


On Rational Recursive Sequences

Lorenzo Clemente  

University of Warsaw, Poland

Maria Donten-Bury  

University of Warsaw, Poland

Filip Mazowiecki 

University of Warsaw, Poland

Michał Pilipczuk 

University of Warsaw, Poland

Abstract

We study the class of rational recursive sequences (ratrec) over the rational numbers. A ratrec sequence is defined via a system of sequences using mutually recursive equations of depth 1, where the next values are computed as rational functions of the previous values. An alternative class is that of simple ratrec sequences, where one uses a single recursive equation, however of depth k : the next value is defined as a rational function of k previous values.

We conjecture that the classes ratrec and simple ratrec coincide. The main contribution of this paper is a proof of a variant of this conjecture where the initial conditions are treated symbolically, using a formal variable per sequence, while the sequences themselves consist of rational functions over those variables. While the initial conjecture does not follow from this variant, we hope that the introduced algebraic techniques may eventually be helpful in resolving the problem.

The class ratrec strictly generalises a well-known class of polynomial recursive sequences (polyrec). These are defined like ratrec, but using polynomial functions instead of rational ones. One can observe that if our conjecture is true and effective, then we can improve the complexities of the zeroness and the equivalence problems for polyrec sequences. Currently, the only known upper bound is Ackermanian, which follows from results on polynomial automata. We complement this observation by proving a PSPACE lower bound for both problems for polyrec. Our lower bound construction also implies that the Skolem problem is PSPACE-hard for the polyrec class.

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

Keywords and phrases recursive sequences, polynomial automata, zeroness problem, equivalence problem

Digital Object Identifier 10.4230/LIPIcs.STACS.2023.24

Related Version *arXiv Version*: <https://arxiv.org/abs/2210.01635>

Funding This work is a result of research conducted within projects no. 2017/26/D/ST6/00201 (Lorenzo Clemente) and 2017/26/E/ST1/00231 (Maria Donten-Bury) financed by National Science Centre, Poland, as well as a part of projects INFSYS (Lorenzo Clemente and Filip Mazowiecki) and BOBR (Michał Pilipczuk) that have received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreements no. 950398 and 948057, respectively).



Acknowledgements We thank Szymon Toruńczyk for helpful discussions.

1 Introduction

The topic of this paper are recursively defined sequences of rational numbers $\mathbb{N} \rightarrow \mathbb{Q}$. There are two natural ways to define such sequences. In a *simple recursion of depth k* one fixes k initial values and defines the next value as a function of the previous k values. This is how the Fibonacci sequence is usually defined (with $k = 2$): $f_0 = 0$, $f_1 = 1$, and $f_{n+2} = f_{n+1} + f_n$. In a *mutual recursion of width k* one defines a system of k sequences such that every sequence



© Lorenzo Clemente, Maria Donten-Bury, Filip Mazowiecki, and Michał Pilipczuk; licensed under Creative Commons License CC-BY 4.0

40th International Symposium on Theoretical Aspects of Computer Science (STACS 2023).

Editors: Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté; Article No. 24; pp. 24:1–24:21



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



has its initial value and the update function can access the immediately previous value of all k sequences, but no older value. For example, we can define $a_n = n^2$ with an extra sequence $b_n = n$ as follows: $a_0 = b_0 = 0$ and $a_{n+1} = a_n + 2b_n + 1$, $b_{n+1} = b_n + 1$. Both styles allow to define various classes of sequences depending on what operations are allowed in the equations, and in general mutual recursion of width k can simulate simple recursion of depth k (by adding sufficiently many auxiliary sequences).

One of the most well-known classes of sequences is the class of *linear recursive sequences*, which is obtained by allowing the update function to use addition and multiplication with constants. These are usually defined with a simple recursion, like in the Fibonacci example, but in fact, as a consequence of the Cayley-Hamilton theorem, one obtains the same class when using mutual recursion [20, Lemma 1.1]. In particular, all the example sequences f_n , a_n and b_n are linear recursive.

Another natural class of sequences are the *polynomial recursive sequences (polyrec)*, which are defined with mutual recursion and updates from the ring of polynomial functions $\mathbb{Q}[x_1, \dots, x_k]$. An example sequence from this class is $c_n = n!$, where one can use the already defined sequence b_n and define $c_0 = 1$ and $c_{n+1} = c_n \cdot (b_n + 1)$. To see the polynomials behind this definition, let x and y be variables corresponding to b_n and c_n , respectively. The polynomial to define b_{n+1} is $P_b(x, y) = x + 1$, and the polynomial to define c_{n+1} is $P_c(x, y) = y(x + 1)$. The class of *simple polynomial recursive sequences* is obtained by using polynomial updates and a simple recursion (instead of mutual recursion) and it is known to be strictly included in the class of all polyrec sequences. In particular, the sequence c_n is polyrec but not simple polyrec [13, Theorem 3.1].

The definition via mutual recursion appears in the area of control theory (under the name *implicit representation* of the space of states), and, in computer science, in the context of *weighted automata* over \mathbb{Q} . Such automata output a rational number for every word over a finite alphabet Σ , and they are defined by linear updates [17]. Linear recursive sequences are thus equivalent to weighted automata with a 1-letter alphabet $\Sigma = \{a\}$ [3]. Similarly, polyrec sequences are equivalent to *polynomial automata* [4] (also known as *cost-register automata* [1]) with a 1-letter alphabet [13].

We are interested in two classical decision problems for such automata. *Equivalence*: Given two automata \mathcal{A} and \mathcal{B} do they output the same number for every word, and *zeroness*: Does the input automaton \mathcal{A} output 0 for every word. These problems are well-known to be efficiently equivalent to each other: Zeroness is clearly a special case of equivalence (just take \mathcal{B} to output zero for every word), and equivalence of \mathcal{A}, \mathcal{B} reduces to zeroness of the difference automaton $\mathcal{A} - \mathcal{B}$ with the expected semantics. Therefore, we will consider only the zeroness problem. From the seminal work of Schützenberger on minimisation of weighted automata it follows that the zeroness problem for weighted automata is in PTIME [32] (in fact even in NC^2 [36]). For polynomial automata over a binary alphabet, zeroness is known to be Ackermann-complete [4]. Using the connection between sequences and automata one immediately obtains NC^2 and Ackermann upper bounds for the zeroness problem of linear recursive sequences, resp., polyrec sequences.

Let us take a closer look at the zeroness problem for recursive sequences, i.e., given a sequence u_n is it the case that $u_n = 0$ for all $n \in \mathbb{N}$? The zeroness problem is a fundamental problem for number sequences. It is a basic building block in computer algebra, e.g., in proving identities involving recursively defined sequences. It is also important from a theoretical point of view as a yardstick of the well-behavedness of classes of number sequences, i.e., interesting classes of sequences should at least have a decidable zeroness problem. The difficulty of solving the zeroness problem in general depends on how the sequence is presented. If the sequence is

defined with a simple recursion of depth k such as $u_{n+k} = f(u_{n+k-1}, \dots, u_n)$, then zeroness trivially reduces to checking that the first k values are 0 and that the recursive update f is well-defined and needs to output 0 when the previous values are 0, i.e., $f(0, \dots, 0) = 0$. However, this simple reasoning is flawed in the case of mutual recursion, because the auxiliary sequences employed in the mutual recursion need not be zero. However, for linear recursive sequences the zeroness problem is easily solved even in the case of mutual recursion, because the reduction to simple recursion [20, Lemma 1.1] implies that a_n is zero if, and only if, its first $k + 1$ values $a_0 = \dots = a_k$ are zero. For polyrec sequences we cannot apply this argument since mutual recursion cannot be simulated by simple recursion in the case of polynomial updates.

Our results. In this paper we introduce the class of *rational recursive sequences* (*ratrec*). This class is defined with mutual recursion and updates from the field of rational functions $\mathbb{Q}(x_1, \dots, x_k)$. For example, the Catalan numbers $C_{n+1} = \frac{2(2n+1)}{n+2}C_n$ can be defined using b_n as an auxiliary sequence. Namely, $C_0 = 1$ and $C_{n+1} = \frac{2(2b_n+1)}{b_n+2}C_n$, where the rational function used to define C_{n+1} is $R(x, y) = \frac{2(2x+1)}{x+2}y$. By definition, the class of polyrec sequences is included in the class of ratrec sequences, and in fact the inclusion is strict as witnessed by the fact that the Catalan numbers C_n are not polynomially recursive [13, Corollary 4.1]. Moreover, ratrec sequences also include the well-known and wide-spread *P-recursive sequences*¹ [22], which according to a 2005 estimate comprise at least 25% of the OEIS archive [31].

A natural question is whether the class of ratrec sequences semantically collapses to the class of *simple rational recursive sequences* obtained by adopting simple recursion. Unlike in the case of polynomial updates, we conjecture that for rational updates we do have such a collapse.

► **Conjecture 1.** *The class of rational recursive sequences coincides with the class of simple rational recursive sequences.*

To see the power of ratrec sequences recall that $c_n = n!$ is not a simple polyrec sequence. However, when in the recursion we allow rational functions, then c_n can be defined with a simple recursion, namely: $c_{n+2} = \frac{(c_{n+1})^2}{c_n} + c_{n+1}$. Thus c_n is simple ratrec.

We introduce a technique towards proving Conjecture 1, which comes from commutative algebra. Instead of looking at the elements of a ratrec sequence as numbers in the field of rationals \mathbb{Q} , we symbolically view them as elements of the field of rational functions $\mathbb{Q}(x_1, \dots, x_k)$. More precisely, we assume that the sequences $\mathbf{F}^{(1)}, \dots, \mathbf{F}^{(k)}$ are initialised by setting $F_0^{(i)} = x_i$ for all $i \in \{1, \dots, k\}$; then, a system of recursive equations governed by rational functions defines further entries of the sequences. Thus, the recursive definition will output elements in $\mathbb{Q}(x_1, \dots, x_k)$ rather than \mathbb{Q} . Intuitively, this corresponds to treating the initial conditions of a system of ratrec sequences symbolically, rather than instantiating them with actual rational values.

Informally speaking, we prove Conjecture 1 for symbolic ratrec sequences, as explained above. Here is a semi-formal statement of our main result, see Theorem 6 for a formalization.

