On the Counting Complexity of the Skolem Problem

Gorav Jindal

Max Planck Institute for Software Systems, Saarbrücken, Germany

Joël Ouaknine

Max Planck Institute for Software Systems, Saarbrücken, Germany

Abstract

The Skolem Problem asks, given an integer linear recurrence sequence (LRS), to determine whether the sequence contains a zero term or not. Its decidability is a longstanding open problem in theoretical computer science and automata theory. Currently, decidability is only known for LRS of order at most 4. On the other hand, the sole known complexity result is NP-hardness, due to Blondel and Portier.

A fundamental result in this area is the celebrated Skolem-Mahler-Lech theorem, which asserts that the zero set of any LRS is the union of a finite set and finitely many arithmetic progressions. This paper focuses on a computational perspective of the Skolem-Mahler-Lech theorem: we show that the problem of counting the zeros of a given LRS is #P-hard, and in fact #P-complete for the instances generated in our reduction.

2012 ACM Subject Classification Theory of computation → Models of computation; Theory of computation → Complexity theory and logic; Theory of computation → Complexity classes; Theory of computation → Problems, reductions and completeness

Keywords and phrases Linear recurrence sequences, Skolem Problem, Counting Complexity, #P-completeness, Subset Sum Problem

1 Introduction

1.1 LRS and the Skolem Problem

Linear recurrence sequences (LRS), such as the Fibonacci numbers, are a fundamental class of sequences ubiquitous across diverse domains within mathematics and computer science.

▶ **Definition 1.1** (Linear Recurrence Sequence). *An integer sequence $\boldsymbol{u} = \langle u_n \rangle_{n=0}^{\infty}$ is a linear recurrence sequence (LRS) of order k if k is the least positive integer such that the n^{th} term u_n of the sequence $\boldsymbol{u}$ can be written as:*

$$u_n = a_{k-1}u_{n-1} + \cdots + a_1 u_{n-k-1} + a_0 u_{n-k}\,, \tag{1}$$

for every $n \geq k$, where $a_j \in \mathbb{Z}$ for $0 \leq j \leq k-1$ and $a_0 \neq 0$. The LRS $\boldsymbol{u}$ is then uniquely determined by the initial values $u_0, u_1, u_2, \ldots, u_{k-1}$.

The primary motivation of this paper comes from the following famous problem.

▶ **Problem 1.1** (Skolem). *Given an LRS $\boldsymbol{u}$, decide if there exists $n \in \mathbb{N}$ such that $u_n = 0$.*

This longstanding open problem, going back nearly a hundred years, has garnered considerable attention in the literature. Currently, decidability of the Skolem Problem is known only for LRS of orders up to 4 [22, 17], a result established some forty years ago and unimproved since. A related question is the Positivity Problem, defined as follows:

▶ **Problem 1.2** (Positivity). *Given a LRS $\boldsymbol{u}$, decide if there exists $n \in \mathbb{N}$ such that $u_n < 0$.*

It is relatively straightforward to show that the Skolem Problem reduces to the Positivity Problem, by using the fact that LRS are closed under pointwise addition and multiplication. Both Skolem and Positivity serve as flagship problems in various subfields of theoretical computer science, such as program-termination analysis, see e.g. [4]. Decidability of the Positivity Problem is known for LRS of order at most 5 [18]. Moreover, it is known that decidability of the Positivity Problem for LRS of order 6 would entail major breakthroughs in the field of Diophantine approximation [18]. On the complexity front, the only known lower bounds are that the Skolem Problem (and consequently the Positivity Problem) is NP-hard [9, 20].

When investigating the Skolem Problem, it is natural to consider the zero sets of LRS. The well-known Skolem-Mahler-Lech theorem sheds light on the structure of the zero set $Z(\boldsymbol{u})$ of $\boldsymbol{u}$ defined as: $Z(\boldsymbol{u}) := \{n \in \mathbb{N} \mid u_n = 0\}$. This theorem asserts that the zero set of any linear recurrence sequence is the union of a finite set and finitely many arithmetic progressions. Formally:

▶ **Theorem 1.2** (Skolem-Mahler-Lech theorem, [13, 14]). *The zero set $Z(\boldsymbol{u})$ of an LRS $\boldsymbol{u}$ is the union of a finite set and a finite number of residue classes (arithmetic progressions) $\{n \in \mathbb{N} \mid n \equiv r \bmod m\}$.*

For an LRS $\boldsymbol{u}$ of order k as in Definition 1.1, the characteristic polynomial $\chi_{\boldsymbol{u}}$ of $\boldsymbol{u}$ is defined as the following univariate polynomial:

$$\chi_{\boldsymbol{u}}(x) := x^k - a_{k-1}x^{k-1} - a_{k-2}x^{k-2} - \cdots - a_1 x - a_0.$$

The roots of $\chi_{\boldsymbol{u}}$ are known as the *characteristic roots* of $\boldsymbol{u}$. Suppose that the characteristic roots are $\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$ with corresponding multiplicities $\{m_1, m_2, \ldots, m_d\}$. These characteristic roots give rise to an exponential-polynomial representation of u_n as [13]:

$$u_n = \sum_{j=1}^{d} A_j(n)\lambda_j^n ,$$

where $A_j(x)$ are univariate polynomials with complex coefficients of degree at most $m_j - 1$. We say that $\boldsymbol{u}$ is a *simple* LRS if $m_j = 1$ for all $0 \leq j \leq d$, or equivalently if $d = k$. We say that $\boldsymbol{u}$ is *degenerate* if there exist two distinct characteristic roots λ_i, λ_j such that $\frac{\lambda_i}{\lambda_j}$ is a root of unity. An equivalent formulation of the Skolem-Mahler-Lech theorem is the assertion that the zero set of a non-zero non-degenerate LRS is finite.

