}^{k} \alpha_{\mu,j} \alpha'_{\lambda\mu^{-1},k-j} = \sum_{j=0}^{k} \Big(\sum_{\mu \in K^\times} \alpha_{\mu,j} \Big) \Big(\sum_{\mu \in K^\times} \alpha'_{\lambda\mu^{-1},k-j} \Big) \\ &= \sum_{j=0}^{k} S_j(q) S_{k-j}(q'), \end{aligned}$$

where the last step uses $K^\times = \{\, \lambda\mu^{-1} \mid \mu \in K^\times \,\}$. ◀

We will have need to consider sums of k-degree coefficients of EPS $(a_n)_n$ in positive characteristic. Now there are several exponential polynomials representing the EPS, and the sums of k-degree coefficients depend on the particular representation, not just on the sequence $(a_n)_n$.

▶ **Example 41.** Over $K = \mathbb{F}_p$, we have $\overline{n}^p + \overline{n} + \overline{1} = \overline{2}\,\overline{n} + \overline{1}$. However, $q = x^p + x + \overline{1}$ has $S_p(q) = S_1(q) = S_0(q) = \overline{1}$ and $S_k(q) = \overline{0}$ for all other k. By contrast, the polynomial $q' = \overline{2}x + \overline{1}$ has $S_1(q) = \overline{2}$, $S_0(q) = \overline{1}$ and $S_k(q) = \overline{0}$ for all other k.

However, the obstruction in the example, that terms of the form x^p can be replaced by x without changing the induced function, is the only one. More formally, we still have the following.

▶ **Lemma 42.** *Suppose* char $K = p > 0$. *Let q, q' be two exponential polynomials inducing the same EPS, that is, with $q(n) = q'(n)$ for all $n \in \mathbb{N}$. Then $S_0(q) = S_0(q')$ and for every $r \in \{1, \dots, p-1\}$,*

$$\sum_{k\geq 0} S_{k(p-1)+r}(q) = \sum_{k\geq 0} S_{k(p-1)+r}(q').$$

While the sums in the lemma are formally infinite (for notational convenience), they only involve finitely many nonzero terms. Applied to the minimal degree exponential polynomial q representing an EPS $(a_n)_n$, it follows that $S_r(q) = \sum_{k\geq 0} S_{k(p-1)+r}(q')$ for every other exponential polynomial q' representing $(a_n)_n$.

Proof of Lemma 42. Write $q = \sum_{\lambda\in K^\times} q_\lambda \lambda^x$ and $q' = \sum_{\lambda\in K^\times} q'_\lambda \lambda^x$ with polynomials q_λ, q'_λ (only finitely many of which are nonzero). It will suffice to show the claimed formula for all pairs q_λ and q'_λ in place of q and q'.

Fix $\lambda \in K^\times$. Since $q - q'$ vanishes on all of $\mathbb{N}$, by Lemma 38, we must have $(q_\lambda - q'_\lambda)(n) = 0$ for all $n \in \mathbb{N}$. This means that the polynomial $q_\lambda - q'_\lambda$ is divisible by $x(x-1)\cdots(x-p+1) = x^p - x$, that is, there exists a polynomial h such that $q_\lambda - q_{\lambda'} = (x^p - x)h$. In particular $S_0(q_\lambda - q_{\lambda'}) = 0$, and so $S_0(q_\lambda) = S_0(q_{\lambda'})$. For $r \geq 1$, using Lemma 40,

$$\begin{aligned}\sum_{k\geq 0} S_{k(p-1)+r}(q_\lambda - q'_\lambda) &= \sum_{k\geq 0} S_{k(p-1)+r}\big((x^p - x)h\big)\\ &= \sum_{k\geq 1} S_p(x^p - x) S_{k(p-1)+r-p}(h) + \sum_{k\geq 0} S_1(x^p - x) S_{k(p-1)+r-1}(h)\\ &= \sum_{k\geq 1} S_{k(p-1)+r-p}(h) - \sum_{k\geq 0} S_{k(p-1)+r-1}(h) = 0.\end{aligned}$$

