International Journal of Algebra and Computation


# Knapsack and the power word problem in solvable Baumslag-Solitar groups

Moses Ganardi
*Max Planck Institute for Software Systems (MPI-SWS), Germany*
*ganardi@mpi-sws.org*

Markus Lohrey
*Universität Siegen, Germany*
*lohrey@eti.uni-siegen.de*

Georg Zetzsche
*Max Planck Institute for Software Systems (MPI-SWS), Germany*
*georg@mpi-sws.org*

Received (Day Month Year)
Accepted (Day Month Year)

Communicated by [editor]

We prove that the power word problem for certain metabelian subgroups of $\mathrm{GL}(2,\mathbb{C})$ (including the solvable Baumslag-Solitar groups $\mathrm{BS}(1,q) = \langle a,t \mid tat^{-1} = a^q\rangle$) belongs to the circuit complexity class $\mathsf{TC}^0$. In the power word problem, the input consists of group elements $g_1,\dots,g_d$ and binary encoded integers $n_1,\dots,n_d$ and it is asked whether $g_1^{n_1}\cdots g_d^{n_d} = 1$ holds. Moreover, we prove that the knapsack problem for $\mathrm{BS}(1,q)$ is $\mathsf{NP}$-complete. In the knapsack problem, the input consists of group elements $g_1,\dots,g_d,h$ and it is asked whether the equation $g_1^{x_1}\cdots g_d^{x_d} = h$ has a solution in $\mathbb{N}^d$. For the more general case of a system of so-called exponent equations, where the exponent variables $x_i$ can occur multiple times, we show that solvability is undecidable for $\mathrm{BS}(1,q)$.

*Keywords*: computational group theory, matrix problems, Baumslag-Solitar groups

Mathematics Subject Classification 2010: 20F10; 68Q06

## 1. Introduction

### 1.1. *The power word problem*

The study of multiplicative identities and equations has a long tradition in computational algebra, and has recently been extended to the non-abelian case. Here, the multiplicative identities we have in mind have the form $g_1^{n_1}g_2^{n_2}\cdots g_d^{n_d} = 1$, where $g_1,\dots,g_d$ are elements of a group $G$ and $n_1,n_2,\dots,n_d \in \mathbb{N}$ are non-negative

integers[a]. Typically, the numbers $n_i$ are given in binary representation, whereas the representation of the group elements $g_i$ depends on the underlying group $G$. Here, we consider the case where $G$ is a finitely generated (f.g. for short) group, and elements of $G$ are represented by finite words over a fixed generating set $\Sigma$ (the concrete choice of $\Sigma$ is not relevant). In this setting, the problem of deciding whether $g_1^{n_1} g_2^{n_2} \cdots g_d^{n_d} = 1$ is a true identity has recently been introduced as the *power word problem* for $G$ [35]. It extends the classical word problem for $G$ (does a given word over the group generators represent the group identity?) in the sense that the word problem trivially reduces to the power word problem (take an identity $w^1 = 1$). Recent results on the power word problem in specific f.g. groups are:

- For every f.g. free group the power word problem belongs to deterministic logspace [35]. This result has recently been generalized in [42], where it is shown that the power word problem in a fixed graph product of groups is logspace-reducible (even $\mathsf{AC}^0$-Turing-reducible) to the word problem of the free group of rank two and the power word problem of the base groups of the graph product.
- For the following groups the power word problem belongs to the circuit complexity class $\mathsf{TC}^0$:[b] f.g. nilpotent groups [35], iterated wreath products of f.g. free abelian groups and (as a consequence of the latter) free solvable groups [16].
- If $G$ is a so-called uniformly efficiently non-solvable group (this is a large class of non-solvable groups that was recently introduced in [4] and that includes all finite non-solvable groups and f.g. free non-abelian groups) then the power word problem for the wreath product $G \wr \mathbb{Z}$ is $\mathsf{coNP}$-hard [16]. As a consequence, the power word problem for Thompson's group $F$ is $\mathsf{coNP}$-complete [16].

Historically, the power word problem appeared earlier in the area of computational (commutative) algebra. Ge [21] proved that one can check in polynomial time whether an identity $\alpha_1^{n_1} \alpha_2^{n_2} \cdots \alpha_d^{n_d} = 1$ holds, where the $n_i$ are binary encoded integers and the $\alpha_i$ are drawn from an algebraic number field (and suitably encoded).

In this paper we investigate the power word problem for certain 2-generated subgroups of $\mathrm{GL}(2, \mathbb{C})$: for a fixed complex number $\alpha \in \mathbb{C} \setminus \{0\}$ we consider the group $T(\alpha)$ generated by the two matrices

$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix}.$$

[a]One could also allow negative $n_i$. This would not make a difference, since we can replace a $g_i$ by its inverse $g_i^{-1}$

[b]$\mathsf{TC}^0$ is a very small complexity class within polynomial time; see Section 2.1 for more details. In this paper, $\mathsf{TC}^0$ always refers to the $\mathsf{DLOGTIME}$-uniform version.

The group $T(\alpha)$ is a semi-direct product $\mathbb{Z}[\alpha] \rtimes \mathbb{Z}$, where $\mathbb{Z}$ acts by multiplication with $\alpha$ on $\mathbb{Z}[\alpha]$. In a purely group theoretic context, the groups $T(\alpha)$ were studied in [23,24]. For the special that $\alpha$ satisfies a quadratic equation $\alpha^2 - c\alpha + 1$ for an integer $c \geq 3$, the groups $T(\alpha)$ turned out to be the main obstacles for solving the semigroup membership problem for the special affine group $\mathsf{SA}(2, \mathbb{Z})$ [13]. Other important special cases of $T(\alpha)$ are the wreath product $\mathbb{Z} \wr \mathbb{Z}$ (for $\alpha$ transcendental) and the solvable Baumslag-Solitar groups $\mathrm{BS}(1, q) = \langle a, t \mid tat^{-1} = a^q \rangle$ (for $\alpha = q \geq 2$ an integer). Our first main result states that the power word problem for every group $T(\alpha)$ belongs to $\mathsf{TC}^0$ (Theorem 3.1). For the word problem of $T(\alpha)$, membership in $\mathsf{TC}^0$ follows from [28][c] since $T(\alpha)$ is a linear solvable group. Theorem 3.1 is directly related to recent results on matrix powering problems [1,19]. These problems can be quite difficult to analyze. For instance, it is not known whether a certain bit of the $(1,1)$-entry of a matrix power $A^n$ can be computed in polynomial time, when $n$ is given in binary notation and $A$ is a $(2 \times 2)$-matrix over $\mathbb{Z}$. The related problem of checking whether the $(1,1)$-entry (or any other entry) of $A^n$ is positive can be solved in polynomial time by [19].