► **Theorem 2.** *The class of rational recursive sequences over $\mathbb{Q}(x_1, \dots, x_k)$, with the system initialised by $F_0^{(i)} = x_i$, coincides with the class of simple rational recursive sequences.*

¹ Sometimes P-recursive sequences are also called *holonomic sequences*, due to a connection with holonomic generating functions.

24:4 On Rational Recursive Sequences

The proof proceeds as follows. From the functions defining the ratrec system we build a sequence of field extensions

$$\mathbb{Q} \subseteq \mathbb{F}_0 \subseteq \mathbb{F}_1 \subseteq \mathbb{F}_2 \subseteq \dots \subseteq \mathbb{Q}(x_1, \dots, x_k)$$

and translate the problem of belonging to the class of simple ratrec sequences to the question of whether this sequence of field extensions eventually stabilises. In order to estimate at which level the stabilisation occurs we use certain results on basic algorithms for rational function fields [25]. We believe that this technique could be extended to prove Conjecture 1, but we also show an example why our current results are not strong enough.

Note that if Conjecture 1 is moreover efficient, it gives a simple algorithm to check zeroness for polyrec. Indeed, since polyrec is a particular case of ratrec, then once a sequence is expressed as a simple ratrec it suffices to check whether the first elements of the sequence are 0. This would improve the Ackermann upper bound inherited from polynomial automata from [4]. This suggests that for polyrec sequences the natural object of study are rational function fields, which are of more algebraic nature and could provide better complexity bounds than the order-theoretic techniques based on sequences of polynomial ideals and Hilbert's finite basis theorem [4].

Our final result is a complexity lower bound for the zeroness problem of polyrec sequences.

► **Theorem 3.** *The zeroness problem for polynomial recursive sequences is PSPACE-hard.*

As far as we know, prior to this work nothing was known about the complexity of zeroness for polyrec sequences, except for the Ackermann upper bound following from polynomial automata [4]. The lower bound is proved by reducing from the QBF validity problem.

Given Conjecture 1 it seems natural to investigate the zeroness problem for ratrec sequences (not just for polyrec sequences). The issue is that it is not clear what would be the input for such a decision problem. Recall that to define ratrec sequences we allow for rational functions in the recursion, which means that we have to deal with division in order to compute the elements of the sequences. Then either one would require that the input sequence comes with a promise that all elements are well-defined and no division by 0 occurs; or one would need to verify whether division by 0 occurs in the input sequence. We find the former solution unnatural, and the latter is at least as hard as the so-called Skolem problem (cf. below), which is not known to be decidable even for linear recursive sequences.

Related work. The zeroness problem has been extensively studied. In the field of automata theory, we can mention applications to the equivalence problem of several classes of automata and grammars, starting from weighted finite automata [32] and polynomial automata [4] already mentioned above, and including context-free grammars [14], multiplicity equivalence of finite automata [36] and multitape finite automata [21, 37], unambiguous context-free grammars [30, Theorem 5.5] (cf. [19, 15] for a PSPACE upper bound), polynomial grammars (which generalise polynomial automata) [8, Chapter 11], deterministic top-down tree-to-string transducers [33], MSO transductions on unordered forests [6, 7], MSO transductions of bounded treewidth under a certain equivalence relation [9], Parikh automata [10], and unambiguous register automata [2]. By replacing (pointwise) multiplication with convolution in the definition of polyrec sequences we obtain the so-called *convolution recursive sequences*, for which the zeroness problem can be solved in PSPACE [15, Theorem 4].

The zeroness problem of D-finite [38] and, more generally, D-algebraic power series [16, 35] is known to be decidable, but its computational complexity has not been investigated. We remark that the class of numerical sequences with D-algebraic power series is incomparable

with ratrec. For instance, the sequence 2^{2^n} is ratrec (in fact, already polyrec) but it cannot be D-algebraic since any D-algebraic sequence has growth rate $n!^{O(1)}$ [24, Ch. 8, Theorem 16]. On the other hand, the exponential generating function of the sequence n^{n-1} counting labelled rooted trees with n nodes is known to be D-algebraic [11]. In other words, the ordinary generating function of $n^{n-1}/n!$ is D-algebraic. If the latter sequence were ratrec, so it would be n^n since $n!$ and n are ratrec and ratrec sequences are closed under Hadamard (i.e., pointwise) product. However n^n it is not ratrec [13, Theorem 5.3].

A natural problem related to the zeroness problem is the so-called *Skolem problem*, which asks whether a given sequence a_n has a zero, i.e., whether for some n we have $a_n = 0$. As a corollary of the constructions used to prove Theorem 3, it follows that the Skolem problem for polyrec sequences is PSPACE-hard. Only NP-hardness was formerly known, and already for linear recursive sequences [5, Corollary 2.1]. Decidability of the Skolem problem for linear recursive sequences is a long-standing open problem (cf. the survey paper [27]). It is interesting to notice that those lower bounds are obtained already on the fixed field with two elements $\{0, 1\}$, and are thus of a combinatorial rather than numerical nature. The Skolem problem for weighted automata over \mathbb{Q} (that generalise linear recursive sequences) is undecidable [29].

2 Preliminaries

By \mathbb{N} we denote the set of nonnegative integers. We denote an arbitrary field by \mathbb{F} , and we use 0 and 1 to denote the zero, resp., one elements thereof. Example fields of interest in this paper are: rationals \mathbb{Q} ; and the two-element field \mathbb{F}_2 . A *sequence* over a *domain* \mathbb{D} is a function $u: \mathbb{N} \rightarrow \mathbb{D}$. The sequences considered in this work are over domains that have a field structure, like rationals \mathbb{Q} . We use bold-face letters as a short-hand for sequences, e.g., $\mathbf{u} = \langle u_n \rangle_{n \in \mathbb{N}}$.

In this paper we work with multivariate polynomials and rational functions. The (*combined*) *degree* of a monomial $x_1^{d_1} \cdots x_k^{d_k}$ is $d_1 + \cdots + d_k$ and the degree of a polynomial $P \in \mathbb{Q}(x_1, \dots, x_k)$, written $\deg P$, is the maximum degree of monomials appearing in it. A *rational function* is a formal fraction of two polynomials, where the denominator is required to be non-zero. The degree of a rational function is the maximum of the degrees of the numerator and the denominator. Recall that for any field \mathbb{F} and a set of variables x_1, \dots, x_n , polynomials over x_1, \dots, x_n form the ring $\mathbb{F}[x_1, \dots, x_n]$, while rational functions over x_1, \dots, x_n form the field $\mathbb{F}(x_1, \dots, x_n)$. We also write $\mathbb{F}[\mathbf{x}]$ and $\mathbb{F}(\mathbf{x})$, where $\mathbf{x} = (x_1, \dots, x_n)$.

The computational aspects of multivariate polynomials, in particular their representation on input to algorithms, are explained in Appendix A, as they will be of no concern in Sections 3 and 4.

3 Rational recursive sequences

We start with the central definitions, which were already discussed in Section 1.

► **Definition 4.** A sequence $\mathbf{u}^{(1)}$ over a field \mathbb{F} is *rationaly recursive* (or *ratrec for short*) of dimension k and degree D if there exist auxiliary sequences $\mathbf{u}^{(2)}, \dots, \mathbf{u}^{(k)}$ over \mathbb{F} and rational functions $P_1, \dots, P_k \in \mathbb{F}(x_1, \dots, x_k)$ of degree at most D such that for all $n \in \mathbb{N}$, we have

$$\begin{cases} u_{n+1}^{(1)} &= P_1(u_n^{(1)}, \dots, u_n^{(k)}), \\ &\vdots \\ u_{n+1}^{(k)} &= P_k(u_n^{(1)}, \dots, u_n^{(k)}). \end{cases} \quad (1)$$

24:6 On Rational Recursive Sequences

A sequence \mathbf{u} over a field \mathbb{F} is polynomially recursive (or polyrec for short) if it satisfies the same definition above, where P_1, \dots, P_k are taken as polynomials in $\mathbb{F}[x_1, \dots, x_k]$. We refer to $(\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)})$ as the system defining $\mathbf{u}^{(1)}$.

In what follows we assume that whenever \mathbf{u} is a ratrec sequence, say defined by a system $(\mathbf{u} = \mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)})$, for all $n \in \mathbb{N}$ all the right hand sides of equations (1) are well-defined, that is, denominators on the right hand side never evaluate to zero.²

For instance, the sequence of *Catalan numbers* $C_n = \frac{1}{n+1} \cdot \binom{2n}{n}$ is ratrec. This can be seen in several ways. For example, they satisfy the recurrence $C_{n+1} = \frac{2(2n+1)}{n+2} \cdot C_n$, giving rise to the following ratrec system:

$$\begin{cases} u_{n+1} &= \frac{2(2v_n+1)}{v_n+2} \cdot u_n, \\ v_{n+1} &= v_n + 1. \end{cases}$$

More generally, any P-recursive sequence a_n is ratrec. A sequence a_n is *P-recursive* [34, Sec. 6.4] if it satisfies a single recursion of the form

$$P_0(n) \cdot a_n + P_1(n) \cdot a_{n+1} + \dots + P_d(n) \cdot a_{n+d} = 0, \quad (2)$$

for every n large enough, where $P_0, \dots, P_d \in \mathbb{Q}[n]$ are polynomials of the index variable n with $P_d \neq 0$. This is essentially transformed into the ratrec system

$$\begin{cases} u_{n+1}^{(d)} &= -\frac{P_0(v_n)}{P_d(v_n)} \cdot u_n^{(0)} - \dots - \frac{P_{d-1}(v_n)}{P_d(v_n)} \cdot u_n^{(d-1)}, \\ u_{n+1}^{(d-1)} &= u_n^{(d)}, \\ &\vdots \\ u_{n+1}^{(0)} &= u_n^{(1)}, \\ v_{n+1} &= v_n + 1, \end{cases} \quad (3)$$

However, (3) works only assuming that $P_d(n) \neq 0$. Since P_d is a fixed polynomial, there exists n_d such that $\forall n \geq n_d P_d(n) \neq 0$. Thus in the final version of (3) we add n_d more sequences in the system to include the first n_d elements of the P-recursive sequence a_n .