So $\sum_{k\geq 0} S_{k(p-1)+r}(q_\lambda) = \sum_{k\geq 0} S_{k(p-1)+r}(q'_\lambda)$. ◀

▶ Remark 43. One more thing can be observed (but will not be needed): Given any LRS $(a_n)_n$ over a field of characteristic $p > 0$, there exists some power p^k such that the subsequences $(a_{np^k+r})_n$ are representable by an EPS (for every r and sufficiently large n). For instance, while the sequence $(\overline{T_n})_n$ in Example 36 is not 2-periodic, it is 4-periodic, and splitting it into four subsequences modulo 4, each subsequence is constant, and hence obviously an EPS.

In general, the periodicity appears because $n \mapsto \overline{\binom{n+j-1}{j-1}}$ is still periodic with period p^k for sufficiently large k. This can be seen as a consequence Lucas's theorem for expressing binomial coefficients modulo p, but is also easy to prove directly.

B Equivalence and Zeroness of CCRA

The two problems are defined as follows (for any classes of automata):

- *equivalence*: given two automata $\mathcal{A}$ and $\mathcal{B}$, decide if $\mathcal{A}(w) = \mathcal{B}(w)$ for all $w \in \Sigma^*$.
- *zeroness*: given an automaton $\mathcal{A}$, decide if $\mathcal{A}(w) = 0$ for all w.

It is folklore that for (polynomially) weighted automata and CCRA the two problems are effectively interreducible. Indeed, to decide zeroness of $\mathcal{A}$ it suffices to check equivalence with $\mathcal{B}$ that outputs 0 on all words. Conversely, to check equivalence of $\mathcal{A}$ and $\mathcal{B}$ one can check zeroness of $\mathcal{A} - \mathcal{B}$, which can be efficiently constructed for these models. Therefore we will deal only with zeroness.

For polynomially ambiguous weighted automata, even unrestricted weighted automata, we know that zeroness is in polynomial time [25] and in NC^2 [26]. Thus, we focus on the complexity of zeroness for CCRA. For the problem to make sense we need to introduce the size of the input CCRA. Given $\mathcal{C} = (Q, q_0, d, \Sigma, \delta, \mu, \nu)$, we say that its size is $|Q| + d + |\Sigma| + \max_p(|p|)$, where p ranges over all polynomials and constants used in δ, μ and ν. We assume that polynomials are represented in the natural succinct form of arithmetic tree circuits, not as a list of all monomials.

Recall that in the update function one cannot use polynomials like x^2 because two copies of x are needed. However, in some sense CCRA can evaluate any polynomial. For example there is a CCRA $\mathcal{C}$ such that $\mathcal{C}(1^n) = n^2$, simply by having two registers storing n and defining the output function as their product. We can say that evaluating x^2 requires two copies of x. We generalise this observation to any polynomial. Given $x_1, \ldots, x_k$, we say that a polynomial $p(x_1, \ldots, x_k)$ is d-copyless if there exists a copyless polynomial $p'(\mathbf{x})$, where $\mathbf{x} = x_{1,1}, \ldots, x_{1,k}, \ldots x_{d,1}, \ldots, x_{d,k}$ (d copies of every x_i) such that $p'(\mathbf{x}) = p(x_1, \ldots, x_k)$, substituting $x_{i,j} = x_j$ for all $1 \leq i \leq d$ and $1 \leq j \leq k$. In particular 1-copyless is copyless.

First we prove a standard and convenient lemma that allows us to turn formulas into polynomials. Note that we assume that formulas, like polynomials, are represented as tree circuits. By the size of the formula, denoted $|\varphi|$, we understand the size of the circuit.