### 1.2. *The knapsack problem*

If in the power word problem, one replaces the exponents $n_i$ by pairwise distinct variables $x_i$ and the right-hand side 1 by an arbitrary group element $h \in G$, one obtains a so-called knapsack equation $g_1^{x_1} g_2^{x_2} \cdots g_d^{x_d} = h$. The problem of deciding whether such an equation has a solution in $\mathbb{N}^d$ is known as the *knapsack problem* for $G$. In the general context of finitely generated groups the knapsack problem has been introduced by Myasnikov, Nikolaev, and Ushakov [40]. As for the power word problem, this problem has been studied in the commutative setting before. For the case $G = \mathbb{Z}$ one obtains a variant of the classical $\mathsf{NP}$-complete knapsack problem; a proof of the $\mathsf{NP}$-hardness of our variant of the knapsack problem for the integers can be found in [25]. For this hardness result it is important that integers are represented in binary notation. For unary encoded integers the complexity of the knapsack problem goes down to $\mathsf{TC}^0$. For the case that the $g_i$ are commuting matrices over an algebraic number field, the knapsack problem has been studied in [3,12].

For the case of (in general) non-commutative groups, the knapsack problem has been studied in [14,16,18,20,29,34,36,40]. In these papers, group elements are usually represented by finite words over the generators (although in [36] a more succinct representation by so-called straight-line programs is studied as well). Note that for the group $\mathbb{Z}$ this corresponds to a unary representation of integers. Hyperbolic groups, which are of fundamental importance in the area of geometric group theory, are an important class of groups where knapsack can be decided in polynomial time (even in $\mathsf{LogCFL}$, i.e., the closure of the context-free languages under logspace

[c]For the special case $\mathrm{BS}(1, q)$ membership of the word problem in $\mathsf{TC}^0$ was shown in [44].

reductions). This result can be extended to the class of all groups that can be built from hyperbolic groups by the operations of (i) direct products with $\mathbb{Z}$ and (ii) free products [36]. On the other hand, for many groups the knapsack problem is NP-complete. Examples are certain right-angled Artin groups (like the direct product of two free groups of rank two [36]), wreath products (e.g. the wreath product $\mathbb{Z} \wr \mathbb{Z}$ [20]) and free solvable groups [16]. For wreath products $G \wr \mathbb{Z}$, where $G$ is finite non-solvable or free of rank at least two, the knapsack problem is complete for $\Sigma_2^p$ (the second existential level of the polynomial time hierarchy) [16]. Finally, for finitely generated nilpotent groups, the knapsack problem is in general undecidable [20,39].

Our second main result is that for the Baumslag-Solitar groups $\mathrm{BS}(1, q)$ with $q \geq 2$, the knapsack problem is NP-complete (Theorem 4.1). This extends a result from [14], where decidability (without any complexity bound) was shown for a restriction of the knapsack problem for $\mathrm{BS}(1, q)$. In this restriction, all group elements $g_i$ must be represented by words where the exponent sum of all occurrences of $t$ is zero (here we refer to the presentation $\langle a, t \mid tat^{-1} = a^q \rangle$ of $\mathrm{BS}(1, q)$). Showing NP-hardness of the knapsack problem for $\mathrm{BS}(1, q)$ is easy (based on the result that knapsack for $\mathbb{Z}$ with binary encoded integers is NP-hard). For membership in NP we use a recent result of Guépin, Haase, and Worrell [22] according to which the existential fragment of Büchi arithmetic (an extension of Presburger arithmetic) belongs to NP. The NP-membership of the knapsack problem for $\mathrm{BS}(1, q)$ is a bit of a surprise, since one can show that minimal solutions of knapsack equations over $\mathrm{BS}(1, q)$ can be of size doubly exponential in the length of the equation, see Theorem 4.2. This rules out a simple guess-and-verify strategy.

### 1.3. *Solvability of systems of exponent equations.*

In the final section of the paper we consider the following generalization of the knapsack problem for $\mathrm{BS}(1, q)$: the input is a conjunction

$$\bigwedge_{i=1}^{n} g_{i1}^{x_{i1}} g_{i2}^{x_{i2}} \cdots g_{id_i}^{x_{id_i}} = h_i, \tag{1.1}$$

where the $g_{ij}, h_i$ are elements of $\mathrm{BS}(1, q)$ and the $x_{ij}$ are variables taking values in $\mathbb{N}$. In contrast to the knapsack problem, we do not require these variables to be pairwise different: We also allow $x_{ij} = x_{ik}$. Note that since $x_{ij}$, $x_{ik}$ are variables, if $x_{ij} = x_{ik}$, then a solution *must* put the same exponent at $g_{ij}$ and $g_{ik}$. (This is not to be confused with allowing the two exponents to coincide; this is always allowed.)

We call (1.1) a system of exponent equations. Aside from being a natural generalization of the knapsack problem, systems of exponent equations play a crucial role in a characterization of decidability of the knapsack problem for wreath products [6]: In order to understand for which wreath products $G \wr H$ the knapsack problem is decidable, we need to clarify for which groups $G$ one can decide solvability of systems of exponent equations. For example, the knapsack problem is decidable for

$\mathbb{Z} \wr G$ if and only if solvability of systems of exponent equations is decidable for $G$ (this special case already follows from [20, Proposition 3.1, Theorem 5.3]).

For many groups, solvability of systems of exponent equations is decidable. This holds in fact for all so-called knapsack semilinear groups, i.e., groups where the set of solutions of a knapsack equation is an effectively computable semilinear set. Examples of knapsack semilinear groups are hyperbolic groups [34] and co-context-free groups [29]. Moreover, the class of knapsack semilinear groups is effectively closed under finite extensions [17], wreath products [20], graph products [17], and amalgamated products and HNN-extensions over finite groups [17]. On the other hand, solvability of systems of exponent equations is undecidable for the discrete Heisenberg group [29].