Assuming $v_0 = 0$ and $u_0^{(0)} = a_0, \dots, u_0^{(d)} = a_d$, it is immediate to verify $v_n = n$ and $u_n^{(0)} = a_n, \dots, u_n^{(d)} = a_{n+d}$ for every $n \in \mathbb{N}$.

The family of ratrec sequences strictly includes both P-recursive sequences and polyrec sequences. As an example consider the sequence $u_n = 2^{2^n} + C_n$. On the one hand, this sequence is certainly ratrec because it is the sum of a polyrec and a P-recursive sequence (which are ratrec) and ratrec sequences are closed under sum. On the other hand, u_n is not P-recursive since it grows asymptotically faster than any P-recursive sequence (every P-recursive sequence is in $O((n!)^\gamma)$ for some constant $\gamma \in \mathbb{R}$ [23, Proposition 3.11]). Further, recall that 2^{2^n} is polyrec, and C_n is not polyrec [13, Corollary 4.1]. Since polyrec sequences are closed under sum and subtraction, we conclude that u_n is not polyrec.

In [13, Theorem 7.1], the following property of ratrec sequences is proved: if \mathbf{u} is ratrec, then there exists $m \in \mathbb{N}$ and a *cancelling polynomial* $P \in \mathbb{Q}[y_0, \dots, y_m]$, that is, a non-zero polynomial such that

$$P(u_n, u_{n+1}, \dots, u_{n+m}) = 0 \quad \text{for all } n \in \mathbb{N}.$$

² As mentioned in the introduction, checking whether a ratrec sequence is well-defined is at least Skolem hard, which means it is not known to be decidable. Thus for decision problems one should restrict to classes like polyrec, where sequences are always well-defined.

► **Theorem 5** (Theorem 7.1 in [13]). *Every ratrec sequence admits a cancelling polynomial.*

In [13, Theorem 5.3] it is shown that the sequence $u_n = n^n$ has no cancelling polynomial, and hence is not polyrec and not ratrec.

4 Extension degrees

In this section we consider ratrec sequences as in Definition 4 over the field $\mathbb{Q}(\mathbf{x})$. Let $(\mathbf{F}^{(1)}, \dots, \mathbf{F}^{(k)})$ be a system defining $\mathbf{F}^{(1)}$. In this section we will consider sequences with the following fixed initial conditions: $F_0^{(i)} = x_i$. Note that this technical assumption is important, in particular we cannot initialise $F_0^{(i)}$ with elements in \mathbb{Q} . (If we could, this class would generalise ratrec over the field \mathbb{Q} .)

Theorem 6 below formalises Theorem 2 and is the main result of this paper. In essence, we show that a ratrec definition over $\mathbb{Q}(\mathbf{x})$ can be translated to a simple ratrec over $\mathbb{Q}(\mathbf{x})$ with a polynomial recursion depth. We hope that this insight might lead towards a resolution of Conjecture 1.

► **Theorem 6.** *Let $\mathbf{F}^{(1)}$ be a ratrec sequence over the field $\mathbb{Q}(\mathbf{x})$, defined by a system $(\mathbf{F}^{(1)}, \dots, \mathbf{F}^{(k)})$, with the initial conditions: $F_0^{(i)} = x_i$ for $i = 1, \dots, k$. Then there exists a rational function $R \in \mathbb{Q}(y_0, \dots, y_m)$ such that*

$$F_{n+m+1}^{(1)} = R(F_n^{(1)}, F_{n+1}^{(1)}, \dots, F_{n+m}^{(1)}), \text{ for all } n \in \mathbb{N}.$$

Moreover, if $F_n^{(1)}$ is of dimension k and degree D , then m can be bounded from above by $k + k^3 \log(kD)$.

Before we proceed to the proof, let us note that if we write $R(y_0, \dots, y_m) = \frac{A(y_0, \dots, y_m)}{B(y_0, \dots, y_m)}$, where $A, B \in \mathbb{Q}[y_0, \dots, y_m]$, then Theorem 6 shows that the following polynomial is cancelling for $\mathbf{u}^{(1)}$:

$$P(y_0, \dots, y_m, y_{m+1}) = y_{m+1} \cdot B(y_0, \dots, y_m) - A(y_0, \dots, y_m).$$

Thus, Theorem 6 shows (and in fact, is equivalent to) that every ratrec sequence over $\mathbb{Q}(\mathbf{x})$ admits a cancelling polynomial that is linear in the last variable (here y_{m+1}), improving upon Theorem 5.

The remainder of this section is devoted to the proof of Theorem 6 and to a discussion related to it. In particular, the first part of the theorem (existence) will be proved in Section 4.1 and the concrete bound on the depth m will be proved in Section 4.2.

Let us make a few observations about the sequences $F_n^{(1)}, \dots, F_n^{(k)}$. First, a straightforward estimation shows that the degrees of functions $F_n^{(1)}, \dots, F_n^{(k)}$ grow at most single-exponentially in n .

► **Lemma 7.** *For $n \in \mathbb{N}$, let d_n be the maximum degree of $F_n^{(1)}, \dots, F_n^{(k)}$. Then $d_n \leq (k \cdot D)^n$.*

Proof. We proceed by induction on n . Initially we have $d_0 = 1$ by definition. By Definition 4 $F_{n+1}^{(i)}$ is obtained by substituting rational functions $F_n^{(1)}, \dots, F_n^{(k)}$ of degree at most d_n into a rational function P_i of degree at most D . Let $P_i = \frac{A}{B}$ be the ratio of two polynomials $A, B \in \mathbb{Q}[\mathbf{x}]$ of degree at most D . Let $C \in \mathbb{Q}[\mathbf{x}]$ be the least common multiple of all denominators of $F_n^{(1)}, \dots, F_n^{(k)}$, and thus of degree at most $k \cdot d_n$. We can then write $F_n^{(1)} = \frac{G^{(1)}}{C}, \dots, F_n^{(k)} = \frac{G^{(k)}}{C}$, where the numerators $G^{(1)}, \dots, G^{(k)} \in \mathbb{Q}[\mathbf{x}]$ are polynomials of degree also at most $k \cdot d_n$. It follows that both $A(F_n^{(1)}, \dots, F_n^{(k)})$ and $B(F_n^{(1)}, \dots, F_n^{(k)})$

24:8 On Rational Recursive Sequences

can be written as rational functions of the form $\frac{\hat{A}}{C^D}$, resp., $\frac{\hat{B}}{C^D}$, where the numerators are polynomials $\hat{A}, \hat{B} \in \mathbb{Q}[\mathbf{x}]$ of degree at most $D \cdot k \cdot d_n$ and the same holds for the common denominator $C^D \in \mathbb{Q}[\mathbf{x}]$. It follows that $F_{n+1}^{(i)}$ is a rational function of degree $d_{n+1} \leq k \cdot D \cdot d_n$, as required. \blacktriangleleft

The next lemma is a key property implied by the recurrence: if several consecutive elements of the sequence $F_n^{(i)}$ satisfy some algebraic constraint, then this constraint is also satisfied at every step later in the sequence.

► **Lemma 8** (Substitution lemma). *Suppose $Z(y_0, \dots, y_m) \in \mathbb{Q}[y_0, \dots, y_m]$ is a polynomial such that $Z(F_0^{(i)}(\mathbf{x}), \dots, F_m^{(i)}(\mathbf{x})) = 0$. Then $Z(F_n^{(i)}(\mathbf{x}), \dots, F_{n+m}^{(i)}(\mathbf{x})) = 0$ for all $n \in \mathbb{N}$.*

Proof. By assumption we have

$$Z(F_0^{(i)}(\mathbf{x}), \dots, F_m^{(i)}(\mathbf{x})) = 0. \quad (4)$$

Consider the ring homomorphism $h: \mathbb{Q}[\mathbf{x}] \rightarrow \mathbb{Q}(\mathbf{x})$ that maps the variables x_1, \dots, x_k to rational functions $F_1^{(1)}, \dots, F_1^{(k)}$, respectively. For a rational function P/Q such that $h(Q) \neq 0$, by $h(P/Q)$ we understand the rational function $h(P)/h(Q)$. (Note that such an extension of h to $\mathbb{Q}(\mathbf{x})$ does not have to be a field homomorphism.) From the definition of the sequence $F_n^{(i)}$ it readily follows by induction that

$$h(F_n^{(i)}) = F_{n+1}^{(i)}, \quad \text{for all } n \in \mathbb{N}.$$

Thus, by applying h to both sides of (4), we infer that

$$Z(F_1^{(i)}(\mathbf{x}), \dots, F_{m+1}^{(i)}(\mathbf{x})) = 0.$$

We conclude by repeating this reasoning n times. \blacktriangleleft

In the following we introduce some basic terminology about (commutative) fields (cf. [26, Sec. II.1], [12, Sec. V.3], or [18, Sec. 13.1 and 13.2] for more details). Let \mathbb{E}, \mathbb{F} be two fields. When $\mathbb{E} \subseteq \mathbb{F}$ we say that \mathbb{F} is a *field extension* of \mathbb{E} , which is called the *base field*. Given a field extension \mathbb{F} over \mathbb{E} and elements $f_1, \dots, f_n \in \mathbb{F}$, let $\mathbb{E}(f_1, \dots, f_n)$ be the smallest field extension over \mathbb{E} containing f_1, \dots, f_n . If $\mathbb{F} = \mathbb{E}(f_1, \dots, f_n)$, then we say that \mathbb{F} is *finitely generated over \mathbb{E}* (with *generators* f_1, \dots, f_n).