▶ **Lemma 44.** *Let $\mathbf{x} = (x_1, \ldots, x_k)$ and let $\varphi(\mathbf{x})$ be a Boolean quantifier free formula. There exists a polynomial $p(\mathbf{x})$, of size polynomial in $|\varphi|$, such that for every $\mathbf{v} \in \{0,1\}^k$ we have: $p(\mathbf{v}) \in \{0,1\}$; and $p(\mathbf{v}) = 1$ if and only if $\varphi(\mathbf{v})$ evaluates to true. Moreover, the polynomial $p(\mathbf{x})$ is $|\varphi|$-copyless.*

Proof. By induction on the formula size. For the base case $k = 1$ and then given $\varphi = x_1$ we define $p = x_1$. Otherwise, suppose we have polynomials p, p_1 and p_2 corresponding to some formulas φ, φ_1 and φ_2, respectively. We build polynomials as follows.

- For the formula $\neg\varphi$ we define the polynomial $1 - p$.
- For the formula $\varphi_1 \wedge \varphi_2$ we define the polynomial $p_1 \cdot p_2$.

It is easy to see that the construction preserves the properties of the lemma. In particular the base case formulas are trivially 1-copyless; and by induction every subformula ψ is $|\psi|$-copyless. Moreover, every Boolean formula can be build from $\neg$ and $\wedge$, and the final polynomials have polynomial size in the size of the input formula. ◀

▶ **Theorem 45.** *The zeroness problem is PSPACE-complete for CCRA over $\mathbb{Q}$.*

Proof. Regarding the upper bound, by Lemma 5, we know that a CCRA can be translated to a weighted automaton of exponential size. It is known that the equivalence problem for weighted automata is in NC^2 [26]. Since problems in NC^2 can be solved sequentially in polylogarithmic space [24], this essentially yields a PSPACE algorithm. One has to take care that the weighted automaton is not fully precomputed (as it would require too much space). A standard approach computing the states and transitions on the fly solves this issue. See e.g. [16, Section 6.1] for a similar construction.

The rest of the proof is devoted to the matching PSPACE lower bound. We reduce from the validity problem for Quantified Boolean Formulas (QBF), which is known to be PSPACE-complete [21, Theorem 19.1]. One can assume the input is a formula of the form

$$\psi = \forall x_1 \exists y_1 \ldots \forall x_k \exists y_k \;\; \varphi(x_1, y_1, \ldots, x_k, y_k), \tag{2}$$

where φ is quantifier-free. The variables x_i and y_i alternate, x_i are quantified universally and y_i are quantified existentially. For simplicity, we write $\mathbf{x} = (x_1, \ldots, x_k)$ and $\mathbf{y} = (y_1, \ldots, y_k)$. We write $\varphi(\mathbf{x}, \mathbf{y})$ instead of $\varphi(x_1, y_1, \ldots, x_k, y_k)$. Given $\mathbf{v} \in \{0,1\}^{2k}$ we denote by $\varphi(\mathbf{v})$ the truth value of the formula φ with all variables evaluated according to $\mathbf{v}$.

For the reduction, we will need to go through many evaluations of x_i and y_i in a way that respects the quantifiers. It will be convenient to define these evaluations using auxiliary formulas. Let $\mathbf{x}'$ and $\mathbf{y}'$ be fresh copies of variables in $\mathbf{x}$ and $\mathbf{y}$ all of dimension k. We define three quantifier-free formulas: $\textsc{start}(\mathbf{x},\mathbf{y})$, $\textsc{next}(\mathbf{x},\mathbf{y},\mathbf{x}',\mathbf{y}')$ and $\textsc{end}(\mathbf{x},\mathbf{y})$, as follows.

$$\textsc{start}(\mathbf{x},\mathbf{y}) = \bigwedge_{i=1}^{k} \neg x_i, \qquad \textsc{end}(\mathbf{x},\mathbf{y}) = \bigwedge_{i=1}^{k} x_i.$$