It is common to restrict oneself to non-degenerate LRS, since one can effectively decompose any given LRS into finitely many non-degenerate sub-LRS, see e.g. [8]. Upper bounds on the cardinality of the zero set of a given non-degenerate LRS are known; the sharpest such result is due to Amoroso and Viada [5]:

▶ **Theorem 1.3.** *Let $\boldsymbol{u}$ be a non-degenerate LRS having d distinct characteristic roots, each with multiplicity at most m. Then $\boldsymbol{u}$ has at most $(8d^m)^{8d^{6m}}$ zeros.*

Note that one can derive from Theorem 1.3 uniform bounds that depend only on the order of LRS. Unfortunately, no such upper bounds are known on the *magnitude* of zeros of LRS, which is of course why the Skolem Problem remains open to this day.

Taking a computational perspective on the Skolem-Mahler-Lech theorem, we now delve into the counting variant of the Skolem Problem. Let us write $|S|$ to denote the cardinality of a given finite set S. We define:

▶ **Problem 1.3** (#Skolem). *Given an LRS $\boldsymbol{u}$ and a positive integer m in binary, compute $|\{0 \leq n \leq m \mid u_n = 0\}|$.*

1.2 Our Results

We exploit some of the concepts presented in [20] to establish #P-hardness of the #Skolem Problem:

▶ **Theorem 1.4.** #Skolem *is* #P*-hard.*

In fact, we investigate a restricted variant of the #Skolem Problem denoted by $\#\mathsf{Skolem}_\omega$. This is inspired by the problem Skolem_ω defined and explored in [20]. Skolem_ω is a special case of the Skolem Problem for which the characteristic roots of the given LRS are restricted to be roots of unity. The problem Skolem_ω was shown to be NP-complete in [20].

▶ **Problem 1.4** ($\#\mathsf{Skolem}_\omega$). *Given an LRS $\boldsymbol{u}$ whose characteristic roots are roots of unity together with a positive integer m represented in binary, compute $|\{0 \leq n \leq m \mid u_n = 0\}|$.*

We establish that $\#\mathsf{Skolem}_\omega$ is #P-complete, which immediately entails Theorem 1.4. One might have anticipated that the NP-completeness of Skolem_ω would automatically imply the #P-completeness of $\#\mathsf{Skolem}_\omega$, but this is not the case because the NP-hardness reduction provided in [20] is not parsimonious.[1] We modify the reduction of [20] so that primes used in the reduction are chosen carefully, leading to #P-completeness of $\#\mathsf{Skolem}_\omega$. More precisely, we exploit and combine some of the ideas from [24, 20] to establish our results. Finally, we investigate the following inclusion problem for LRS, which serves as a generalization of both the Skolem and Positivity Problems.

▶ **Problem 1.5** (LRSInclusion). *Given two LRS $\boldsymbol{u}$ and $\boldsymbol{v}$, determine whether $\{u_n \mid n \in \mathbb{N}\} \subseteq \{v_n \mid n \in \mathbb{N}\}$.*

By making use of some of the key tools from [20], we establish the following hardness result for the LRSInclusion Problem.

▶ **Theorem 1.5.** LRSInclusion *is* Π_2^{P}*-hard.*

1.3 Proof Idea

To demonstrate the #P-hardness of $\#\mathsf{Skolem}_\omega$, we reduce the counting version of the Subset Sum Problem (#SSP) to $\#\mathsf{Skolem}_\omega$ (see Section 2 for precise definitions). Given an instance (S, b) of #SSP, we first construct a specific type of LRS, denoted $\boldsymbol{u}_{S,b}$, based on prime numbers in certain arithmetic progressions. The proof establishes a correspondence between solutions to the Subset Sum Problem and zeros of the constructed LRS. We demonstrate that modulo a particular prime q (upon which the construction of $\boldsymbol{u}_{S,b}$ is based), the number of solutions of (S, b) and the number of zeros of $\boldsymbol{u}_{S,b}$ are congruent (see Lemma 3.1). By repeating this procedure for various primes q and using Chinese remaindering, we are able to conclude that #SSP reduces to $\#\mathsf{Skolem}_\omega$. To establish that $\#\mathsf{Skolem}_\omega$ belongs to #P, we adapt the key ideas developed in [20]. For details, see Section 3.

2 Preliminaries

We recall relevant definitions and results related to counting complexity and linear recurrence sequences. For a more detailed discussion, we refer the reader to [6] and [13].

[1] There are more elaborate technical conditions on NP-completeness reductions which automatically guarantee the #P-completeness of associated counting variants, see e.g. [16]. However, it is far from clear whether the reductions proposed in [9, 20] satisfy such conditions.

▶ **Definition 2.1** (#P, [6]). *A function $f : \{0,1\}^* \to \mathbb{N}$ is in* #P *if there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time Turing machine M such that for every $x \in \{0,1\}^*$:*

$$f(x) = \left|\left\{y \in \{0,1\}^{p(|x|)} \mid M(x,y) = 1\right\}\right|.$$

We define FP to be the set of functions from $\{0,1\}^*$ to $\mathbb{N}$ computable by a deterministic polynomial-time Turing machine.