The *degree* of \mathbb{F} over \mathbb{E} , written $\deg_{\mathbb{E}} \mathbb{F}$, is the dimension of \mathbb{F} as a vector space over the base field \mathbb{E} . For instance, $\mathbb{Q}(\sqrt{2})$ has degree 2 over \mathbb{Q} (its elements can be put in the form $a + b \cdot \sqrt{2}$) and $\mathbb{Q}(\sqrt[3]{2})$ has degree 3 (its elements can be put in the form $a + b \cdot \sqrt[3]{2} + c \cdot (\sqrt[3]{2})^2$). Field extensions need not have finite degree. For instance, $\mathbb{Q}(\pi)$ and $\mathbb{Q}(x)$ are two field extensions of \mathbb{Q} of infinite degree. The degree is multiplicative:

► **Lemma 9** (cf. [18, Theorem 14]). *Consider field extensions $\mathbb{E} \subseteq \mathbb{F} \subseteq \mathbb{G}$. Then, $\deg_{\mathbb{E}} \mathbb{G} = \deg_{\mathbb{E}} \mathbb{F} \cdot \deg_{\mathbb{F}} \mathbb{G}$ (even for infinite degrees).*

An element $f \in \mathbb{F}$ is *algebraic* over the base field \mathbb{E} if there is a nonzero polynomial $P(x) \in \mathbb{E}[x]$ s.t. $P(f) = 0$. The field extension \mathbb{F} is *algebraic* over the base field \mathbb{E} if every element in \mathbb{F} is algebraic over \mathbb{E} .

Let \mathbb{F} be a field extension of \mathbb{E} . A subset $\{f_1, \dots, f_n\} \subseteq \mathbb{F}$ of elements of \mathbb{F} is *algebraically independent* over \mathbb{E} if there is no nonzero polynomial $P(x_1, \dots, x_n) \in \mathbb{E}[x_1, \dots, x_n]$ such that $P(f_1, \dots, f_n) = 0$. The *transcendence degree* of \mathbb{F} over \mathbb{E} , denoted $\text{tr deg}_{\mathbb{E}} \mathbb{F}$, is the largest cardinality of a subset of elements of \mathbb{F} which are algebraically independent over \mathbb{E} . Note that \mathbb{F} is algebraic over \mathbb{E} if and only if $\text{tr deg}_{\mathbb{E}} \mathbb{F} = 0$. Like the algebraic degree is multiplicative, the transcendence degree is additive:

► **Lemma 10** (cf. [12, Corollary to Theorem 4, A.5.111]). *Consider field extensions $\mathbb{E} \subseteq \mathbb{F} \subseteq \mathbb{G}$. Then, $\text{tr deg}_{\mathbb{E}} \mathbb{G} = \text{tr deg}_{\mathbb{E}} \mathbb{F} + \text{tr deg}_{\mathbb{F}} \mathbb{G}$.*

In the following we will always take as the base field $\mathbb{E} = \mathbb{Q}$, in which case we will write just $\text{tr deg } \mathbb{F}$ instead of $\text{tr deg}_{\mathbb{Q}} \mathbb{F}$. For example, $\text{tr deg } \mathbb{Q}(\sqrt{2}) = 0$ because $\sqrt{2}$ is an algebraic number over \mathbb{Q} , $\text{tr deg } \mathbb{Q}(\sqrt{2}, \pi) = 1$ because π is a transcendental number, and $\text{tr deg } \mathbb{Q}(x_1, \dots, x_n) = n$.

The motivation to look at field extensions is that a ratrec system naturally defines the following sequence of field extensions

$$\mathbb{Q} \subseteq \mathbb{F}_0 \subseteq \mathbb{F}_1 \subseteq \dots \subseteq \mathbb{Q}(\mathbf{x}), \quad (5)$$

where $\mathbb{F}_0 = \mathbb{Q}(x_1)$ and $\mathbb{F}_{n+1} = \mathbb{F}_n(F_{n+1}^{(1)}(\mathbf{x}))$ for $n \in \mathbb{N}$.

4.1 Ascending sequences of field extensions

In this section we prove the following Noether-like result.

► **Theorem 11.** *Consider any ascending sequence of field extensions of the form*

$$\mathbb{Q} \subseteq \mathbb{F}_0 \subseteq \mathbb{F}_1 \subseteq \dots \subseteq \mathbb{Q}(x_1, \dots, x_k).$$

Then the sequence eventually stabilises: there exists n_0 such that $\mathbb{F}_{n_0} = \mathbb{F}_{n_0+1} = \mathbb{F}_{n_0+2} = \dots$

The crucial reason for the result above is that the number k of variables is fixed. In the proof of Theorem 11 we use the following result on finitely generated extensions.

► **Lemma 12** (cf. [12, A.5.118, Cor. 3]). *If \mathbb{G} is a finitely generated extension over \mathbb{E} , then every subextension $\mathbb{E} \subseteq \mathbb{F} \subseteq \mathbb{G}$ of \mathbb{G} over \mathbb{E} is also finitely generated.*

Proof of Theorem 11. We lead the proof along the lines that will be used also in the proof of Lemma 16 to give bounds for the stabilisation point. First of all, observe that

$$\text{tr deg } \mathbb{F}_n \leq \text{tr deg } \mathbb{Q}(x_1, \dots, x_k) = k, \quad \text{for all } n.$$

Hence, there is n_1 such that $\text{tr deg } \mathbb{F}_{n_1} = \text{tr deg } \mathbb{F}_{n_1+1} = \dots$. Let $\mathbb{F}_{\infty} := \bigcup_{n=0}^{\infty} \mathbb{F}_n$ and consider the ascending sequence

$$\mathbb{F}_{n_1} \subseteq \mathbb{F}_{n_1+1} \subseteq \dots \subseteq \mathbb{F}_{\infty}. \quad (6)$$

We have $\text{tr deg } \mathbb{F}_{n_1} = \text{tr deg } \mathbb{F}_{n_1+i}$ for all $i > 0$, and, by Lemma 10, $\text{tr deg}_{\mathbb{F}_{n_1}} \mathbb{F}_{n_1+i} = 0$, i.e., \mathbb{F}_{n_1+i} is algebraic over \mathbb{F}_{n_1} . Moreover, \mathbb{F}_{∞} is also algebraic over \mathbb{F}_{n_1} because any element of \mathbb{F}_{∞} belongs to some \mathbb{F}_{n_1+i} . Since $\mathbb{Q}(x_1, \dots, x_k) \supseteq \mathbb{F}_{n_1}$ is a finitely generated extension of \mathbb{F}_{n_1} and $\mathbb{F}_{n_1} \subseteq \mathbb{F}_{\infty} \subseteq \mathbb{Q}(x_1, \dots, x_k)$ is a subextension, by Lemma 12 we have that \mathbb{F}_{∞} is also a finitely generated extension of \mathbb{F}_{n_1} . In other words, there are generators $f_1, \dots, f_m \in \mathbb{F}_{\infty}$ such that

$$\mathbb{F}_{\infty} = \mathbb{F}_{n_1}(f_1, \dots, f_m).$$

Since the generators f_1, \dots, f_m are algebraic over \mathbb{F}_{n_1} , \mathbb{F}_{∞} is an algebraic extension of finite degree over \mathbb{F}_{n_1} by Lemma 9. (Concretely, an upper bound for the degree is the product of the degrees of minimal polynomials of the generators f_1, \dots, f_m .) It follows that the sequence in (6) is an ascending sequence of vector subspaces of \mathbb{F}_{∞} , where we treat \mathbb{F}_{∞} as a vector space over \mathbb{F}_{n_1} . Since the dimension of \mathbb{F}_{∞} as a vector space over \mathbb{F}_{n_1} is finite, this sequence must eventually stabilise at \mathbb{F}_{n_0} for some $n_0 \geq n_1$. ◀

24:10 On Rational Recursive Sequences

We now prove the existence part of Theorem 6 using Theorem 11.

Proof (of the first part of Theorem 6). By Theorem 11, the sequence in (5) stabilises at some \mathbb{F}_m , that is,

$$\mathbb{F}_m = \mathbb{F}_{m+1} = \mathbb{F}_m(F_{m+1}^{(1)}(\mathbf{x})).$$

Therefore, we have $F_{m+1}^{(1)}(\mathbf{x}) \in \mathbb{F}_m$. Noting that $\mathbb{F}_m = \mathbb{Q}(F_0^{(1)}(\mathbf{x}), \dots, F_m^{(1)}(\mathbf{x}))$, we see that $F_{m+1}^{(1)}(\mathbf{x})$ can be expressed as a rational function of the generators: there exists a rational function $R \in \mathbb{Q}(y_0, \dots, y_m)$ such that

$$F_{m+1}^{(1)}(\mathbf{x}) = R(F_0^{(1)}(\mathbf{x}), \dots, F_m^{(1)}(\mathbf{x})).$$

We may now apply Lemma 8 to the numerator of the rational function $R(y_0, \dots, y_m) - y_{m+1}$, thus obtaining that

$$F_{n+m+1}^{(1)}(\mathbf{x}) = R(F_n^{(1)}(\mathbf{x}), \dots, F_{n+m}^{(1)}(\mathbf{x})), \quad \text{for every } n \in \mathbb{N}. \quad \blacktriangleleft$$

4.2 Upper bound

We now move to the second, quantitative part of the proof of Theorem 6: we need to prove that m is bounded from above by $k + k^3 \log(kD)$. For this, we inspect the proof of Theorem 11 in the special case of the chain of extensions (5) given by a ratrec system. The first observation is that the sequence of transcendence degrees stabilises very quickly.

► **Lemma 13.** *The transcendence degrees $\text{tr deg } \mathbb{F}_n$ of the sequence (5) stabilise after at most k steps.*

Proof. As argued, $\text{tr deg } \mathbb{F}_n \leq \text{tr deg } \mathbb{Q}(x_1, \dots, x_k) = k$ for all n . The next extension \mathbb{F}_{n+1} is obtained by adding a new rational function $F_{n+1}^{(1)}(x_1, \dots, x_k)$ to the previous extension \mathbb{F}_n . This immediately shows that $\text{tr deg } \mathbb{F}_n \leq \text{tr deg } \mathbb{F}_{n+1} \leq \text{tr deg } \mathbb{F}_n + 1$. We argue that if $\text{tr deg } \mathbb{F}_{d+1} = \text{tr deg } \mathbb{F}_d$ for some d , then the transcendence degree cannot change anymore: $\text{tr deg } \mathbb{F}_d = \text{tr deg } \mathbb{F}_{d+1} = \text{tr deg } \mathbb{F}_{d+2} = \dots$. Note that this will conclude the proof, because then the transcendence degree can increase at most k times before eventually stabilising.