Our last main result states that solvability of systems of exponent equations is undecidable for every Baumslag-Solitar group $\mathrm{BS}(1, q)$ with $q \geq 2$ (Theorem 5.1). We prove this result by a reduction from the existential theory of $(\mathbb{N}, +, (x, y) \mapsto x \cdot 2^y)$, which was shown to be undecidable by Büchi and Senger [11, Corollary 5].

A preliminary version of this paper appeared in [37].

## 2. Preliminaries

For $a, b \in \mathbb{Z}$ we write $a \mid b$ if $b = ka$ for some $k \in \mathbb{Z}$ (in other words, $a$ divides $b$). We denote with $[a, b]$ the interval $\{z \in \mathbb{Z} \mid a \leq z \leq b\}$. For complex numbers $\alpha_1, \ldots, \alpha_k \in \mathbb{C}$ we denote with $\mathbb{Z}[\alpha_1, \ldots, \alpha_k]$ the subring of $\mathbb{C}$ obtained by adjoining to the ring of integers $\mathbb{Z}$ the complex numbers $\alpha_1, \ldots, \alpha_k$.

The set of polynomials with variable $x$ and coefficients from $\mathbb{Z}$ is denoted with $\mathbb{Z}[x]$. Let $p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \in \mathbb{Z}[x]$ with $a_n \neq 0$. Then we define $\deg(p) = n$ (the *degree* of $p(x)$) and $\mathrm{height}(p) = \max\{|a_0|, \ldots, |a_n|\}$. The *dense representation* of the above polynomial is the tuple $(a_0, a_1, \ldots, a_n)$, where every $a_i$ is given in binary encoding. We define the *sparse representation* of a polynomial $p(x) = a_0 x^{e_0} + a_1 x^{e_1} + \cdots + a_n x^{e_n}$ with $a_i \in \mathbb{Z} \setminus \{0\}$ for all $0 \in [0, n]$ and $0 \leq e_0 < e_1 < \cdots < e_n$ as the list $(a_0, e_0, a_1, e_1, \ldots, a_n, e_n)$ where all numbers in this list are written in binary representation.

A *Laurent polynomial* is a polynomial that may also contain powers $x^k$ with $k < 0$. Formally, a Laurent polynomial over $\mathbb{Z}$ is an expression $p(x) = \sum_{i \in \mathbb{Z}} a_i x^i$ with $a_i \in \mathbb{Z}$ such that only finitely many $a_i$ are non-zero. With $\mathbb{Z}[x, x^{-1}]$ we denote the set of all Laurent polynomials over $\mathbb{Z}$; it is a ring with the natural addition and multiplication operations. If $p(x) = \sum_{i=k}^{l} a_i x^i$ with $k, l \in \mathbb{Z}$, $k \leq l$ and $a_k \neq 0 \neq a_l$ then we define the *dense unary* (resp., *dense binary*) *representation* of the Laurent polynomial $P(x)$ as the list of unary (resp., binary) encoded integers $a_k, a_{k+1}, \ldots, a_l$ together with $k$ in unary encoding.

For a complex number $\alpha \in \mathbb{C} \setminus \{0\}$ we have a natural homomorphism from $\mathbb{Z}[x, x^{-1}]$ to $\mathbb{Z}[\alpha, \alpha^{-1}]$ obtained by evaluating a Laurent polynomial at $x = \alpha$. Clearly, for an integer $q \in \mathbb{Z} \setminus \{0\}$ we have $\mathbb{Z}[q, q^{-1}] = \mathbb{Z}[1/q]$. If $q \geq 2$, this is the set of all rational numbers that have finite expansion in base $q$, i.e., the set of all

numbers $\sum_{a\leq i\leq b} r_i q^i$ with $r_i \in [0, q-1]$ and $a, b \in \mathbb{Z}$. If $u = \sum_{-k\leq i\leq \ell} r_i q^i \neq 0$ with $k, \ell \geq 0$ and $\ell + k$ minimal, we define $\|u\|_q = \ell + k + 1$. Under the assumption that $q$ is a constant (which will be always the case in this paper), $\|u\|_q$ is the number of digits in the $q$-ary representation of $u$.

### 2.1. *Circuit complexity*

We assume basic knowledge in complexity theory, in particular with the complexity class $\mathsf{NP}$; see [2] for details. We deal with the circuit complexity class $\mathsf{TC}^0$. It contains all languages $L \subseteq \{0,1\}^*$ that can be solved by a family of threshold circuits of polynomial size and constant depth. More formally: we have a family $\mathcal{C} = (C_n)_{n\geq 0}$ of boolean circuits $C_n$ with the following properties:

- $C_n$ has exactly $n$ input gates $x_1, \ldots, x_n$ with fan-in zero (the fan-in of a gate is the number of incoming wires).
- All other gates are either not-gates (with fan-in one), and-gates (with arbitrary fan-in), or majority-gates (with arbitrary fan-in). A majority gate of fan-in $k$ evaluates to 1 if and only if at least $k/2$ many input wires carry the truth value 1.
- Every $C_n$ has a distinguished output gate.
- There is a constant $d$ such that the depth of very circuit $C_n$ is bounded by $d$, where the depth of a circuit is the length of a longest path from an input gate to the output gate.
- There is a polynomial $p(n)$ such that $C_n$ has at most $p(n)$ many gates.
- For every word $w = a_1 a_2 \cdots a_n$ with $a_i \in \{0,1\}$, we have $w \in L$ if and only if the output gate of the circuit $C_n$ evaluates to 1 when every input gate $x_i$ is set to $a_i$.

We can lift this definition to languages over an arbitrary alphabet $\Sigma$ by fixing a binary encoding of the symbols from $\Sigma$. We always assume such encodings implicitly. To compute a function $f\colon \{0,1\}^* \to \{0,1\}^*$ by a circuit family, we encode $f$ by the language $L_f = \{1^i 0 w \mid w \in \{0,1\}^*, \text{the } i\text{-th bit of } f(w) \text{ is } 1\}$.