Note that START and END do not use $\mathbf{y}$, but in this form it will be easier to state the claim later explaining their purpose. We also define

$$\textsc{next}(\mathbf{x},\mathbf{y},\mathbf{x}',\mathbf{y}') = \bigwedge_{i=1}^{k} \left(\neg x_i \wedge \bigwedge_{j=i+1}^{k} x_j\right) \implies$$
$$\left(x'_i \wedge \bigwedge_{j=i+1}^{k} \neg x'_j \;\wedge\; \bigwedge_{j=1}^{i-1} (x_j \iff x'_j) \wedge (y_j \iff y'_j)\right).$$

To understand the formulas, it is easier to ignore the $\mathbf{y}$ and $\mathbf{y}'$ variables at first. Then these formulas essentially encode a binary counter with k bits: START encodes that all x_i are 0; END encodes that all x_i are 1; and NEXT encodes that $\mathbf{x}'$ is $\mathbf{x}$ increased by 1 in binary. The values of $\mathbf{y}$ can be guessed to anything in START and END. In NEXT we keep consistently the guessed existential values for all unchanged universal variables. The following lemma formally states the purpose of the formulas.

▷ Claim 46. The formula ψ in Equation (2) is valid if and only if there exists a sequence $\mathbf{v}_1,\ldots,\mathbf{v}_n \in \{0,1\}^{2k}$ such that:
1. $\varphi(\mathbf{v}_i)$ is true for all $1 \le i \le n$;
2. $\textsc{start}(\mathbf{v}_1)$ is true;
3. $\textsc{next}(\mathbf{v}_i,\mathbf{v}_{i+1})$ is true for all $1 \le i \le n-1$;
4. $\textsc{end}(\mathbf{v}_n)$ is true.

Proof. The formulas START, NEXT and END are defined in such a way that they go through all possible evaluations of universal variables, guessing consistently the values for existential variables. The first condition guarantees that ψ is valid. ◁

Thanks to Claim 46 we will not need to differentiate between universal and existential variables. In the following we will implicitly use Lemma 44. To avoid additional notation we will write formula names for their corresponding polynomials. Let $\ell = \max\{|\textsc{start}|, |\textsc{end}|, |\textsc{next}|, |\varphi|\}$, then all polynomials corresponding to these formulas are ℓ-copyless. To ease the notation we write

$$\mathbf{z} = (x_1^1,\ldots,x_k^1,y_1^1,\ldots,y_k^1,\ldots,x_1^\ell,\ldots,x_k^\ell,y_1^\ell,\ldots,y_k^\ell)$$

for ℓ identical copies of vectors of variables in $\mathbf{x}$ and $\mathbf{y}$. Note that identical copies occur on indices equal modulo $2k$ (this will be useful when defining the transitions). The number of copies will be sufficient to evaluate all polynomials corresponding to the formulas in a copyless manner. To emphasise this, we will write $\textsc{start}(\mathbf{z})$, $\textsc{end}(\mathbf{z})$ and $\textsc{next}(\mathbf{z})$.

We are ready to define the CCRA $\mathcal{C} = (Q, p_0, d, \Sigma, \delta, \mu, \nu)$, where: $Q = \{\, p_i, q_i \mid 0 \le i \le 2k \,\}$; $d = 8\ell k + 1$; $\Sigma = \{0, 1, \#\}$. We denote the $8\ell k+1$ variables as follows: $\mathbf{z}$, $\mathbf{z}'$, $\mathbf{z}''$, $\mathbf{z}^{\mathrm{old}}$ and s. That is: four disjoint copies corresponding to ℓ copies of $\mathbf{x}$, $\mathbf{y}$ and one extra variable