Since $\text{tr deg } \mathbb{F}_{d+1} = \text{tr deg } \mathbb{F}_d$, it follows that $F_{d+1}^{(1)}(\mathbf{x})$ is algebraic over \mathbb{F}_d , which means that it satisfies $P(F_{d+1}^{(1)}(\mathbf{x})) = 0$ for some nonzero polynomial $P(x) \in \mathbb{F}_d[x]$. By clearing out denominators, there is a nonzero polynomial $Z(y_0, \dots, y_{d+1}) \in \mathbb{Q}[y_0, \dots, y_{d+1}]$ such that

$$Z(F_0^{(1)}(\mathbf{x}), \dots, F_{d+1}^{(1)}(\mathbf{x})) = 0. \quad (7)$$

By Lemma 8 we have

$$Z(F_n^{(1)}(\mathbf{x}), \dots, F_{n+d+1}^{(1)}(\mathbf{x})) = 0 \quad \text{for every } n \in \mathbb{N}.$$

This means that $F_{n+d+1}^{(1)}(\mathbf{x})$ is algebraic over \mathbb{F}_{n+d} , implying

$$\text{tr deg } \mathbb{F}_{n+d+1} = \text{tr deg } \mathbb{F}_{n+d}(F_{n+d+1}^{(1)}(\mathbf{x})) = \text{tr deg } \mathbb{F}_{n+d}.$$

This concludes the proof. ◀

Note that even when the transcendence degrees of the fields in (5) stabilise, it may still take several further steps until the fields themselves eventually stabilise. We will later give an example that this may indeed happen.

We are left with estimating the degrees of field extensions after the transcendence degree in the chain (5) stabilises. For this, we use the following two results.

► **Lemma 14** (cf. [25, Lemma 3.4]). *Let $f \in \mathbb{Q}(x_1, \dots, x_k)$ be algebraic over*

$$\mathbb{F}_n = \mathbb{Q}(F_0(x_1, \dots, x_k), \dots, F_n(x_1, \dots, x_k)).$$

Then there is a polynomial $Z(x) \in \mathbb{F}_n[x]$ of degree at most $\deg F_0 \cdots \deg F_n$ s.t. $Z(f) = 0$.

► **Lemma 15** (cf. [26, Exercise III.A.2]). *Let $\mathbb{Q} \subseteq \mathbb{F} \subseteq \mathbb{Q}(x_1, \dots, x_k)$ be a subextension of $\mathbb{Q}(x_1, \dots, x_k)$ over \mathbb{Q} of transcendence degree $r := \text{tr deg}_{\mathbb{Q}} \mathbb{F}$. Then there are (algebraically independent) rational functions $f_1, \dots, f_r \in \mathbb{Q}(x_1, \dots, x_k)$ such that $\mathbb{F} \subseteq \mathbb{Q}(f_1, \dots, f_r)$.*

► **Lemma 16.** *The sequence (5) eventually stabilises after at most $k + k^3 \log(kD)$ steps.*

Proof. By Lemma 13, there exists $j_1 \leq k$ such that

$$r := \text{tr deg } \mathbb{F}_{j_1} = \text{tr deg } \mathbb{F}_j \quad \text{for all } j \geq j_1.$$

In particular, all field extensions \mathbb{F}_j for $j \geq j_1$ are algebraic over \mathbb{F}_{j_1} . As in the proof of Theorem 11, consider the field extension $\mathbb{F}_\infty = \bigcup_{j=0}^{\infty} \mathbb{F}_j$ over \mathbb{F}_{j_1} , which is algebraic.

In particular, $\text{tr deg } \mathbb{F}_\infty = r$. By Lemma 15, there are rational functions $f_1, \dots, f_r \in \mathbb{Q}(x_1, \dots, x_k)$ s.t. $\mathbb{F}_\infty \subseteq \mathbb{Q}(f_1, \dots, f_r)$. Since $\mathbb{Q}(f_1, \dots, f_r)$ has the same transcendence degree as \mathbb{F}_{j_1} , i.e. $\text{tr deg } \mathbb{Q}(f_1, \dots, f_r) = r$, it follows that the f_i 's are algebraic over \mathbb{F}_{j_1} . By Lemma 14, the degree of each $F_{j_1}[f_i]$ is at most $\deg F_{j_1}[f_i] \leq \deg F_0 \cdots \deg F_{j_1}$ over \mathbb{F}_{j_1} . It follows that $\mathbb{Q}(f_1, \dots, f_r)$ is an algebraic extension of degree at most $d = (\deg F_0 \cdots \deg F_{j_1})^r$ over \mathbb{F}_{j_1} . Thus the chain

$$\mathbb{F}_{j_1} \subseteq \mathbb{F}_{j_1+1} \subseteq \mathbb{F}_{j_1+2} \subseteq \cdots$$

is such that the degree of any \mathbb{F}_{j_1+t} over \mathbb{F}_{j_1} is at most d . Recall that all extensions in this chain are algebraic. We show that it stabilises after at most $\log d$ steps. Assume that for some $t \geq 0$ we have $\mathbb{F}_{j_1+t} = \mathbb{F}_{j_1+t+1}$, that is $F_{j_1+t+1}^{(1)}(\mathbf{x}) \in \mathbb{F}_{j_1+t}$. Thus, there is a rational function $R \in \mathbb{F}_{j_1}(y_1, \dots, y_t)$ such that $F_{j_1+t+1}^{(1)}(\mathbf{x}) = R(F_{j_1+1}^{(1)}(\mathbf{x}), \dots, F_{j_1+t}^{(1)}(\mathbf{x}))$. Then by applying Lemma 8 to the numerator of $R(y_1, \dots, y_t) - y_{t+1}$, we may express $F_{j_1+n+t+1}^{(1)}(\mathbf{x})$ as a rational function of $F_{j_1+n+1}^{(1)}(\mathbf{x}), \dots, F_{j_1+n+t}^{(1)}(\mathbf{x})$, i.e., elements of \mathbb{F}_{j_1+n+t} . Hence an equality in the field chain implies stabilisation at this point.

Since the degree grows at each step before stabilisation and the degree is multiplicative, the chain stabilises after at most $\log d$ steps. Indeed, in every step before stabilisation the degree is multiplied by at least two. Thus the chain stabilises at \mathbb{F}_{j_0} for some $j_0 \leq j_1 + \log((\deg F_0 \cdots \deg F_{j_1})^r)$. By Lemma 7 and since $j_1 \leq k$ and $r \leq k$, we have, as required,

$$\begin{aligned} j_0 &\leq j_1 + r \log(\deg F_0 \cdots \deg F_{j_1}) \\ &\leq k + k \log((kD)^0 \cdots (kD)^k) \\ &\leq k + k^3 \log(kD). \end{aligned} \quad \blacktriangleleft$$

We are ready to provide the proof of the quantitative bound promised in Theorem 6.

Proof (of the second part of Theorem 6). It suffices to observe that m in the proof of the first part of Theorem 6 can be bounded by $k + k^3 \log(kD)$ thanks to Lemma 16. \blacktriangleleft

We finish this section by giving an example that shows that in the proof of Lemma 16, it may happen that $j_1 < j_0$, that is, after the stabilisation of the transcendence degree, there can be several non-trivial algebraic extensions until the fields themselves stabilise. Consider the polyrec system

$$\begin{cases} u_{n+1}^{(1)} &= (u_n^{(1)})^2 + (u_n^{(2)})^2, \\ u_{n+1}^{(2)} &= u_n^{(1)} + u_n^{(2)}. \end{cases}$$

We have $F_0^{(1)} = x_1, F_0^{(2)} = x_2$, then $F_1^{(1)} = x_1^2 + x_2^2, F_1^{(2)} = x_1 + x_2$ and $F_2^{(1)} = (x_1^2 + x_2^2)^2 + (x_1 + x_2)^2$. The chain (5) starts with

$$\mathbb{Q} \subseteq \mathbb{F}_0 = \mathbb{Q}(x_1) \subseteq \mathbb{F}_1 = \mathbb{F}_0(x_1^2 + x_2^2) = \mathbb{Q}(x_1, x_2^2) \subseteq \mathbb{F}_2.$$

Note that $\text{tr deg } \mathbb{F}_0 = 1$ and $\text{tr deg } \mathbb{F}_1 = 2$, which is the maximum value. However, the next extension $\mathbb{F}_1 \subseteq \mathbb{F}_2 = \mathbb{F}_1(F_2^{(1)}) = \mathbb{F}_1(x_1x_2)$ is non-trivial, because x_1x_2 does not belong to $\mathbb{Q}(x_1, x_2^2)$. In fact, it is algebraic of degree 2.

4.3 Obstacles towards the zeroness problem for polyrec sequences

Theorem 6 suggests the following algorithm for deciding zeroness of a polyrec sequence \mathbf{u} . Suppose the dimension of \mathbf{u} is k and the degree is D . We compute the first $p + 2$ entries of \mathbf{u} , where $p = k + k^3 \lceil \log(kD) \rceil$, and we verify whether all of them are zero. Obviously, if one of them is non-zero, then \mathbf{u} is non-zero. Otherwise, by Theorem 6, we expect that there is a rational function $R(y_0, \dots, y_m)$ for some $m \leq p$ such that

$$u_{n+m+1} = R(u_n, u_{n+1}, \dots, u_{n+m}) \quad \text{for all } n \in \mathbb{N}. \tag{8}$$

In particular,

$$0 = u_{m+1} = R(u_0, \dots, u_m) = R(0, \dots, 0).$$

Consequently,

$$u_{m+2} = R(u_1, \dots, u_{m+1}) = R(0, \dots, 0) = 0,$$

and a straightforward induction shows that $u_n = 0$ for all $n \in \mathbb{N}$. So we can declare that \mathbf{u} is the zero sequence.

The reasoning above is incorrect for the following reason. By Theorem 6, there is a rational function $R \in \mathbb{Q}(y_0, \dots, y_m)$ such that (8) holds when both sides are treated symbolically, as rational functions over a set of k variables \mathbf{x} that denote the vector of initial entries of the polyrec system defining \mathbf{u} . However, R is a rational function, hence when the variables are substituted with actual entries of the sequence \mathbf{u} , we may get an accidental 0 in the denominator of the right hand side. In other words, assertion (8) may be incorrect due to the right hand side being ill-defined, which renders the remainder of the reasoning flawed. To exemplify the problem we now present a case where this situation actually occurs.