In this paper, we only deal with the $\mathsf{DLOGTIME}$-uniform version of $\mathsf{TC}^0$. In this variant, $\mathsf{TC}^0$ is contained in deterministic logspace and hence in polynomial time. We do not give the quite technical definition of $\mathsf{DLOGTIME}$-uniformity; see [43] for details. In fact, all we need about $\mathsf{TC}^0$ is the fact that the following problems belong to $\mathsf{DLOGTIME}$-uniform $\mathsf{TC}^0$:

(1) iterated addition/multiplication (i.e., addition/multiplication of an arbitrary number) of binary encoded numbers/polynomials that are given in dense representation [15,26],
(2) division with remainder of two polynomials that are given in dense representation [15,26] (in particular, of two binary encoded numbers),
(3) computing the number $|w|_a$ of occurrences of a letter $a$ in a word $w$,

(4) computing an image $h(w)$ where $h : \Sigma^* \to \Gamma^*$ is a homomorphism of free monoids [31],
(5) computing the minimum of a given list of integers.

The results on binary numbers hold for any base, since one can transform between binary representation and $q$-ary representation; this is a consequence of the first two points.

In the rest of the paper, when we speak about $\mathsf{TC}^0$, we always refer to $\mathsf{DLOGTIME}$-uniform $\mathsf{TC}^0$.

### 2.2. *Algebraic numbers*

An *algebraic number* is a complex number that is the root of some non-zero polynomial from $\mathbb{Z}[x]$. For every algebraic number $\alpha$ there is a unique polynomial $p(x) \in \mathbb{Z}[x]$ with $p(\alpha) = 0$ and such that $p(x)$ has minimal degree among all such polynomials, the leading coefficient of $p$ is non-negative, and the coefficients of $p(x)$ have no common divisor $> 1$. This polynomial is called the *minimal polynomial*[d] of $\alpha$. If $p(x)$ is the minimal polynomial of $\alpha$, then we define $\deg(\alpha) = \deg(p)$ and $\text{height}(\alpha) = \text{height}(p)$.

A *root of unity* is an algebraic number $\alpha$ such that $\alpha^m = 1$ for some $m \geq 1$. In other words: the minimal polynomial of $\alpha$ divides the polynomial $x^m - 1$ (and hence all roots of the minimal polynomial are roots of unity). If $\alpha^m = 1$ but there is no $1 \leq n < m$ with $\alpha^n = 1$ then $\alpha$ is an $m^{th}$ *primitive root of unity*. Note that if $\alpha$ is an $m^{th}$ primitive root of unity then $\alpha^k = 1$ if and only if $m$ divides $k$.

We need the following lemma:

**Lemma 2.1.** *Given an irreducible polynomial $p(x) \in \mathbb{Z}[x]$ in dense representation, we can check in $\mathsf{TC}^0$ whether $p(x)$ divides some $x^m - 1$ for $m \geq 1$.*[e] *Moreover, if $p(x)$ divides $x^m - 1$ for some $m \geq 1$, then we can compute in $\mathsf{TC}^0$ the smallest such $m$ in unary encoding.*

This lemma uses an algorithm of Bradford and Davenport [7], which checks a polynomial number of values for $m$. They bound the number of candidates by bounding $m$ in terms of $\varphi(m)$, where $\varphi$ is Euler's phi-function. This suffices because of the following well-known number theory fact (we include a short proof since [7] does not provide a reference).

**Lemma 2.2.** *If $p \in \mathbb{Q}[x]$ is irreducible and divides $x^m - 1$ for some $m \geq 1$ and $m$ is minimal with this property, then* $\deg(p) = \varphi(m)$.

[d]This is slightly non-standard terminology: It is a basic fact that the minimal-degree non-zero polynomials in $\mathbb{Q}[x]$ that vanish at $\alpha$ only differ by factors in $\mathbb{Q} \setminus \{0\}$. Usually, the minimal polynomial is made unique by requiring the leading coefficient to be 1. For us, it is more convenient to obtain a minimal polynomial in $\mathbb{Z}[x]$ by normalizing as above.
[e]We are not aware of a $\mathsf{TC}^0$-algorithm for testing whether a given polynomial is irreducible. Hence, the problem formulated in the lemma is a promise problem.

**Proof.** The irreducible factors of $x^m - 1$ are the cyclotomic polynomials $\Phi_d(x)$ for all divisors $d$ of $m$ with $1 \leq d \leq m$ [30, p. 279]. Moreover, $\deg(\Phi_d) = \varphi(m)$ [30, p. 279]. Since $p(x)$ divides $x^m - 1$, we can write $p(x) = \alpha \cdot \Phi_d(x)$ for some non-zero $\alpha \in \mathbb{Q}$ and some $1 \leq d \leq m$. If $d < m$ then $p(x)$ would divide $x^d - 1$, which is a contradiction. We obtain $\deg(p) = \deg(\Phi_m) = \varphi(m)$. □

We can now describe the algorithm of Bradford and Davenport.

**Proof of Lemma 2.1.** Let $d$ be the degree of $p$. If $d = 1$ then we only have to check whether $p(x) = x \pm 1$. Hence, assume that $d \geq 2$. If $p(x)$ divides $x^m - 1$ with $m \geq 1$ and $m$ is minimal with this property then Lemma 2.2 yields $d = \varphi(m)$. By [7, Corollary, p. 247], we have $m \leq 3\varphi(m)^{3/2}$ and thus $d < m \leq 3d^{3/2}$. Hence, given the irreducible polynomial $p(x)$ of degree $d$, we simply test in parallel for every $m \in \{d+1, \ldots, 3d^{3/2}\}$ whether $p(x)$ divides $x^m - 1$ (this is possible in $\mathsf{TC}^0$). If there is no such $m$ then there is no $n \geq 1$ such that $p(x)$ divides $x^n - 1$. Otherwise, we compute in $\mathsf{TC}^0$ the unary encoding of the smallest $m$ such that $d < m \leq 3d^{3/2}$ and $p(x)$ divides $x^m - 1$. □

*Sparse polynomial root testing* is the following decision problem:

**Input** A polynomial $P(x) \in \mathbb{Z}[x]$ given in sparse representation and an algebraic number $\alpha \in \mathbb{C}$ given by its minimal polynomial in dense representation.[f]
**Question** Is $P(\alpha) = 0$?

Note that we do not specify $\alpha$ uniquely: if $p_\alpha(x)$ is the minimal polynomial of $\alpha$ then by writing down only $p_\alpha(x)$, we cannot distinguish $\alpha$ from its conjugates. On the other hand, for sparse polynomial root testing there is no reason to make this distinction, because $P(\alpha) = 0$ if and only if $p_\alpha(x)$ divides $P(x)$.