▶ **Definition 2.2** ([6]). *A function $f : \{0,1\}^* \to \mathbb{N}$ is* #P*-hard if every $g \in$ #P is also in* FP^f*. f is said to be* #P*-complete if it is* #P*-hard and it is also in* #P*.*

An LRS $\boldsymbol{u}$ is said to be periodic with period p if $u_n = u_{n+p}$ for all non-negative n. For an LRS $\boldsymbol{u}$ of order k, we denote by $\|\boldsymbol{u}\|$ the size of the bit representation of the coefficients of the corresponding recurrence relation (Equation (1)), namely $a_0, a_1, a_2, \ldots, a_{k-1}$, and of the initial values $u_0, u_1, u_2, \ldots, u_{k-1}$. LRS are closed under addition; moreover, if $\boldsymbol{u}$ and $\boldsymbol{v}$ are two given LRS, one can construct the LRS $\boldsymbol{u} + \boldsymbol{v} = \langle u_n + v_n \rangle_{n=0}^{\infty}$ efficiently:

▶ **Theorem 2.3** (Lemma 2 in [20]). *Suppose $\boldsymbol{u^{(1)}}, \boldsymbol{u^{(2)}}, \ldots, \boldsymbol{u^{(\ell)}}$ are LRS of orders k_1, k_2, $\ldots, k_\ell$ respectively, then we have:*

1. *$\boldsymbol{u} := \boldsymbol{u^{(1)}} + \boldsymbol{u^{(2)}} + \cdots + \boldsymbol{u^{(\ell)}}$ is an LRS of order at most $k_1 + k_2 + \cdots + k_\ell$.*
2. *$\boldsymbol{u}$ can be constructed in* $\mathsf{poly}\left(\|\boldsymbol{u^{(1)}}\| + \|\boldsymbol{u^{(2)}}\| + \cdots + \|\boldsymbol{u^{(\ell)}}\|\right)$ *time.*
3. *The characteristic polynomial $\chi_{\boldsymbol{u}}$ is a factor of $\prod_{i=1}^{\ell} \chi_{\boldsymbol{u^{(i)}}}$.*

To show #P-hardness of $\#\mathsf{Skolem}_\omega$, we reduce the Subset Sum Problem to $\#\mathsf{Skolem}_\omega$. Given a set $T \subseteq \mathbb{Z}$, write $\sum(T)$ to denote the sum of its elements. For a pair (S, b) with $S \subseteq \mathbb{Z}$ and $b \in \mathbb{Z}$, let $\mathrm{W}(S, b)$ denote the set of solutions of the subset sum set instance (S, b), *i.e.*,

$$\mathrm{W}(S, b) := \{T \subseteq S \mid \sum(T) = b\}.$$

▶ **Problem 2.1** (Subset Sum Problem). *Given an integer b and a set of integers $S = \{s_1, s_2, \ldots, s_m\}$, we define the following problems:*

1. *Decision variant (*SSP*): Determine whether there exists a subset $T \subseteq S$ such that the sum of all the integers in T equals b, i.e., decide if $\mathrm{W}(S, b) \neq \emptyset$.*
2. *Counting variant (*#SSP*): Count the number of subsets $T \subseteq S$ such that the sum of all the integers in T equals b, i.e., compute $|\mathrm{W}(S, b)|$.*

It is known that SSP is NP-complete and this fact was used in [20] to show that Skolem_ω is itself NP-complete. Moreover, it is also known that #SSP is #P-complete [12, 10], as stated below.

▶ **Theorem 2.4** ([12, 10]). #SSP *is* #P*-complete.*

3 #P-completeness of $\#\mathsf{Skolem}_\omega$

In this section we show that $\#\mathsf{Skolem}_\omega$ is #P-complete.

3.1 #Skolem$_\omega$ is #P-hard

Here we extend the ideas of [20] to prove that #Skolem$_\omega$ is #P-hard. To this end, we invoke the #P-hardness of the #SSP Problem defined in Problem 2.1. Let us assume that we are given a set $S = \{s_1, s_2, \ldots, s_m\} \subseteq \mathbb{Z}$ of integers together with $b \in \mathbb{Z}$. Suppose q is an odd prime. We now construct the desired LRS $\boldsymbol{u}_{S,b}$. Suppose $p_1, p_2, \ldots, p_m$ are m distinct primes in the arithmetic progression $\{aq + 2 \mid a \in \mathbb{N}\}$. Then for each $1 \le i \le m$, we define the LRS $\boldsymbol{u}^{(i)}$ whose n^{th} term is given as:

$$\boldsymbol{u}^{(i)}_n := \begin{cases} s_i & \text{if } n = 0 \\ 0 & \text{if } 1 \le n < p_i \\ \boldsymbol{u}^{(i)}_{n-p_i} & \text{otherwise}\,. \end{cases}$$

So $\boldsymbol{u}^{(i)}$ is a periodic LRS where every p_i^{th} term is s_i and all the other terms are zero. We can now define the LRS $\boldsymbol{u}_{S,b}$, as follows:

$$(\boldsymbol{u}_{S,b})_n = \sum_{i=1}^{m} \boldsymbol{u}^{(i)}_n - b\,.$$

Notice that the characteristic polynomial $\chi_{\boldsymbol{u}^{(i)}}$ of $\boldsymbol{u}^{(i)}$ is $x^{p_i} - 1$ and that of the constant LRS b is $x - 1$. Thus thanks to Theorem 2.3, the characteristic polynomial $\chi_{\boldsymbol{u}_{S,b}}$ of $\boldsymbol{u}_{S,b}$ is a factor of $(x-1)\prod_{i=1}^{m}(x^{p_i} - 1)$. Therefore all characteristic roots of $\boldsymbol{u}_{S,b}$ are roots of unity, and counting the number of zeros of $\boldsymbol{u}_{S,b}$ is an instance of the #Skolem$_\omega$ Problem.

For an LRS $\boldsymbol{u}$ and positive integer B, let us write $Z_B(\boldsymbol{u})$ to denote the set $\{0 \le n < B \mid u_n = 0\}$ of zeros of $\boldsymbol{u}$ of magnitude less than B. The following result is key to showing that #Skolem is #P-hard.