Fix some $d \in \mathbb{N}$, and let

$$P(x) = x(x - 1) \dots (x - d + 1).$$

Define the sequence \mathbf{u} by setting

$$u_n = P(n) \quad \text{for all } n \in \mathbb{N}.$$

It is straightforward to see that \mathbf{u} is polyrec of dimension 2 and degree d : one can simply use one auxiliary sequence \mathbf{v} with $v_n = n$.

Observe that if instead of setting $v_0 = 0$, we set $v_0 = x$ for a formal variable x , the same polyrec system defines a sequence of polynomials $\hat{\mathbf{u}}$ over x defined as

$$\hat{u}_n = P(x + n) \quad \text{for all } n \in \mathbb{N}.$$

(Here, we also set initial condition $\hat{u}_0 = P(x)$.) Now, we may apply the reasoning behind Theorem 6 to find the rational function $R(y_0, y_1) \in \mathbb{Q}(y_0, y_1)$, defined as

$$R(y_0, y_1) = y_1 \cdot \frac{(d+1) \cdot y_1 - y_0}{y_1 + (d-1) \cdot y_0},$$

such that

$$\hat{u}_{n+2} = R(\hat{u}_n, \hat{u}_{n+1}) \quad \text{for all } n \in \mathbb{N}.$$

This, however, should be regarded as an equality of two rational functions over the variable x , which means that we cannot infer that

$$u_{n+2} = R(u_n, u_{n+1}) \quad \text{for all } n \in \mathbb{N},$$

because the right hand side can be undefined for specific values; and indeed, $R(0, 0)$ is undefined. The flawed reasoning from the beginning of this section would suggest that in order to verify the zeroness of \mathbf{u} , it suffices to check that the first three entries of \mathbf{u} are zero. However, we have $u_0 = u_1 = \dots = u_{d-1} = 0$ and $u_d = d! \neq 0$, so the algorithm would provide an incorrect answer.

Notice that if we had a promise that we never encounter a division by zero when recursively applying (8) from the given initial conditions, then the naïve zeroness algorithm presented at the beginning of the section would be sound. (The naïve algorithm is complete even without the promise.) However, deciding whether no division by zero occurs is essentially the Skolem problem for polyrec sequences, which, as mentioned in the introduction, is a long-standing open problem.

We are hopeful that the problem with accidentally hitting a singularity of R when starting from a polyrec sequence, as present in the example above, can somehow be circumvented, hence we state the following conjecture.

► **Conjecture 17.** *There is an elementary function $g: \mathbb{N} \rightarrow \mathbb{N}$ such that the following holds. Suppose \mathbf{u} is a polyrec sequence of dimension at most N and degree at most N such that $u_n = 0$ for all $n \leq g(N)$. Then $u_n = 0$ for all $n \in \mathbb{N}$.*

Note that a positive resolution to Conjecture 17 would immediately imply that the complexity of the zeroness problem for polyrec sequences is elementary.

5 Zeroness for polyrec is PSPACE-hard

In this section we discuss the following lower bound. The technical part of this section can be found in the appendix.

► **Theorem 18.** *For every fixed field \mathbb{F} , the zeroness problem for polyrec sequences over \mathbb{F} is PSPACE-hard.*

The lower bound claimed in the introduction follows from the theorem above by taking $\mathbb{F} = \mathbb{Q}$. Note also that together with Theorem 19 below, we can conclude that the problem is actually PSPACE-complete for every fixed *finite* field \mathbb{F} .

► **Theorem 19.** *For every fixed finite field \mathbb{F} , the zeroness problem for polyrec sequences over \mathbb{F} is in PSPACE.*

Proof. Let m be the cardinality of \mathbb{F} ; note that m is a fixed constant. A standard periodicity argument, e.g. as in the proof of [13, Theorem 4.1], shows that if \mathbf{u} is a polyrec sequence of dimension k , then it is zero if, and only if, it is zero for the first m^k steps. We can check the latter condition by storing in memory a k -tuple of values and computing the first m^k values of the sequence, which takes an amount of space which is polynomial in k . ◀

6 Conclusion

We believe that ratrec is a natural class of sequences with various promising questions deserving further investigation. Questions about decision problems are more natural for polyrec sequences due to their connection to polynomial automata and the issues with division by 0 in ratrec discussed in the introduction. Nevertheless, as discussed in this paper, understanding the properties of ratrec might lead to concrete complexity results for polyrec. The most natural problem for future work is to overcome the obstacles discussed in Section 4.3.

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, 2013. doi:10.1109/LICS.2013.65.
- 2 Corentin Barloy and Lorenzo Clemente. Bidimensional linear recursive sequences and universality of unambiguous register automata. In Markus Bläser and Benjamin Monmege, editors, *Proc. of STACS’21*, volume 187 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 8:1–8:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- 3 Corentin Barloy, Nathanaël Fijalkow, Nathan Lhote, and Filip Mazowiecki. A robust class of linear recurrence sequences. In *28th EACSL Annual Conference on Computer Science Logic, CSL 2020, January 13-16, 2020, Barcelona, Spain*, pages 9:1–9:16, 2020. doi:10.4230/LIPIcs.CSL.2020.9.
- 4 M. Benedikt, T. Duff, A. Sharad, and J. Worrell. Polynomial automata: Zeroness and applications. In *Proc. of LICS’17*, pages 1–12, June 2017. doi:10.1109/LICS.2017.8005101.
- 5 Vincent D. Blondel and Natacha Portier. The presence of a zero in an integer linear recurrent sequence is NP-hard to decide. *Linear Algebra and its Applications*, 351–352:91–98, 2002. Fourth Special Issue on Linear Systems and Control. doi:10.1016/S0024-3795(01)00466-9.
- 6 Adrien Boiret, Radosław Piórkowski, and Janusz Schmude. Reducing transducer equivalence to register automata problems solved by “Hilbert Method”. In Sumit Ganguly and Paritosh Pandya, editors, *Proc. of FSTTCS’18*, volume 122 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 48:1–48:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.FSTTCS.2018.48.
- 7 Mikołaj Bojańczyk. The Hilbert method for transducer equivalence. *ACM SIGLOG News*, 6(1):5–17, February 2019. doi:10.1145/3313909.3313911.
- 8 Mikołaj Bojańczyk and Wojciech Czerwiński. An automata toolbox, February 2018. URL: <https://www.mimuw.edu.pl/~bojan/paper/automata-toolbox-book>.
- 9 Mikołaj Bojańczyk and Janusz Schmude. Some remarks on deciding equivalence for graph-to-graph transducers. In Javier Esparza and Daniel Král, editors, *Proc. of MFCS’20*, volume 170 of *LIPIcs*, pages 19:1–19:14, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2020.19.

- 10 Alin Bostan, Arnaud Carayol, Florent Koechlin, and Cyril Nicaud. Weakly-unambiguous parikh automata and their link to holonomic series. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *Proc. of ICALP'20*, volume 168 of *LIPICs*, pages 114:1–114:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPICs.ICALP.2020.114.
- 11 Alin Bostan and Antonio Jiménez-Pastor. On the exponential generating function of labelled trees. *Comptes Rendus. Mathématique*, 358(9-10):1005–1009, 2020.
- 12 N. Bourbaki. *Algebra II*. Elements of Mathematics. Springer Verlag Berlin Heidelberg, 2003.
- 13 Michaël Cadilhac, Filip Mazowiecki, Charles Paperman, Michał Pilipczuk, and Géraud Sénizergues. On polynomial recursive sequences. *Theory of Computing Systems*, pages 1–22, 2021.
- 14 N. Chomsky and M. P. Schützenberger. The algebraic theory of context-free languages. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, volume 35 of *Studies in Logic and the Foundations of Mathematics*, pages 118–161. Elsevier, 1963. doi:10.1016/S0049-237X(08)72023-8.
- 15 Lorenzo Clemente. On the complexity of the universality and inclusion problems for unambiguous context-free grammars. In Laurent Fribourg and Matthias Heizmann, editors, Proceedings 8th International Workshop on *Verification and Program Transformation* and 7th Workshop on *Horn Clauses for Verification and Synthesis*, Dublin, Ireland, 25-26th April 2020, volume 320 of *EPTCS*, pages 29–43. Open Publishing Association, 2020. doi:10.4204/EPTCS.320.2.
- 16 J. Denef and L. Lipshitz. Decision problems for differential equations. *Journal of Symbolic Logic*, 54(3):941–950, 1989. doi:10.2307/2274755.
- 17 Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. Springer, 1st edition, 2009.
- 18 David S. Dummit and Richard M. Foote. *Abstract Algebra*. Wiley, 3rd edition, 2003. URL: <http://gen.lib.rus.ec/book/index.php?md5=36e6532b72807b9ef6b27e52e8c62ccc>.
- 19 Vojtěch Forejt, Petr Jančar, Stefan Kiefer, and James Worrell. Language equivalence of probabilistic pushdown automata. *Information and Computation*, 237:1–11, 2014.
- 20 Vesa Halava, Tero Harju, Mika Hirvensalo, and Juhani Karhumäki. Skolem’s problem - on the border between decidability and undecidability, 2005.
- 21 T. Harju and J. Karhumäki. The equivalence problem of multitape finite automata. *Theoretical Computer Science*, 78(2):347–355, 1991. doi:10.1016/0304-3975(91)90356-7.
- 22 Manuel Kauers and Peter Paule. *The Concrete Tetrahedron - Symbolic Sums, Recurrence Equations, Generating Functions, Asymptotic Estimates*. Texts & Monographs in Symbolic Computation. Springer, 2011. doi:10.1007/978-3-7091-0445-3.
- 23 Leonard Lipshitz. D-finite power series. *Journal of Algebra*, 122(2):353–373, 1989. doi:10.1016/0021-8693(89)90222-6.
- 24 Kurt Mahler. *Lectures on Transcendental Numbers*. Lecture Notes in Mathematics 546. Springer-Verlag Berlin Heidelberg, 1st edition, 1976.
- 25 J. Müller-Quade and R. Steinwandt. Basic algorithms for rational function fields. *Journal of Symbolic Computation*, 27(2):143–170, 1999. doi:10.1006/jsco.1998.0246.
- 26 Masayoshi Nagata. *Theory of Commutative Fields*. Translations of Mathematical Monographs, Vol. 125. American Mathematical Society, 1993. URL: <http://gen.lib.rus.ec/book/index.php?md5=249c3cba331671e0fd3c692d01b54b94>.
- 27 Joël Ouaknine and James Worrell. On linear recurrence sequences and loop termination. *ACM SIGLOG News*, 2(2):4–13, April 2015. doi:10.1145/2766189.2766191.
- 28 Christos H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- 29 Azaria Paz. *Introduction to probabilistic automata*. Academic Press, 1971.
- 30 Arto Salomaa and Marti Soittola. *Automata-theoretic aspects of formal power series*. Texts and Monographs in Computer Science. Springer, 1978. doi:10.1007/978-1-4612-6264-0.
- 31 Bruno Salvy. D-finiteness: Algorithms and applications. In *Proc. of ISAAC'05*, pages 2–3, New York, NY, USA, 2005. ACM. doi:10.1145/1073884.1073886.