**Theorem 2.3.** *Sparse polynomial root testing is in* $\mathsf{TC}^0$.

**Proof.** Let $p_\alpha(x)$ be the minimal polynomial of $\alpha$, which is part of the input in dense representation. Using Lemma 2.1 we first check whether $\alpha$ is a root of unity and in the positive case we compute in $\mathsf{TC}^0$ the unary encoding of the number $m$ such that $\alpha$ is an $m^{\text{th}}$ primitive root of unity.

Let us now first consider the case that $\alpha$ is not a root of unity. For this case, it was shown in [32, Proposition 2.3] that sparse polynomial root testing belongs to polynomial time using the following gap theorem: Let $P(x) = P_0(x) + x^s P_1(x) \in \mathbb{Z}[x]$ be a polynomial with $k+1$ monomials and $u = \deg(P_0)$ and let $d \geq 1$ be an integer such that

$$s - u > \frac{\ln k + \ln \text{height}(P)}{c(d)} \tag{2.1}$$

[f]By the previous footnote, also sparse polynomial root testing is a promise problem.

where

$$c(1) = \ln 2 \quad \text{and} \quad c(d) = \frac{2}{d \cdot (\ln(3d))^3} \text{ for } d \geq 2.$$

If $\alpha$ is an algebraic number of degree at most $d$ which is not a root of unity then $P(\alpha) = 0$ if and only if $P_0(\alpha) = 0$ and $P_1(\alpha) = 0$.

Note that the number on the right-hand side of (2.1) is polynomial in the input length if $P$ is given in spare representation and the minimal polynomial of $\alpha$ is given in dense representation. This allows to split in $\mathsf{TC}^0$ the input polynomial $P(x)$ into several polynomials $p_0(x), \ldots, p_k(x)$ such that $P(\alpha) = 0$ if and only if $p_i(\alpha) = 0$ for all $1 \leq i \leq k$. Moreover, all $p_i$ are computed in dense representation. Finally, we check for every $i$ whether $p_\alpha(x)$ divides $p_i(x)$.

It remains to consider the case where $\alpha$ is an $m^{\text{th}}$ primitive root of unity. We have computed the unary encoding of $m$. We can then replace in the polynomial $P(x)$ every binary encoded monomial $x^n$ by $x^{n \bmod m}$. In this way we can compute a polynomial $\tilde{P}(x)$ in dense representation such that $\tilde{P}(\alpha) = 0$ if and only if $P(\alpha) = 0$. Finally, we check in $\mathsf{TC}^0$ whether $p_\alpha(x)$ divides $\tilde{P}(x)$. □

### 2.3. *Groups*

We assume that the reader is familiar with the basics of group theory. Let $G$ be a group. We always write 1 for the group identity element. We say that $G$ is *finitely generated (f.g.)* if there is a finite subset $\Sigma \subseteq G$ such that every element of $G$ can be written as a product of elements from $\Sigma$; such a $\Sigma$ is called a *(finite) generating set* for $G$. We always assume that $a \in \Sigma$ implies $a^{-1} \in \Sigma$; such a generating set is also called *symmetric*. We write $G = \langle \Sigma \rangle$ if $\Sigma$ is a symmetric generating set for $G$. In this case, we have a canonical surjective morphism $h : \Sigma^* \to G$ that maps a word over $\Sigma$ to its product in $G$ (the so called *evaluation morphism*). If $h(w) = 1$ we also say that $w = 1$ *in* $G$. On $\Sigma^*$ we can define a natural involution $\cdot^{-1}$ by $(a_1 a_2 \cdots a_n)^{-1} = a_n^{-1} \cdots a_2^{-1} a_1^{-1}$ for $a_1, a_2, \ldots, a_n \in \Sigma$.

#### 2.3.1. *Matrix groups*

For a complex number $\alpha \in \mathbb{C} \setminus \{0\}$ let $T(\alpha)$ be the subgroup of $\mathrm{GL}(2, \mathbb{C})$ consisting of the upper triangular matrices

$$\begin{pmatrix} \alpha^k & u \\ 0 & 1 \end{pmatrix} \tag{2.2}$$

with $k \in \mathbb{Z}$ and $u \in \mathbb{Z}[\alpha, \alpha^{-1}]$. This means we have the multiplication

$$\begin{pmatrix} \alpha^k & u \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha^\ell & v \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha^{k+\ell} & u + \alpha^k \cdot v \\ 0 & 1 \end{pmatrix}. \tag{2.3}$$

This group can be also written as the semi-direct product $\mathbb{Z}[\alpha, \alpha^{-1}] \rtimes \mathbb{Z}$, where $\mathbb{Z}$ acts on $\mathbb{Z}[\alpha, \alpha^{-1}]$ by multiplication with $\alpha$. The groups $T(\alpha)$ are also studied in [23,24].

We encode the matrix (2.2) by the pair $(k, p)$, where $k$ is given in unary encoding and $p$ is a Laurent polynomial with $u = p(\alpha)$ that is given in dense unary representation. The group $T(\alpha)$ is generated by the two matrices

$$a = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad t = \begin{pmatrix} \alpha & 0 \\ 0 & 1 \end{pmatrix} \tag{2.4}$$

and their inverses. We denote with $h\colon \{a, a^{-1}, t, t^{-1}\}^* \to T(\alpha)$ the canonical evaluation morphism. Hence, $h(w)$ is the identity matrix if and only if $w = 1$ in $T(\alpha)$.

We now have two encodings of elements from $T(\alpha)$: as pairs $(k, p)$ describing a matrix (2.2) and as words over the alphabet $\{a, a^{-1}, t, t^{-1}\}$. By the the following lemma, we can switch in $\mathsf{TC}^0$ between these encodings.