▶ **Lemma 3.1.** *For any #SSP instance (S, b) and odd prime q, define $B := \prod_{i=1}^{m} p_i$. Then we have:*

$$|\mathrm{W}\,(S,b)| \equiv |Z_B(\boldsymbol{u}_{S,b})| \pmod q\,.$$

Proof. Suppose $T \in \mathrm{W}\,(S,b)$. Define $N_T := \prod_{s_i \in T} p_i$. Now consider the set:

$$C_T := \{0 \le k < B \mid \gcd(k, B) = N_T\}\,. \tag{2}$$

It is clear that $(\boldsymbol{u}_{S,b})_k = 0$ for all $k \in C_T$. By definition of C_T, it is also clear that C_T and $C_{T'}$ are disjoint for distinct $T, T' \in \mathrm{W}\,(S,b)$. In fact, C_T is exactly the set of indices k less than B *corresponding* to T such that $(\boldsymbol{u}_{S,b})_k = 0$. Moreover, Equation (2) implies:

$$C_T = \{mN_T \mid 0 \le m < B/N_T, \gcd(m, B/N_T) = 1\}\,. \tag{3}$$

Equation (3) clearly entails that $|C_T| = \varphi(B/N_T) = \prod_{s_i \notin T}(p_i - 1)$, where φ is Euler's totient function. Since the p_i's are in the set $\{aq + 2 \mid a \in \mathbb{N}\}$, we have that $p_i = a_i q + 2$ for some $a_i \in \mathbb{N}$. Hence:

$$p_i - 1 = a_i q + 1$$
$$p_i - 1 \equiv 1 \pmod q$$

Therefore $|C_T| \equiv 1 \pmod q$. By summing over all $T \in \mathrm{W}\,(S,b)$, we get:

$$|\mathrm{W}\,(S,b)| \equiv |Z_B(\boldsymbol{u}_{S,b})| \pmod q\,,$$

as required. ◀

To complete the proof of #P-hardness of #Skolem$_\omega$, we also need to show that the p_i's above are not "too large" and also that $\boldsymbol{u}_{S,b}$ can be constructed efficiently from a given instance (S, b) of #SSP. To this end, we make use of the results established in Section 4.

▶ **Theorem 3.2.** *#Skolem$_\omega$ is #P-hard.*

Proof. As stated above, we reduce #SSP to #Skolem$_\omega$. We are given a subset sum instance $(S = \{s_1, s_2, \ldots, s_m\}, b)$. By using Corollary 4.4, we find odd primes $q_1, q_2, \ldots, q_m$ and primes $\{p_{ij} \mid 1 \leq i, j \leq m\}$ such that p_{ij} is in the arithmetic progression $\{aq_i + 2 \mid a \in \mathbb{N}\}$. This is always possible as long as $m \geq m_0$, where m_0 is some absolute effective constant. If $m < m_0$ then #SSP can be trivially solved in polynomial time by a brute-force algorithm, hence we can assume that $m \geq m_0$. Corollary 4.4 implies that we can find such q_i, p_{ij} in $\mathsf{poly}(m)$ time and also that q_i, p_{ij} are polynomially bounded in m.

Fix a prime $q \in \{q_1, q_2, \ldots, q_m\}$. Consider m primes $p_1, p_2, \ldots, p_m$ in the arithmetic progression $\{aq + 2 \mid a \in \mathbb{N}\}$, generated by Corollary 4.4. Now construct the LRS $\boldsymbol{u}_{S,b}$ as above. By using Theorem 2.3 and bounds on the p_i's, $\boldsymbol{u}_{S,b}$ can be constructed in $\mathsf{poly}(I)$ time, where I is the length of the binary description of the input $(S = \{s_1, s_2, \ldots, s_m\}, b)$. We then use a #Skolem$_\omega$ oracle on the #Skolem$_\omega$ instance $(\boldsymbol{u}_{S,b}, B)$, where $B := \prod_{i=1}^m p_i$. By invoking Lemma 3.1, we obtain that

$$|\mathrm{W}\,(S, b)| \equiv |Z_B(\boldsymbol{u}_{S,b})| \pmod{q}.$$

Now we repeat the above procedure for all $q \in \{q_1, q_2, \ldots, q_m\}$. Hence by invoking the #Skolem$_\omega$ oracle m times, we can compute $|\mathrm{W}\,(S, b)| \bmod q_i$ for all $i \in \{1, \ldots, m\}$. Since $|\mathrm{W}\,(S, b)| \leq 2^m$ and $\prod_{i=1}^m q_i > 2^m$, by using Chinese remaindering we can recover the exact value of $|\mathrm{W}\,(S, b)|$. It is moreover known that Chinese remaindering can be performed in polynomial time, see e.g. [23, Chapter 5]. This implies that #SSP is in $\mathsf{FP}^{\#\mathsf{Skolem}_\omega}$. By using Theorem 2.4, we conclude that #Skolem$_\omega$ is #P-hard. ◀

3.2 #Skolem$_\omega$ is in #P

The proof of membership of #Skolem$_\omega$ in #P follows from the ideas developed in the proof of Theorem 8 in [20]. Given an LRS $\boldsymbol{u}$ whose characteristic roots are roots of unity, in the proof of Theorem 8 in [20], the following facts are proved:

- If $\boldsymbol{u}$ has a zero at all then there exists an $N \leq 2^{\mathsf{poly}(\|\boldsymbol{u}\|)}$ such that $u_N = 0$.
- For any $N \leq 2^{\mathsf{poly}(\|\boldsymbol{u}\|)}$, the condition $u_N = 0$ can be verified in deterministic polynomial time.