- 32 Marcel Paul Schützenberger. On the definition of a family of automata. *Information and Control*, 4(2–3):245–270, 1961.
- 33 Helmut Seidl, Sebastian Maneth, and Gregor Kemper. Equivalence of deterministic top-down tree-to-string transducers is decidable. *J. ACM*, 65(4):21:1–21:30, April 2018. doi:10.1145/3182653.
- 34 Richard P. Stanley and Sergey Fomin. *Enumerative combinatorics*, volume 2 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1 edition, 2001.
- 35 Joris van der Hoeven. Computing with d-algebraic power series. *Applicable Algebra in Engineering, Communication and Computing*, 30(1):17–49, 2019. doi:10.1007/s00200-018-0358-y.
- 36 Tzeng Wen-Guey. On path equivalence of nondeterministic finite automata. *Information Processing Letters*, 58(1):43–46, 1996. doi:10.1016/0020-0190(96)00039-7.
- 37 James Worrell. Revisiting the equivalence problem for finite multitape automata. In Fedor V. Fomin, Rūsiņš Freivalds, Marta Kwiatkowska, and David Peleg, editors, *Proc. of ICALP’13*, pages 422–433, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- 38 Doron Zeilberger. A holonomic systems approach to special functions identities. *Journal of Computational and Applied Mathematics*, 32(3):321–368, 1990. doi:10.1016/0377-0427(90)90042-X.

A Zeroness for polyrec is PSPACE-hard

In order to speak about computational aspects of poly-rec sequences, we need to fix how they are encoded on input. For robustness, we choose to use arithmetic circuits. Formally, for a fixed field \mathbb{F} , a polynomial $P \in \mathbb{F}[x_1, \dots, x_k]$ is encoded by a circuit C that may use the following gates:

- binary addition and multiplication gates;
- nullary input gates, bijectively labelled with variables x_1, \dots, x_k ; and
- nullary constant gates, each labelled with an element of \mathbb{F} .

Note that subtraction can be emulated using addition and multiplication by the constant -1 . One of the gates is designated as the output gate. Given a valuation of variables with elements of \mathbb{F} , the values of the gates can be computed as expected, and the value yielded by the circuit C is the one computed for the output gate.

A.1 Extended polyrec sequences

In the reductions leading to the lower bound of Theorem 18 it is convenient to construct polyrec sequences according to a definition slightly more general than what we allowed in Definition 4. Namely, the definition of an *extended polyrec system* is the same as before, except that we generalize the format of the i th equation $u_{n+1}^{(i)} = P_i(\dots)$ by allowing $u_{n+1}^{(i)}$ to additionally depend on $u_{n+1}^{(1)}, \dots, u_{n+1}^{(i-1)}$. Thus, the i th equation takes the form:

$$u_{n+1}^{(i)} = P_i(u_n^{(1)}, \dots, u_n^{(k)}, u_{n+1}^{(1)}, \dots, u_{n+1}^{(i-1)}), \quad (9)$$

where now P_i is a polynomial in $k + i - 1$ variables. This more relaxed definition will help focus on the important aspects of the reduction presented in the rest of this section. The following lemma shows that the modification does not affect the complexity of the zeroness problem.

► **Lemma 20.** *Suppose \mathbf{u} is a sequence defined by an extended polyrec system S of dimension k , where each polynomial P_i is represented by circuit C_i . Then given the circuits C_i , one can in polynomial time construct a circuit C that represents a polyrec system S' of dimension k that also defines \mathbf{u} (with the same initial condition as S).*

Proof. Let the input gates of circuit C_i be labelled with $x_1, \dots, x_k, z_1, \dots, z_{i-1}$, where variables z_1, \dots, z_{i-1} respectively correspond to the values $u_{n+1}^{(1)}, \dots, u_{n+1}^{(i-1)}$ in (9). Construct the circuit C from the union of circuits C_1, \dots, C_k by performing the following operations for each $i \in \{1, \dots, k\}$:

- Fuse all input gates labelled x_i in circuits C_1, \dots, C_k into a single input gate labelled x_i .
- Fuse the output gate of C_i with all input gates labelled z_i in circuits C_{i+1}, \dots, C_k .

The output gates of C are the output gates of C_1, \dots, C_k . (Formally, we assumed that output gates must have fan-out 0, but this can be easily obtained by making a copy of each output gate.) It is straightforward to verify that the polyrec system S' that C represents defines the same k -tuple of sequences as S under the same initial condition. ◀

A.2 Reduction

We now proceed to the proof of Theorem 18. Let us fix the field \mathbb{F} ; in the reduction we will use only two constants from \mathbb{F} , namely 0 and 1. We reduce from the validity problem for Quantified Boolean Formulas (QBF), which is known to be PSPACE-complete (see, e.g., [28, Theorem 19.1]). Recall that the QBF validity problem amounts to determine whether a given QBF of the form

$$\psi = \exists x_1 \forall x_2 \dots Q_k x_k \varphi(x_1, \dots, x_k) \quad (10)$$

is true, where $\varphi(x_1, \dots, x_k)$ is quantifier-free, the variables with odd indices are quantified existentially, the remaining variables are quantified universally, and Q_k is either \exists or \forall depending on the parity of k . Hence, we are given a QBF ψ and we wish to construct, in polynomial time, a polyrec system S and its initial condition that define a sequence \mathbf{u} over \mathbb{F} such that the zeroness of \mathbf{u} is equivalent to the invalidity of ψ . By Lemma 20, it suffices to construct an extended polyrec system S with this property, where each polynomial P_i involved is represented by a separate circuit. In the following, the *size* of an extended polyrec system is the total size of its representation through circuits, which is constructed implicitly.

In the reduction it will be convenient to consider formulas obtained by fixing the truth values of a subset of the bound variables of ψ . For every $i \in \{0, \dots, k\}$ and $c_{i+1}, \dots, c_k \in \{0, 1\}$ we define the formula

$$\psi|_{c_{i+1}, \dots, c_k} = \exists x_1 \forall x_2 \dots Q_i x_i \varphi(x_1, \dots, x_i, c_{i+1}, \dots, c_k),$$

where Q_i is either \exists or \forall depending on the parity of i . In particular, for $i = k$ we get back ψ , and for $i = 0$ the formula $\psi|_{c_1, \dots, c_k}$ reduces to the truth value of $\varphi(c_1, \dots, c_k)$. We encode a quantifier Boolean formula φ into a polynomial P_φ using the following simulation of Boolean operators \neg , \wedge and \vee by arithmetic operations:

$$\begin{aligned} P_x &= x, \\ P_{\neg\varphi} &= 1 - P_\varphi, \\ P_{\varphi \wedge \psi} &= P_\varphi \cdot P_\psi, \\ P_{\varphi \vee \psi} &= P_{\neg(\neg\varphi \wedge \neg\psi)}. \end{aligned} \quad (11)$$

For example, $\tau(x, y) = (x \wedge y) \vee \neg y$ is encoded as $P_\tau(x, y) = 1 - (1 - xy)y$. The following straightforward claim shows that with the standard interpretation of 1 and 0 representing true, resp., false, such polynomials evaluate as expected. Note that this claims holds in any fixed field.

24:18 On Rational Recursive Sequences

▷ **Claim 21.** Let $\tau(x_1, \dots, x_k)$ be a Boolean formula and $P_\tau(x_1, \dots, x_k)$ its corresponding polynomial. For every $c_1, \dots, c_k \in \{0, 1\}$ we have $P_\tau(c_1, \dots, c_k) \in \{0, 1\}$ and

$$(c_1, \dots, c_k) \models \tau \iff P_\tau(c_1, \dots, c_k) = 1.$$

To ease the notation we will directly write formulas as polynomials; for instance, $P_\tau(x, y) = (x \wedge y) \vee \neg y$. All sequences in this section will be over $\{0, 1\}$ and the involved polynomials will be of the form P_τ .

Sequences $\mathbf{c}^1, \dots, \mathbf{c}^k$. The truth valuations of variables x_1, \dots, x_k will be encoded by sequences $\mathbf{c}^1, \dots, \mathbf{c}^k$, where for every i and n we have

$$c_n^i = \begin{cases} 0 & \text{if } n \bmod 2^i \text{ is less than } 2^{i-1}, \\ 1 & \text{otherwise.} \end{cases} \quad (12)$$

For example, the first eight values of $\mathbf{c}^1, \mathbf{c}^2, \mathbf{c}^3$ are

$$\begin{array}{rcccccccc} \mathbf{c}^1 & = & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \mathbf{c}^2 & = & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ \mathbf{c}^3 & = & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{array} .$$

▷ **Claim 22.** For every $i \geq 1$, the sequence \mathbf{c}^i is definable by an extended polyrec system over \mathbb{F} of size polynomial in i .