**Lemma 2.4.** *Given a word $w \in \{a, a^{-1}, t, t^{-1}\}^*$ we can compute in $\mathsf{TC}^0$ the matrix $h(w)$ encoded as a pair $(k, p)$ as above. Vice versa, given a matrix $A \in T(\alpha)$ in the above encoding, we can compute in $\mathsf{TC}^0$ a word $w \in h^{-1}(A)$.*

**Proof.** First consider a word $w \in \{a, a^{-1}, t, t^{-1}\}^*$ and let $h(w)$ be the matrix in (2.2). Then $k = |w|_t - |w|_{t^{-1}}$, which can be computed in $\mathsf{TC}^0$. It remains to compute a Laurent polynomial $p(x)$ in dense unary representation such that $u = p(\alpha)$. Let $w_1 a^{\epsilon_1}, \ldots, w_l a^{\epsilon_l}$ be all prefixes of $w$ that end with $a$ or $a^{-1}$ ($\epsilon_1, \ldots, \epsilon_l \in \{-1, 1\}$). Let $k_i = |w_i|_t - |w_i|_{t^{-1}}$, which can be computed in $\mathsf{TC}^0$ in unary notation. Then, $u = p(\alpha)$ with $p(x) = \sum_{i=1}^{l} \epsilon_i x^{k_i}$. The dense unary representation of this polynomial can be easily computed in $\mathsf{TC}^0$.

The inverse transformation is straightforward: take the matrix (2.2), where $k$ is given in unary encoding and $u = p(x)$ for a Laurent polynomial $p(x)$ in dense unary representation. A matrix of the form $\left(\begin{smallmatrix} 1 & \alpha^z \\ 0 & 1 \end{smallmatrix}\right)$ (for a unary encoded $z$) can be produced by the word $t^z a t^{-z}$. By concatenating such words (which is possible in $\mathsf{TC}^0$ by point 4 from page 6), one can produce from a given Laurent polynomial $p(x)$ in dense unary representation a word for the matrix $\left(\begin{smallmatrix} 1 & p(\alpha) \\ 0 & 1 \end{smallmatrix}\right)$. Finally, one has to concatenate $t^k$ on the right in order to produce the matrix (2.2). □

### 2.3.2. *Baumslag-Solitar groups*

For $p, q \in \mathbb{Z} \setminus \{0\}$, the *Baumslag-Solitar group* $\mathrm{BS}(p, q)$ is defined as the finitely presented group $\mathrm{BS}(p, q) = \langle a, t \mid ta^p t^{-1} = a^q \rangle$. We can w.l.o.g. assume that $q \geq 1$. Of particular interest are the Baumslag-Solitar groups $\mathrm{BS}(1, q)$ for $q \geq 2$. They are solvable and linear. It is well-known (see e.g. [46, III.15.C]) that $\mathrm{BS}(1, q)$ is isomorphic to $T(q)$. Moreover, the generator $a$ (resp., $t$) of $\mathrm{BS}(1, q)$ corresponds to the matrix $a$ (resp., $t$) from (2.4). From Lemma 2.4 we immediately get:

**Lemma 2.5.** *Given a word $w \in \{a, a^{-1}, t, t^{-1}\}^*$ we can compute in $\mathsf{TC}^0$ the matrix $h(w)$ with matrix entries given in $q$-ary encoding. Vice versa, given a matrix $A \in T(q)$ with $q$-ary encoded entries, we can compute in $\mathsf{TC}^0$ a word $w \in h^{-1}(A)$.*

By the previous lemma, we can represent elements of $\mathrm{BS}(1,q)$ either as words over the alphabet $\{a, a^{-1}, t, t^{-1}\}$ or by matrices from $T(q)$ with $q$-ary encoded entries. For the matrix $A \in T(q)$ in (2.2) (with $\alpha = q$) we define $\|A\| = |k| + \|u\|_q$. Hence $\|A\|$ is the length of the encoding of $A$.

Another well known special case of the group $T(\alpha)$ is obtained when $\alpha$ is transcendental. In this case $T(\alpha)$ is isomorphic to the wreath product $\mathbb{Z} \wr \mathbb{Z}$: It is isomorphic to the group of all matrices

$$\begin{pmatrix} x^k & P(x) \\ 0 & 1 \end{pmatrix} \tag{2.5}$$

where $k \in \mathbb{Z}$ and $P(x) \in \mathbb{Z}[x, x^{-1}]$ (see e.g. [38, Section 2.2]). In contrast to $\mathrm{BS}(1,q)$ the group $\mathbb{Z} \wr \mathbb{Z}$ is not finitely presented [5]. A well-known infinite presentation of $\mathbb{Z} \wr \mathbb{Z}$ is $\langle a, t \mid [a^{t^i}, a^{t^j}] = 1 \ (i,j \in \mathbb{Z})\rangle$.

### 2.3.3. *Knapsack, exponent equations and the power word problem*

Let $G = \langle \Sigma \rangle$ be a f.g. group. Moreover, let $x_1, x_2, \ldots, x_l$ be pairwise distinct variables. A *knapsack expression* over $G$ is an expression of the form

$$E = v_0 u_1^{x_1} v_1 u_2^{x_2} v_2 \cdots u_l^{x_l} v_l \tag{2.6}$$

with $l \geq 1$, words $v_0, \ldots, v_l \in \Sigma^*$ and non-empty words $u_1, \ldots, u_l \in \Sigma^*$. A tuple $(n_1, \ldots, n_l) \in \mathbb{N}^l$ is a *$G$-solution* of $E$ if $v_0 u_1^{n_1} v_1 u_2^{n_2} v_2 \cdots u_l^{n_l} v_l = 1$ in $G$. With $\mathrm{sol}(G, E)$ we denote the set of all $G$-solutions of $E$. The *size* of $E$ is defined as $|E| = |v_0| + \sum_{i=1}^{l} |u_i| + |v_i|$. The *knapsack problem for $G$*, $\mathrm{Knapsack}(G)$ for short, is the following decision problem:

**Input** A knapsack expression $E$ over $G$.
**Question** Is $\mathrm{sol}(G, E)$ non-empty?

It is easy to observe that the concrete choice of the generating set $\Sigma$ has no influence on the decidability/complexity status of $\mathrm{Knapsack}(G)$. W.l.o.g. we can restrict to knapsack expressions of the form $u_1^{x_1} u_2^{x_2} \cdots u_l^{x_l} v$: for $E = v_0 u_1^{x_1} v_1 u_2^{x_2} v_2 \cdots u_l^{x_l} v_l$ and

$$E' = (v_0 u_1 v_0^{-1})^{x_1} (v_0 v_1 u_2 v_1^{-1} v_0^{-1})^{x_2} \cdots (v_0 \cdots v_{l-1} u_l v_{l-1}^{-1} \cdots v_0^{-1})^{x_l} v_0 \cdots v_{l-1} v_l$$

we have $\mathrm{sol}(G, E) = \mathrm{sol}(G, E')$.