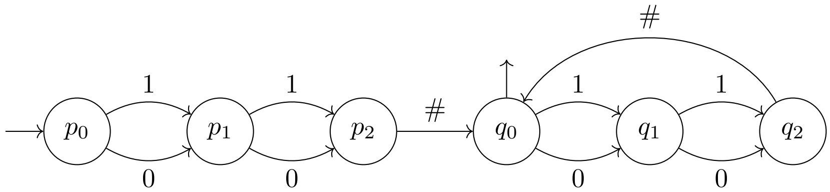


Figure 11 Example for $k = 1$. The state q_0 has an outgoing edge as it is the only one that has a possibly nonzero output.

s. We denote the variables in the copies by z_i, z_i', z_i'', z_i^{old} for $1 \leq i \leq 2\ell k$. The initial function assigns the value 1 to all variables. It will be important that $\mu(s) = 1$; for all other variables the initial value could be arbitrary. The final function is defined by $\nu(x) = 0$ for all $x \in Q \setminus \{q_0\}$ and $\nu(q_0) = s \cdot \textsc{end}(\mathbf{z}^{\mathrm{old}})$.

Before we define the transitions we give an intuition on how the automaton works. We call a subword of length $2k$ a block. The automaton will read a sequence of blocks which correspond to consecutive evaluations $\mathbf{v}_i$ from Claim 46 and store them in multiple copies of $\mathbf{x}$ and $\mathbf{y}$. After reading every block the automaton will check whether: $\textsc{next}$ holds with the previous block; and whether φ holds on the current block. As an invariant, the register s will have value 1 if no error has been detected, and 0 otherwise.

Most of the transitions will initialise some registers. Given a set of variables Z and $b \in \{0, 1\}$, we define the copyless polynomial map $P_{Z,b}$ as: $P_{Z,b}(z) = b$ for $z \in Z$ and $P_{Z,b}(z) = z$ otherwise. In words, the variables in Z are initialised to b and all others keep their previous value. We will use one type of sets Z, defined as follows: $Z_i = \{ z_j, z_j', z_j'' \mid j \equiv i \mod 2k \}$. This will allow us to remember 3ℓ copies at once.

Formally, we define the transitions as follows (see Figure 11 for the shape of the automaton without the register updates):

1. $\delta(p_{i-1}, b) = (p_i, P_{Z_i,b})$ for all $1 \leq i \leq 2k$ and $b \in \{0, 1\}$.
2. $\delta(q_{i-1}, b) = (q_i, P_{Z_i,b})$ for all $1 \leq i \leq 2k$ and $b \in \{0, 1\}$.
3. $\delta(p_{2k}, \#) = (q_0, Q_0)$, where Q_0 resets all variables to 0 except for: $\mathbf{z}^{\mathrm{old}}$ where it puts the content of $\mathbf{z}''$, i.e., $Q_0(z_i^{\mathrm{old}}) = z_i''$ for all $1 \leq i \leq 2\ell k$; and $Q_0(s) = s \cdot \textsc{start}(\mathbf{z}') \cdot \varphi(\mathbf{z})$.
4. $\delta(q_{2k}, \#) = (q_0', R_0)$, where R_0 resets all variables to 0 except for: $\mathbf{z}^{\mathrm{old}}$ where it puts the content of $\mathbf{z}''$, i.e., $Q_0(z_i^{\mathrm{old}}) = z_i''$ for all $1 \leq i \leq 2\ell k$; and $Q_0(s) = s \cdot \textsc{next}(\mathbf{z}^{\mathrm{old}}, \mathbf{z}') \cdot \varphi(\mathbf{z})$.

Note that all polynomials are copyless.