Proof. We proceed by induction on i . For $i = 1$, by definition we have $c_n^1 = 1 - c_{n-1}^1$, and thus we let

$$c_n^1 = P(c_{n-1}^1), \text{ with } P(x) = \neg x. \quad (13)$$

Now, suppose $i > 1$ and we have defined \mathbf{c}^{i-1} . We start by proving the following equality for every $n > 1$

$$c_n^i = \begin{cases} 1 - c_{n-1}^i & \text{if } c_{n-1}^{i-1} = 1 \text{ and } c_n^{i-1} = 0; \\ c_{n-1}^i & \text{otherwise.} \end{cases} \quad (14)$$

Notice that \mathbf{c}^i is periodic with period 2^i , i.e., $c_n^i = c_{n+2^i}^i$ for all n . Thus it suffices to prove (14) for $n \in \{1, \dots, 2^i\}$. By definition, $c_{n-1}^{i-1} = 1$ and $c_n^{i-1} = 0$ hold precisely for two values of $n \in \{1, \dots, 2^i\}$, namely for $n = 2^{i-1}$ and $n = 2^i$. Thus (14) is proved since $c_n^i = 0$ for $0 \leq n < 2^{i-1}$; $c_n^i = 1$ for $2^{i-1} \leq n < 2^i$; and $c_{2^i}^i = 0$.

Using (14) one can determine c_n^i given c_{n-1}^i, c_{n-1}^{i-1} , and c_n^{i-1} :

$$c_n^i = Q(c_{n-1}^i, c_{n-1}^{i-1}, c_n^{i-1}), \quad (15)$$

where $Q(x, y, z) = (\neg x \wedge (y \wedge \neg z)) \vee (x \wedge (\neg y \vee z))$. This follows from (14) and from the fact that $y \wedge \neg z$ and $\neg y \vee z$ are mutually exclusive formulas encoding the ‘‘if’’ condition in (14). It is clear that the constructed extended polyrec system is of size polynomial in i . ◁

Sequences $\mathbf{d}^0, \dots, \mathbf{d}^k$. We define sequences $\mathbf{d}^0, \dots, \mathbf{d}^k$, where for any $i \geq 0$ we have:

$$d_0^i = 0, \quad d_n^i = \begin{cases} 0 & \text{if } 2^i \nmid n \\ \llbracket \psi|_{c_{n-1}^{i+1}, \dots, c_{n-1}^k} \rrbracket & \text{otherwise,} \end{cases} \quad (16)$$

where for a closed formula ξ (i.e., with no free variables) $\llbracket \xi \rrbracket$ is 1 if ξ is true and 0 otherwise. Notice that the formula depends on $\mathbf{c}^1, \dots, \mathbf{c}^k$. Since \mathbf{d}^k is the zero sequence if, and only if, ψ is false, it suffices to show that each \mathbf{d}^i can be defined by an extended polyrec system of polynomial size.

We proceed by induction on i . In the base case $i = 0$,

$$d_n^0 = P_\varphi(c_{n-1}^1, \dots, c_{n-1}^k), \quad (17)$$

where P_φ is the polynomial obtained from the quantifier-free formula φ according to the rules in (11). (Notice that P_φ can be represented by an arithmetic circuit of size polynomial in the size of φ – this is where we use the conciseness of representation using circuits.) This fulfills the conditions in (16) since, for $n > 0$, $d_n^0 = 1$ if $(c_{n-1}^1, \dots, c_{n-1}^k) \models \varphi$ and $d_n^0 = 0$ otherwise.

Now, fix $i \geq 1$ and suppose that \mathbf{d}^{i-1} is defined. The goal is to define \mathbf{d}^i . Recall that if i is odd then x_i is quantified existentially, and otherwise x_i is quantified universally.

▷ **Claim 23.** Let $\otimes_i = \vee$ if i is odd and $\otimes_i = \wedge$ if i is even. For every $n > 0$ and $0 < i \leq k$, we have

$$d_n^i = \begin{cases} d_n^{i-1} \otimes_i d_{n-2^{i-1}}^{i-1} & \text{if } 2^i \mid n \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

Proof. We may focus only on the case $2^i \mid n$. Since x_i is quantified according to the parity of i , we have

$$\psi|_{c_{n-1}^{i+1}, \dots, c_{n-1}^k} = \psi|_{0, c_{n-1}^{i+1}, \dots, c_{n-1}^k} \otimes_i \psi|_{1, c_{n-1}^{i+1}, \dots, c_{n-1}^k}.$$

We claim that

$$d_n^{i-1} = \psi|_{1, c_{n-1}^{i+1}, \dots, c_{n-1}^k} \quad \text{and} \quad d_{n-2^{i-1}}^{i-1} = \psi|_{0, c_{n-1}^{i+1}, \dots, c_{n-1}^k}.$$

By (12) and the fact that $2^i \mid n$, we get $c_{n-1}^i = c_{2^{i-1}-1}^i = 1$, which proves the first equation. For the second equation, we observe that $c_{(n-1)-2^{i-1}}^i = c_{2^{i-1}-1}^i = 0$ and that $c_{(n-1)-2^{i-1}}^j = c_{n-1}^j$ for all $j > i$. The latter assertion readily follows from $2^i \mid n$ and (12). ◁

As an immediate consequence of Claim 23, we can write

$$d_n^i = S(d_n^{i-1}, d_{n-2^{i-1}}^{i-1}, c_{n-1}^{i-1}, c_n^{i-1}), \quad (19)$$

where $S(x, y, z, t) = (x \otimes_i y) \wedge (z \wedge \neg t)$ (by recalling that $c_{n-1}^{i-1} = 1, c_n^{i-1} = 0$ holds if, and only if, $2^i \mid n$, where $n > 0$). The issue with this recursive definition is that it requires access to the value $d_{n-2^{i-1}}^{i-1}$, which in general is not allowed in a polyrec system for $i \geq 2$ (not even in the extended variant). This will be addressed in the next section by introducing the last family of recursive sequences.

Sequences $\mathbf{f}^0, \dots, \mathbf{f}^{k-1}$. For every $1 \leq i \leq k$, the sequence \mathbf{f}^{i-1} is defined as

$$f_n^{i-1} = \begin{cases} 0 & \text{if } n \bmod 2^i \text{ is less than } 2^{i-1} \\ d_m^{i-1} & \text{otherwise,} \end{cases}$$

24:20 On Rational Recursive Sequences

where $m \leq n$ is the unique number such that $n - m < 2^{i-1}$ and $2^{i-1} \mid m$. Thus, \mathbf{f} is divided into blocks of length 2^{i-1} of equal elements, where every other block is either filled with zeros, or its value is determined by the value of an appropriate entry d_m^{i-1} . Observe that in particular, if $2^i \mid n$ then $f_{n-1}^{i-1} = d_{n-2^{i-1}}^{i-1}$. Thus, intuitively, the sequence \mathbf{f}^{i-1} is a “memory” that allows us to store the relevant value of \mathbf{d}^{i-1} from $2^{i-1} - 1$ steps back.

We now proceed to defining sequences $\mathbf{f}^0, \dots, \mathbf{f}^{k-1}$ using polyrec systems. Observe that $f_0^{i-1} = 0$ and for $n > 0$, we can write

$$f_n^{i-1} = \begin{cases} d_n^{i-1} & \text{if } c_{n-1}^{i-1} = 0 \text{ and } c_n^{i-1} = 1; \\ 0 & \text{if } c_{n-1}^{i-1} = 1 \text{ and } c_n^{i-1} = 0; \\ f_{n-1}^{i-1} & \text{otherwise.} \end{cases} \quad (20)$$

Notice that the value of f_n^{i-1} is copied from f_{n-1}^{i-1} unless c_{n-1}^{i-1}, c_n^{i-1} differ. To conclude, recall from (12) that this happens if, and only if, $2^{i-1} \mid n$.

▷ **Claim 24.** For every $i \geq 1$, the sequences \mathbf{d}^i and \mathbf{f}^{i-1} are definable by extended polyrec systems over \mathbb{F} of size polynomial in i and the size of φ .

Proof. Using (20), we may write \mathbf{f}^{i-1} as an extended polyrec sequence $f_0^{i-1} = 0$ and, for $n > 0$,

$$f_n^{i-1} = R(c_{n-1}^{i-1}, c_n^{i-1}, d_n^{i-1}, f_{n-1}^{i-1}), \quad (21)$$

where

$$R(x, y, z, t) = (z \wedge (\neg x \wedge y)) \vee (t \wedge ((x \wedge y) \vee (\neg x \wedge \neg y))).$$

In turn, this allows us to rewrite (19) as

$$d_n^i = S(d_n^{i-1}, f_{n-1}^{i-1}, c_{n-1}^{i-1}, c_n^{i-1}), \quad (22)$$

where S was defined in (19). Note that (21) and (22) are in the extended polyrec format provided that we write the equations for the \mathbf{f}^i 's after the equations for the \mathbf{d}^i 's, and the latter after the equations for \mathbf{c}^i 's (in order to avoid creating a cyclic dependency). In other words, the final extended polyrec system consists of equations (15), followed by (22), and followed by (21), where each set of equations is numbered naturally according to the indices of sequences.

The involved polynomials P_φ , R and S are all of size polynomial in the input size when represented as arithmetic circuits (R and S are even of constant size), and we have a polynomial number of equations. Thus, the definition above is an extended polyrec system of polynomial size. ◁

As discussed, Claim 24 finishes the proof of Theorem 18.

In the end, we discuss the Skolem problem: given a sequence \mathbf{u} to determine whether there is n such that $u_n = 0$. This problem was extensively studied for the class of linear recursive sequences (see e.g. [27]). For linear recursive sequences it is open whether the Skolem problem is decidable, but only NP-hardness is known [5, Corollary 2.1]. For polyrec sequences, decidability of the Skolem problem is also open, but we can improve the lower bound.

► **Corollary 25.** *The Skolem problem is PSPACE-hard for polyrec sequences.*

Proof. Notice that in the proof of Theorem 18 we define a system of sequences over $\{0, 1\}$. It remains to observe that for such sequences the zeroness problem and the Skolem problem reduce to each other. Indeed, the nonzeroness problem of a sequence \mathbf{u} over $\{0, 1\}$ is equivalent to the Skolem problem of \mathbf{v} defined as $v_n = 1 - u_n$. ◀

We conclude this section by noting that the reduction from QBF that we have presented produces a polyrec sequence which is identically zero if and only if the first exponentially many initial values thereof are zero. We are not aware of examples requiring longer witnesses of zeroness for polyrec sequences.