An *exponent expression* over $G = \langle \Sigma \rangle$ is a formal expression $E$ as in (2.6), but in contrast to knapsack expressions, we allow $x_i = x_j$ for $i \neq j$. The set of solutions $\mathrm{sol}(G, E)$ for the exponent expression $E$ can be defined analogously to knapsack expressions. We define *solvability of systems of exponent equations over $G$*, $\mathrm{ExpEq}(G)$ for short, as the following decision problem:

**Input** A finite list of exponent expressions $E_1, \ldots, E_n$ over $G$.
**Question** Is $\bigcap_{i=1}^{n} \mathrm{sol}(G, E_i)$ non-empty?

This problem has been studied for various groups in [16,20,34,36].

A *power word* (over $\Sigma$) is a tuple $(u_1, k_1, u_2, k_2, \ldots, u_l, k_l)$ where $u_1, \ldots, u_l \in \Sigma^*$ are words over the group generators and $k_1, \ldots, k_l \in \mathbb{Z}$ are integers that are given in binary notation. Such a power word represents the word $u_1^{k_1} u_2^{k_2} \cdots u_l^{k_l}$. Quite often, we will identify the power word $(u_1, k_1, u_2, k_2, \ldots, u_l, k_l)$ with the word $u_1^{k_1} u_2^{k_2} \cdots u_l^{k_l}$. The *power word problem* for the f.g. group $G$, PowerWP($G$) for short, is defined as follows:

**Input** A power word $(u_1, k_1, u_2, k_2, \ldots, u_l, k_l)$.
**Question** Does $u_1^{k_1} u_2^{k_2} \cdots u_l^{k_l} = 1$ hold in $G$?

Due to the binary encoded exponents, a power word can be seen as a succinct description of an ordinary word. The size of the above power word $w$ is $\sum_{i=1}^{l} |u_i| + \lceil \log_2 k_i \rceil$ which is the length of the binary encoding of $w$.

## 3. Power word problem for matrix groups $T(\alpha)$

In this section we prove our first main result:

**Theorem 3.1.** *The following problem belongs to* $\mathsf{TC}^0$*:*

**Input** *A power word* $(u_1, k_1, u_2, k_2, \ldots, u_l, k_l)$ *over the alphabet* $\{a, a^{-1}, t, t^{-1}\}$ *and an algebraic number* $\alpha \in \mathbb{C} \setminus \{0\}$ *given by its minimal polynomial in dense representation.*
**Question** *Does* $u_1^{k_1} u_2^{k_2} \cdots u_l^{k_l} = 1$ *hold in* $T(\alpha)$*?*

*In particular, for every fixed* $\alpha \in \mathbb{C} \setminus \{0\}$, PowerWP($T(\alpha)$) *belongs to* $\mathsf{TC}^0$.

**Proof.** Note that the second statement of the theorem is clear for the case that $\alpha$ is transcendental: in this case, $T(\alpha)$ is isomorphic to $\mathbb{Z} \wr \mathbb{Z}$ for which the power word problem belongs to $\mathsf{TC}^0$ [35]. Hence, it suffices to prove the first statement of theorem. By Theorem 2.3, it suffices to reduce (in $\mathsf{TC}^0$) the problem from the first statement to sparse polynomial root testing.

Let us fix an algebraic number $\alpha \in \mathbb{C}\setminus\{0\}$ that is given by its minimal polynomial $p_\alpha(x)$. By Lemma 2.1 we can check in $\mathsf{TC}^0$ whether $\alpha$ is a root of unity and in the positive case compute in $\mathsf{TC}^0$ an $m$ in unary encoding such that $\alpha$ is an $m^{th}$ primitive root of unity. In particular, for a given (binary encoded) integer $k$ we can check in $\mathsf{TC}^0$ whether $\alpha^k = 1$: if $\alpha$ is not a root of unity we only have to check whether $k = 0$, otherwise we check for the above $m$ whether $m$ divides $k$.

Consider a power word of the form

$$P := \begin{pmatrix} \alpha^{k_1} & p_1(\alpha) \\ 0 & 1 \end{pmatrix}^{n_1} \cdot \begin{pmatrix} \alpha^{k_2} & p_2(\alpha) \\ 0 & 1 \end{pmatrix}^{n_2} \cdots \begin{pmatrix} \alpha^{k_l} & p_l(\alpha) \\ 0 & 1 \end{pmatrix}^{n_l}.$$

Here, the $n_i$ are binary encoded integers, all $p_i(x)$ are Laurent polynomials over $\mathbb{Z}$ that are given in dense unary representation, and the $k_i$ are given in unary

representation. We want to check whether $P$ evaluates to the identity matrix. We have

$$\begin{pmatrix} \alpha^{k_i} & p_i(\alpha) \\ 0 & 1 \end{pmatrix}^{n_i} = \begin{pmatrix} \alpha^{k_i n_i} & (1+\alpha^{k_i}+\alpha^{2k_i}+\cdots+\alpha^{(n_i-1)k_i})\cdot p_i(\alpha) \\ 0 & 1 \end{pmatrix}$$
$$= \begin{cases} \begin{pmatrix} \alpha^{k_i n_i} & \frac{\alpha^{k_i n_i}-1}{\alpha^{k_i}-1}\cdot p_i(\alpha) \\ 0 & 1 \end{pmatrix} & \text{if } \alpha^{k_i} \neq 1 \\ \begin{pmatrix} 1 & n_i \cdot p_i(\alpha) \\ 0 & 1 \end{pmatrix} & \text{if } \alpha^{k_i} = 1. \end{cases}$$

For $0 \leq j \leq l$ and $1 \leq i \leq l$ let us define

$$s_j = k_1 n_1 + \cdots + k_j n_j, \tag{3.1}$$

$$q_i(x) = \begin{cases} n_i \cdot p_i(x) & \text{if } \alpha^{k_i} = 1 \\ \dfrac{x^{k_i n_i}-1}{x^{k_i}-1}\cdot p_i(x) & \text{if } \alpha^{k_i} \neq 1 \end{cases} \tag{3.2}$$

Then, the power word $P$ can be written as

$$\begin{pmatrix} \alpha^{k_1\cdot n_1} & q_1(\alpha) \\ 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \alpha^{k_2\cdot n_2} & q_2(\alpha) \\ 0 & 1 \end{pmatrix} \cdots \begin{pmatrix} \alpha^{k_l\cdot n_l} & q_l(\alpha) \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \alpha^{s_l} & \sum\limits_{1\leq i\leq l} \alpha^{s_{i-1}} q_i(\alpha) \\ 0 & 1 \end{pmatrix}.$$

The binary encodings of the integers $s_j$ can be computed in $\mathsf{TC}^0$. Note that in case $\alpha^{k_i} \neq 1$, $q_i(x)$ is a rational algebraic expression.