We adapt the above ideas to show that #Skolem$_\omega$ ∈ #P. To this end, consider a given #Skolem$_\omega$ instance $(\boldsymbol{u}, B)$. We define the function $f_{\boldsymbol{u},B} : \{0, 1, \ldots, B\} \to \{0, 1\}$ by:

$$f_{\boldsymbol{u},B}(n) = \begin{cases} 1 & \text{If } u_n = 0 \\ 0 & \text{Otherwise}. \end{cases}$$

We now show that $f_{u,B}$ can be computed in $\mathsf{poly}(\|\boldsymbol{u}\|, \log B)$ time, which in turn implies that #Skolem$_\omega$ is in #P. Recall the exponential-polynomial representation of $\boldsymbol{u}$, whereby we have

$$u_n = \sum_{j=1}^{d} A_j(n)\lambda_j^n,$$

where λ_j are roots of unity and $A_j(n)$ are polynomials. Since λ_j's are roots of unity, the asymptotic behavior of u_n is controlled by the A_j polynomials. Hence the bit size of the absolute value of u_n should grow polynomially in $\|\boldsymbol{u}\|$ and $\log n$. This intuition is made precise in the proof of Theorem 8 in [20]. We now recall and adapt the key ideas of [20], to show that $f_{\boldsymbol{u},B}$ can be computed in $\mathsf{poly}(\|\boldsymbol{u}\|, \log B)$ time. To this end, we recall the following definitions from [20]: for $m \in \mathbb{N}, P_m(x) := \sum_{j=1}^{d} A_j(x)\lambda_j^m$ and $P = \{P_m \mid m \in \mathbb{N}\}$.

▶ **Lemma 3.3** (Lemma 10 in [20]). *The set P is finite. In fact $P = \{P_m \mid m \in \{0, 1, \ldots, k^{3k}\}\}$, where k is the order of $\boldsymbol{u}$.*

Moreover, it is also shown in [20] that the coefficients of all polynomials in P are rational numbers which are also $\mathsf{poly}(\|\boldsymbol{u}\|)$ bounded in bit size and can be computed in polynomial time. Notice that for any $n \in \mathbb{N}$, we have $u_n = P_n(n)$. Since $P_n \in P$, the coefficients of P_n are also bounded by $\mathsf{poly}(\|\boldsymbol{u}\|)$ in bit size. This implies that u_n is bounded by $\mathsf{poly}(\|\boldsymbol{u}\|, \log n)$ in bit size, and thus we can compute u_n in $\mathsf{poly}(\|\boldsymbol{u}\|, \log n)$ time (using iterated squaring of the companion matrix, see e.g [11, 20]). One can then easily check whether $u_n = 0$. Therefore the function $f_{\boldsymbol{u},B}$ can indeed be computed in $\mathsf{poly}(\|\boldsymbol{u}\|, \log B)$ time, whence:

▶ **Theorem 3.4.** *$\#\mathsf{Skolem}_\omega$ is in $\#\mathsf{P}$.*

▶ Remark 3.5. The above approach for deciding $u_n = 0$ given $\boldsymbol{u}$ and n (by iterated squaring of the companion matrix) also applies for general LRS. However in such instances it can happen that the binary representation of u_n has bit size $\mathsf{poly}(\|\boldsymbol{u}\|, n)$ instead of $\mathsf{poly}(\|\boldsymbol{u}\|, \log n)$. Checking zeroness of such huge u_n is not known to be feasible in deterministic polynomial time. This is a special case of the well-known EquSLP Problem [3], which can be handled in randomized polynomial time.

4 Finding Primes in Arithmetic Progression

This section presents an algorithm for finding primes in arithmetic progressions. This algorithm is adapted from [24, Section 4] with only minor modifications, and is included here for completeness. We use $\pi(x)$ to denote the prime-counting function, *i.e.*,

$$\pi(x) := \{n \in \mathbb{N}, 2 \le n \le x \mid n \text{ is prime}\}.$$

For $a, d \in \mathbb{N}$ with $\gcd(a, d) = 1$, we define $\pi(x; d, a)$ to be the number of primes below x that belong to the arithmetic progression $\{a + dt \mid t \in \mathbb{N}\}$, *i.e.*,

$$\pi(x; d, a) := \{n \in \mathbb{N}, 2 \le n \le x, n \equiv a \bmod d \mid n \text{ is prime}\}.$$

With this notation we have $\pi(x) = \pi(x; 1, 0)$. The prime number theorem for arithmetic progressions states that

$$\pi(x; d, a) \sim \frac{\pi(x)}{\varphi(d)} \text{ as } x \to \infty \tag{4}$$

provided $\gcd(a, d) = 1$, where φ is Euler's totient function. For an effective version of Equation (4), the following result was proven in [2]:

▶ **Theorem 4.1** ([2], Fact 4.10 in [24]). *There exist positive constants $x_0, \eta \in \mathbb{R}$, $t \in \mathbb{N}$ such that for all $x, y \in \mathbb{R}$ with $y \ge x \ge x_0$ there exists $E \subseteq \mathbb{N}$ with $|E| < t, \min(E) \ge \log x$ such that*

$$\pi(y; d, a) \ge \frac{\pi(x)}{2\varphi(d)}$$

whenever $\gcd(a,d) = 1$, $1 \le d \le \min\{x^{\frac{2}{5}}, yx^{-\frac{3}{5}}\}$, *and* E *contains no divisor of* d. *Furthermore,* $e \ge x^{\eta}$ *for all but at most one* $e \in E$. *We can assume* $\eta \le \frac{2}{7}$.

▶ **Corollary 4.2.** *There exist positive constants* $x_0, \eta \in \mathbb{R}$ *such that for all* $x \in \mathbb{R}$ *with* $x \ge \max\{5, x_0\}$ *there exists* $E \subseteq \mathbb{N}$ *with* $\min(E) \ge \log x$ *and we have*

$$\pi(x; q, 2) \ge \frac{\pi(x)}{2\varphi(q)}$$

for any odd prime q *with* $q \le x^{\frac{2}{5}}$ *and* $q \notin E$. *Furthermore, we can assume* $\eta \le \frac{2}{7}$ *and* $e \ge x^{\eta}$ *for all but at most one* $e \in E$.