The proof that the reduction works follows essentially from Claim 46. The transitions in Item 1 and Item 2 guess the evaluations $\mathbf{v}_i$. These are stored in three copies: $\mathbf{z}$, $\mathbf{z}'$, $\mathbf{z}''$. The remaining two transitions verify the correctness of these evaluations, i.e., whether they satisfy the conditions in Claim 46. Note that as an invariant these transitions keep in $\mathbf{v}^{\mathrm{old}}$ the previous valuation. In both Item 3, Item 4 we check whether $\varphi(\mathbf{v}_i)$ holds. Additionally, in Item 3 we check whether $\textsc{start}(\mathbf{v}_1)$ is true; and in Item 4 we check whether $\textsc{next}(\mathbf{v}_{i-1}, \mathbf{v}_i)$ is true. All checks are multiplied into the register s, which becomes 0 if any error occurs, and remains 1 otherwise. Finally, the output function guarantees that a nonzero value can be output only if $\textsc{end}(\mathbf{v}_n)$ holds. ◀

C Additional Material

C.1 Proof of Lemma 27

▶ **Lemma 27.** *Let $\mathcal{A} = (d, I, M, F)$ be a trim polynomially ambiguous weighted automaton. Then, for every word $w \in \Sigma^*$, there exists a permutation matrix P such that $P \cdot M(w^{d!}) \cdot P^{-1}$ is upper triangular. Furthermore, all nonzero eigenvalues of $M(w^{d!})$ are products of transition weights (that is, of entries of the matrices $M(a)$ for letters $a \in \Sigma$).*

Proof. The *transition graph* of a word w is the unlabeled directed graph, possibly containing loops, on the vertex set $\{1, ..., d\}$ in which there is an edge $i \to j$ if and only if $M(w)[i, j] \neq 0$. Let us consider the transition graph G corresponding to $M(w)$ and how it relates to the transition graph H corresponding to $M(w^{d!})$. Any edge of H comes from a directed walk of length $d!$ in G.

We first show that the only closed directed walks in H are loops. Indeed, suppose to the contrary that H contains a closed directed cycle of length at least two. Then H contains a directed cycle[3] of length $l \geq 2$ from some vertex i to itself. This cycle arises from a directed walk C from i to itself in G of length $ld!$. In particular, since there exists a directed walk in G from i to itself, there exists a directed cycle D_0 in G that is based at i. The length k of D_0 is of course at most d, and hence divides $d!$. Let $D \coloneqq D_0^{ld!/k}$ denote the $ld!/k$-fold repetition of D_0. Then D and C are two distinct directed walks in G of length $ld!$: after $d!$ steps, the walk D will be at i, but C will not. This leads to a contradiction with polynomial ambiguity: the word $w^{nld!}$ for $n \geq 1$, gives rise to at least 2^n directed walks from i to itself in G, because in each repetition of $w^{ld!}$ we can choose to either follow C or D. Since $\mathcal{A}$ is trim, the state i lies on an accepting run, and hence there exist words u, v such that $u(w^{nld!})v$ has at least 2^n accepting runs.

Now, since the only closed directed walks in H are loops, we can define a total order $\preccurlyeq$ on $\{1, ..., d\}$ such that there is no directed walk in H from i to j if $j \prec i$. Permuting the standard basis vectors correspondingly, which means conjugating the matrix $M(w^{d!})$ by a permutation matrix P, we find that $PM(w^{d!})P^{-1}$ is upper triangular.

In particular, the eigenvalues of $M(w^{d!})$ are precisely the diagonal entries. The i-th diagonal entry is the sum of the weights of all directed walks from i to itself in $\mathcal{A}$ that are labeled by $w^{d!}$. If there were two such walks, that would directly contradict polynomial ambiguity. This means that the diagonal entries of $M(w^{d!})$ come from at most one walk, meaning they are products of transition weights of the automaton. ◀

C.2 Proof of Lemma 21

▶ **Lemma 21.** *If R is a subsemiring, $(a_n)_n$ is an R-generatable EPS and q is the exponential polynomial of minimal degree representing $(a_n)_n$, the sum of k-degree coefficients of q is in R.*

Proof. First notice that all the sequences in Definition 6 have this property. We need to show that the property is preserved under sums and products. Let $(a_n)_n$, $(a'_n)_n$ be R-generatable EPS, and let q, q' be the exponential polynomials of minimal degree representing them, respectively. Then $q + q'$ represents $(a_n + a'_n)_n$ and is of minimal degree (since this just means $\deg(q + q') < p$ if $\operatorname{char} K = p > 0$). Now qq' represents $(a_n a'_n)_n$. By Lemma 40, all the sums of k-degree coefficients of qq' are in R.