Proof. In Theorem 4.1, we choose $y = x$ and hence $\min\{x^{\frac{2}{5}}, yx^{-\frac{3}{5}}\}$ can be replaced by $x^{\frac{2}{5}}$. Clearly the condition $\gcd(a,d) = 1$ is satisfied because $a = 2$ and $d = q$ is an odd prime. First notice that $1 \notin E$, since $\min(E) \ge \log x$. Hence the only way E can contain a divisor of $q = d$ is if $q \in E$. Hence the condition of E containing no divisor of d can be replaced by $q \notin E$. Observe that we can now just drop the parameter t from Theorem 4.1 to get Corollary 4.2. ◀

Suppose x_0, η are the constants defined in Corollary 4.2. We define:

$$N := \max\left\{2 \cdot 10^{34}, \frac{2}{5\eta}\right\}.$$

Algorithm 1 Primes in Arithmetic Progression

Input : $n \in \mathbb{N}$ in unary.
Output: n odd primes $\{q_i \mid i \in [n]\}$ and n^2 odd primes $\{p_{ij} \mid i, j \in [n]\}$ such that $p_{ij} \equiv 2 \bmod q_i$ for all $i, j \in [n]$.

$B \leftarrow 3n \log^2 n$.
By using the primality testing algorithm of [1], find first $n+1$ primes $q_0, q_1, q_2, \ldots, q_n$ that are at least $\frac{B}{2}$.
foreach $i \in \{0, 1, 2, \ldots, n\}$ **do**
  $k \leftarrow 0$. `//` k `is the number of primes found in` $\{2 + q_i t \mid t \in \mathbb{N}\}$.
  foreach $j \in \{2 + q_i t \mid t \in \mathbb{N}\}$ **do**
    if $j > \max\{B^{\frac{1}{\eta}}, x_0\}$ or $k \ge n$ **then**
      `// Found` n `primes in` $\{2 + q_i\mathbb{N}\}$ `or` j `is "too large".`
      break
    end
    Using the primality testing algorithm of [1], check if j is prime.
    if j is prime **then**
      $k \leftarrow k + 1$
      $p_{ik} \leftarrow j$
    end
  end
end

▶ **Theorem 4.3.** *Algorithm 1 uses* poly(n) *bit operations and if* $n \ge N$ *then its output has the following properties for all* $i \in [n]$:

1. $n \log^2 q_i \le \frac{B}{2} \le q_i \le B$.

2. $p_{ij} \le \max\{B^{1/\eta}, x_0\}$ *and* $p_{ij} \in \{2 + q_i t \mid t \in \mathbb{N}\}$ *for all* $j \in [n]$.

Proof. We use the following fact from [19]:

$$B - \pi\left(\frac{B}{2}\right) \ge \frac{3B}{10\log\left(\frac{B}{2}\right)}$$

to obtain for $n \ge 11$:

$$\pi(B) - \pi\left(\frac{B}{2}\right) \ge \frac{3B}{10\log\left(\frac{B}{2}\right)} = \frac{9n\log^2 n}{10} \cdot \frac{1}{\log\left(\frac{3}{2}n\log^2 n\right)} \ge n+1.$$

Hence $\frac{B}{2} \le q_i \le B$. First notice that:

$$\frac{3}{2}\log^2 n \ge \log^2(3n\log^2 n) \ge \log^2 B.$$

Hence:

$$q_i \ge q_0 \ge \frac{B}{2} = \frac{3}{2}n\log^2 n \ge n\log^2 B \ge n\log^2 q_n \ge qn\log^2 q_i.$$

Hence the first condition $n\log^2 q_i \le \frac{B}{2} \le q_i \le B$ is proven now.

Suppose $x := \max\{B^{1/\eta}, x_0\}$. It is clear that $q_i \le B \le x^\eta$ for all $i \in [n]$. We first see:

$$\begin{array}{rcl} (2-5\eta)/2\eta & \ge & 1, \\ \eta B^{(2-5\eta)/2\eta} & \ge & \eta n^{(2-5\eta)/2\eta} \ge \frac{2}{5}, \\ \eta B^{1/\eta} & \ge & \frac{2}{5}B^{5/2}. \end{array}$$

For convenience: $Q := \{q_i \mid i \in [n]\}$. We first notice that $|Q \cap E| \le 1$. This is because we have $q_i \le x^\eta$ for all $i \in [n]$ and we have $e \ge x^\eta$ for all but at most one $e \in E$. Suppose $q_i \in Q \setminus E$. We also have $q_i \le B \le x^\eta < x^{\frac{2}{5}}$. Hence all the conditions on q in Corollary 4.2 are satisfied. We use the fact that $\pi(x) \ge \frac{x}{\log x}$. Now Corollary 4.2 implies that:

$$\begin{array}{rcl} \pi(x; q_i, 2) \ge & \frac{\pi(x)}{2\phi(q_i)} \ge & \dfrac{B^{1/\eta}}{2\log(B^{1/\eta})B} \\ = & \frac{\eta B^{1/\eta - 1}}{2\log B} \ge & \dfrac{\frac{2}{5}B^{\frac{3}{2}}}{2\log B} = \dfrac{(3n\log^2 n)^{\frac{3}{2}}}{5\log(n\log^2 n)} > n. \end{array}$$

Since there can be at most one "bad prime" $q_i \in E$, by renaming this prime to q_0, we obtain the desired claim. The polynomial running time of the algorithm is easy to verify. ◀

▶ **Corollary 4.4.** *There exists* $m_0 \in \mathbb{N}$ *such that for all* $m \ge m_0$*, in* $\mathsf{poly}(m)$ *bit operations we can find* m *odd primes* $q_1, q_2, \ldots, q_m$ *and* m^2 *odd primes* $\{p_{ij} \mid i, j \in [m]\}$ *such that:*

1. $q_i \le 3m\log^2 m$ *for all* $i \in [m]$.
2. *For* $i, j \in [m]$, p_{ij} *is in the arithmetic progression* $\{2 + q_i t \mid t \in \mathbb{N}\}$.
3. *For all* $i, j \in [m]$, $p_{ij} \le \left(3m\log^2 m\right)^4$.

Proof. We directly use Theorem 4.3 for $m = n$ and $N = m_0$. It is clear that primes q_i, p_{ij} satisfy the properties claimed in Item 1 and Item 2. The bound on p_{ij} claimed in Item 3 follows from the fact that x_0 is an absolute constant in Corollary 4.2 and η can be chosen to be anything smaller than $2/7$; here we use $\eta = 1/4$. ◀

5 Checking inclusion of LRS

In this section, we consider the problem LRSInclusion defined in Section 1.2. Recall, given two LRS $\boldsymbol{u}$ and $\boldsymbol{v}$ as input, that we want to determine whether $\{u_n \mid n \in \mathbb{N}\} \subseteq \{v_n \mid n \in \mathbb{N}\}$. Observe that the Positivity Problem reduces to LRSInclusion, as can be seen by choosing $\boldsymbol{u}$ to be the LRS of positive integers. Therefore both Skolem and Positivity reduce to LRSInclusion. This section is now concerned with showing that LRSInclusion is hard for the second level of the polynomial hierarchy, though only NP-hardness is known for Skolem and Positivity. For a discussion of the polynomial hierarchy, we refer the reader to [6].

To establish hardness of LRSInclusion, we reduce from the known hardness of following problem.

▶ **Problem 5.1** (Generalized Subset Sum (GSSP)). *Given two vectors $a, b \in \mathbb{Z}^m$ and $t \in \mathbb{Z}$, determine whether $\exists x \, \forall y \, [ax + by \neq t]$ holds, where $x, y \in \{0,1\}^m$.*

▶ **Theorem 5.1** ([7, 15, 21]). GSSP *is Σ_2^{P}-complete.*

Theorem 5.1 immediately implies the following:

▶ **Corollary 5.2.** *Consider the problem: given two vectors $a, b \in \mathbb{Z}^m$ and $t \in \mathbb{Z}$, determine whether $\forall x \, \exists y \, [ax + by = t]$ holds, where $x, y \in \{0,1\}^m$. This problem is Π_2^{P}-complete.*

▶ **Theorem 5.3.** LRSInclusion *is Π_2^{P}-hard.*

Proof. Given two vectors $a, b \in \mathbb{Z}^m$ and $t \in \mathbb{Z}$, we want to decide if $\forall x \, \exists y \, [ax + by = t]$ holds, with $x, y \in \{0,1\}^m$. This problem is Π_2^{P}-complete by Corollary 5.2. We reduce this problem to LRSInclusion. For every $i \in \{1, 2, \ldots, m\}$, let p_i be the i^{th} prime. Then, for each $i \in \{1, 2, \ldots, m\}$, we have an LRS $\boldsymbol{u}^{(i)}$, whose n^{th} term is defined as:

$$\boldsymbol{u}_n^{(i)} = \begin{cases} a_i & \text{if } n = 0 \\ 0 & \text{if } 1 \leq n < p_i \\ \boldsymbol{u}_{n-p_i}^{(i)} & \text{otherwise}\,. \end{cases}$$

So $\boldsymbol{u}^{(i)}$ is a periodic LRS in which every p_i^{th} term is a_i and all the other terms are zero. We now define the LRS $\boldsymbol{u}$ as:

$$\boldsymbol{u} = \sum_{i=1}^{m} \boldsymbol{u}^{(i)}\,.$$

For each $i \in \{1, 2, \ldots, m\}$, we define an LRS $\boldsymbol{v}^{(i)}$, whose n^{th} term is defined as:

$$\boldsymbol{v}_n^{(i)} = \begin{cases} b_i & \text{if } n = 0 \\ 0 & \text{if } 1 \leq n < p_i \\ \boldsymbol{v}_{n-p_i}^{(i)} & \text{otherwise}\,. \end{cases}$$

So $\boldsymbol{v}^{(i)}$ is a periodic LRS in which every p_i^{th} term is b_i and all the other terms are zero. We now define the LRS $\boldsymbol{v}$ as:

$$\boldsymbol{v} = t - \sum_{i=1}^{m} \boldsymbol{u}^{(i)}\,.$$

By using Theorem 2.3 and the ideas in Section 3, it is easy to see that $\boldsymbol{u}$ and $\boldsymbol{v}$ can be constructed in polynomial time (in terms of the input size). Let us write $S_{\boldsymbol{u}}$ and $S_{\boldsymbol{v}}$ to denote the sets of integers occurring in $\boldsymbol{u}$ and $\boldsymbol{v}$ respectively. Observe that:

$$S_{\boldsymbol{u}} = \{ax \mid x \in \{0,1\}^m\},$$
$$S_{\boldsymbol{v}} = \{t - by \mid y \in \{0,1\}^m\}.$$

Thanks to the above, our LRS inclusion $S_{\boldsymbol{u}} \subseteq S_{\boldsymbol{v}}$ can be rewritten as $\forall x \, \exists y \, [ax + by = t]$. Hence LRSInclusion is Π_2^{P}-hard, as claimed. ◀

6 Conclusion and Open Problems

We have established that counting the zeros of a given linear recurrence sequence is #P-hard, extending the known NP-hardness of the Skolem Problem. We have moreover shown that for instances of the Skolem Problem generated in our reduction, counting zeros is #P-complete. Finally, we have introduced the LRSInclusion Problem, which serves as a generalization of both the Skolem and Positivity Problems, and established its Π_2^{P}-hardness.