[3] A closed directed walk $i_0 \to i_1 \to \cdots \to i_l$ with $i_0 = i_l$ and $i_j \neq i'_j$ unless $\{j, j'\} = \{0, l\}$

If char $K = 0$, then qq' is the unique exponential polynomial representing $(a_n a'_n)_n$, and we are done.

Suppose char $K = p > 0$. Then qq' may not be the minimal degree exponential polynomial representing $(a_n a'_n)_n$ (we only have $\deg(qq') \le 2p-2$, recall Example 12). Let q'' be the exponential polynomial of minimal degree representing $(a_n a'_n)_n$. Now Lemma 42 implies $S_0(q'') = S_0(qq') \in R$ and $S_r(q'') = S_r(qq') + S_{r+p-1}(qq') \in R$ for all $r \in \{1, \dots, p-1\}$. For $k \ge p$, we have $S_k(q'') = 0$. So again, it holds that $S_k(q'') \in R$ for all k. ◀

C.3 Proof in Example 22

We prove that f is an $\frac{1}{2}\mathbb{Z}$-generatable EPS. To see this, let $u = u_1 u_2 \dots u_r$, $w = w_1 w_2 \dots w_t$, $v = v_1 v_2 \dots v_l$. Then (with some constants C, $D \in \mathbb{N}$, independent of n)

$$\begin{aligned} f(uw^n v) &= \sum_{j=1}^{r} j u_j + \sum_{k=0}^{n-1} \sum_{j=1}^{t} (r + kt + j) w_j + \sum_{j=1}^{l} (r + nt + j) v_j \\ &= Cn + D + \sum_{j=1}^{t} \Big(\sum_{k=0}^{n-1} (r + kt + j) \Big) w_j \\ &= Cn + D + \sum_{j=1}^{t} \Big((r+j)n + \frac{(n-1)n}{2} t \Big) w_j. \end{aligned}$$

We see that $n \mapsto f(uw^n v)$ is a polynomial function, with coefficients in $A = \frac{1}{2}\mathbb{Z}$. Hence, the LRS $(f(uw^n v))_n$ is an A-generatable EPS.

C.4 Additional Example in Subsection 3.2

Another promising approach to showing the non-sufficiency of the conditions in Theorem 10, is to show that the class of functions recognised by CCRA is not closed under reversal. For tropical semirings this is known [19].

▶ **Conjecture 47.** *For $|\Sigma| \ge 2$, there exists a function $f \colon \Sigma^* \to \mathbb{Q}$ that is recognisable by a $\mathbb{Q}$-CCRA, but for which the reversal $w \mapsto f(w^r)$, with w^r denoting the reversal of w, is not recognisable by a $\mathbb{Q}$-CCRA.*

Again, there is a promising candidate for which we are currently unable to prove that the reverse is not recognisable.

▶ **Example 48.** Consider the following function $f \colon \{0,1\}^* \to \mathbb{Q}$.

$$\underbrace{0\dots0}_{m_1}\underbrace{1\dots1}_{k_1}\underbrace{0\dots0}_{m_2}\underbrace{1\dots1}_{k_2}\dots\underbrace{0\dots0}_{m_t}\underbrace{1\dots1}_{k_t} \mapsto (\dots((m_1 + k_1)m_2 + k_2)\dots)m_t + k_t$$

It is easy to evaluate $f(w)$ with a CCRA — the bracketing in the formula above can be interpreted as a recipe for doing so. However, there is no obvious way to recognise $f^r(w) \coloneqq f(w^r)$.