We conclude by listing a few important open problems for further research:

1. The Skolem Problem is not known to be decidable, yet our best lower bound for it is NP-hardness; can this huge gap be narrowed?
2. The NP-hardness of Skolem is witnessed in LRS of high order; can one improve by showing NP-hardness for LRS of constant order, as even decidability of Skolem at order 5 remains open?
3. Given an LRS $\boldsymbol{u}$ and $n \in \mathbb{N}$ in binary, what is the complexity of determining whether $u_n = 0$? This problem can be reduced to EquSLP, implying a polynomial-time randomized algorithm for it. Can one obtain a deterministic alternative? This in turn would likely lead to a better understanding of the EquSLP Problem.
4. We showed that #Skolem$_\omega$ is #P-complete. Is #Skolem also #P-complete?

References

1 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Ann. Math. (2)*, 160(2):781–793, 2004. doi:10.4007/annals.2004.160.781.

2 W. R. Alford, Andrew Granville, and Carl Pomerance. There are infinitely many Carmichael numbers. *Annals of Mathematics*, 139(3):703–722, 1994. URL: http://www.jstor.org/stable/2118576.

3 Eric Allender, Peter Bürgisser, Johan Kjeldgaard-Pedersen, and Peter Bro Miltersen. On the complexity of numerical analysis. *SIAM Journal on Computing*, 38(5):1987–2006, 2009.

4 S. Almagor, T. Karimov, E. Kelmendi, J. Ouaknine, and J. Worrell. Deciding ω-regular properties on linear recurrence sequences. *Proc. ACM Program. Lang.*, 5(POPL), 2021.

5 Francesco Amoroso and Evelina Viada. On the zeros of linear recurrence sequences. *Acta Arithmetica*, 147(4):387–396, 2011.

6 Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach.* Cambridge University Press, USA, 1st edition, 2009.

7 Piotr Berman, Marek Karpinski, Lawrence L. Larmore, Wojciech Plandowski, and Wojciech Rytter. On the complexity of pattern matching for highly compressed two-dimensional texts. *Journal of Computer and System Sciences*, 65(2):332–350, 2002. URL: https://www.sciencedirect.com/science/article/pii/S0022000002918520, doi:https://doi.org/10.1006/jcss.2002.1852.

8 Yuri Bilu, Florian Luca, Joris Nieuwveld, Joël Ouaknine, David Purser, and James Worrell. Skolem Meets Schanuel. In Stefan Szeider, Robert Ganian, and Alexandra Silva, editors, *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, volume 241 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:15, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2022/16818, doi:10.4230/LIPIcs.MFCS.2022.20.

9 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, August 2002. doi:10.1016/s0024-3795(01)00466-9.

10 Qi Cheng, Joshua E. Hill, and Daqing Wan. Counting value sets: Algorithm and complexity, 2011. URL: https://arxiv.org/abs/1111.1224, doi:10.48550/ARXIV.1111.1224.

11 Ventsislav Chonev, Joël Ouaknine, and James Worrell. On the complexity of the orbit problem. *J. ACM*, 63(3), jun 2016. doi:10.1145/2857050.

12 Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

13 Graham Everest, Alf van der Poorten, Igor Shparlinski, and Thomas Ward. *Recurrence Sequences*. American Mathematical Society, July 2003. doi:10.1090/surv/104.

14 Vesa Halava, Tero Harju, Mika Hirvensalo, and Juhani Karhumäki. Skolem's problem - on the border between decidability and undecidability. Technical Report 683, 2005.

15 Michael Krüger. *On the Complexity of Alternative Solutions*. PhD thesis, Friedrich-Schiller-University Jena, Germany, 2008.

16 Noam Livne. A note on #P-completeness of NP-witnessing relations. *Inf. Process. Lett.*, 109(5):259–261, feb 2009. doi:10.1016/j.ipl.2008.10.009.

17 M. Mignotte, T. N. Shorey, and R. Tijdeman. The distance between terms of an algebraic recurrence sequence. *Journal für die reine und angewandte Mathematik*, 349, 1984.

18 Joël Ouaknine and James Worrell. *Positivity Problems for Low-Order Linear Recurrence Sequences*, pages 366–379. 2014. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611973402.27, arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611973402.27, doi:10.1137/1.9781611973402.27.

19 J. Barkley Rosser and Lowell Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois Journal of Mathematics*, 6(1):64 – 94, 1962. doi:10.1215/ijm/1255631807.

20 Akshay S., Nikhil Balaji, and Nikhil Vyas. Complexity of Restricted Variants of Skolem and Related Problems. In Kim G. Larsen, Hans L. Bodlaender, and Jean-Francois Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 78:1–78:14, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: http://drops.dagstuhl.de/opus/volltexte/2017/8130, doi:10.4230/LIPIcs.MFCS.2017.78.

21 Marcus Schaefer and Christopher Umans. Completeness in the polynomial-time hierarchy a compendium. *Sigact News - SIGACT*, 33, 01 2002.

22 N. K. Vereshchagin. Occurrence of zero in a linear recursive sequence. *Mathematical Notes of the Academy of Sciences of the USSR*, 38(2):609–615, August 1985. doi:10.1007/bf01156238.

23 J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2013. URL: https://books.google.co.in/books?id=AE5PN5QGgvUC.

24 Joachim von zur Gathen, Marek Karpinski, and Igor E. Shparlinski. Counting curves and their projections. *Comput. Complex.*, 6(1):64–99, 1997. doi:10.1007/BF01202042.