We have to check whether

$$\alpha^{s_l} = 1, \tag{3.3}$$

$$\sum_{1\leq i\leq l} \alpha^{s_{i-1}} q_i(\alpha) = 0. \tag{3.4}$$

Equality (3.3) can be checked in $\mathsf{TC}^0$ by the above remark. The verification of (3.4) can be reduced to sparse polynomial root testing as follows. Let us replace the algebraic number $\alpha$ by a variable $x$ in the left-hand side of (3.4). We have to check whether $\alpha$ is a solution of the equation

$$\sum_{1\leq i\leq l} x^{s_{i-1}} q_i(x) = 0. \tag{3.5}$$

Let $J = \{i \in [1,l] \mid \alpha^{k_i} \neq 1\}$ and define the Laurent polynomial

$$r(x) = \prod_{i\in J} (x^{k_i} - 1).$$

We can compute in $\mathsf{TC}^0$ the set $J$ and the dense representation of $r(x)$ (iterated multiplication of densely represented polynomials is in $\mathsf{TC}^0$). We now multiply (3.5) with $r(x)$ and obtain the equation

$$\sum_{i=1}^{l} r_i(x) = 0, \tag{3.6}$$

where

$$r_i(x) := \begin{cases} n_i \cdot x^{s_{i-1}} \cdot r(x) \cdot p_i(x) & \text{if } \alpha^{k_i} = 1 \\ x^{s_{i-1}} \cdot (x^{k_i n_i} - 1) \cdot \prod_{j \in J \setminus \{i\}} (x^{k_j} - 1) \cdot p_i(x) & \text{if } \alpha^{k_i} \neq 1 \end{cases}$$

is a Laurent polynomial. Note that since $r(\alpha) \neq 0$, $\alpha$ is a solution of (3.5) if and only if $\alpha$ is a solution of (3.6).

We can compute for all $i \in [1, l]$ the sparse representation of $r_i(x)$ in $\mathsf{TC}^0$. For instance, in the second case ($\alpha^{k_i} \neq 1$), we first compute in $\mathsf{TC}^0$ the dense representation of $\prod_{j \in J \setminus \{i\}} (x^{k_j} - 1) \cdot p_i(x)$ (this is iterated multiplication of densely represented polynomials). This dense representation can be easily multiplied in $\mathsf{TC}^0$ with the sparse representation of $x^{s_{i-1}} \cdot (x^{k_i n_i} - 1) = x^{s_i} - x^{s_{i-1}}$, which yields the sparse representation of $r_i(x)$.

We can easily compute in $\mathsf{TC}^0$ the binary representation of a number $m \in \mathbb{N}$ such that all $x^m \cdot r_i(x)$ are ordinary polynomials from $\mathbb{Z}[x]$. Moreover, we can compute in $\mathsf{TC}^0$ the sparse representations of these polynomials. By multiplying (3.6) with $x^m$, we obtain the equation

$$S(x) = 0. \tag{3.7}$$

where $S(x) = \sum_{i=1}^{l} x^m \cdot r_i(x)$. The sparse representations of $S(x)$ can be computed in $\mathsf{TC}^0$. Since $\alpha \neq 0$, $\alpha$ is a solution of (3.6) if and only if $\alpha$ is a solution of (3.7). Finally, by Theorem 2.3 we can check in $\mathsf{TC}^0$ whether $S(\alpha) = 0$. This concludes the proof of the theorem. □

## 4. Knapsack for BS(1, q)

Whether the knapsack problem is decidable for $\mathrm{BS}(1, q)$ was left open in [14]. Our second main result gives a positive answer and also settles the computational complexity:

**Theorem 4.1.** *For every $q \geq 2$,* $\mathrm{Knapsack}(\mathrm{BS}(1, q))$ *is* $\mathsf{NP}$*-complete.*

Let us first remark that $\mathrm{BS}(1, q)$ is unusual in terms of its knapsack solution sets. In almost all groups where knapsack is known to be decidable, knapsack equations have semilinear solution sets [16,17,20,29,34,36]. After the discrete Heisenberg group [29], the groups $\mathrm{BS}(1, q)$ are only the second known example where this is not the case: the knapsack equation $t^{-x_1} a^{x_2} t^{x_3} = a$ has the non-semilinear solution set $\{(k, q^k, k) \mid k \in \mathbb{N}\}$.

Another unusual aspect is that knapsack is in $\mathsf{NP}$ although there are knapsack equations over $\mathrm{BS}(1, 2)$ whose solutions are all at least doubly exponential in the size of the equation:

**Theorem 4.2.** *There is a family $E_k = E_k(x, y, z)$, $k \geq 1$, of solvable knapsack expressions over* $\mathrm{BS}(1, 2)$ *such that $|E_k| = \Theta(k)$ and $z \geq (2^{2 \cdot 3^{k-1}} - 1)/3^k - 1$ for every solution of $E_k = 1$.*

**Proof.** It is a well-known fact in elementary number theory that for every $k \geq 1$, 2 is a primitive root modulo $3^k$, i.e., 2 generates the group $(\mathbb{Z}/3^k\mathbb{Z})^*$ (the group of units of the ring $\mathbb{Z}/3^k\mathbb{Z}$). See, for example, Theorem 3.6 and the remarks before Theorem 3.8 in [41]. Consider the knapsack equation

$$\begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}^x \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2^{-1} & 0 \\ 0 & 1 \end{pmatrix}^y \begin{pmatrix} 1 & -3^k \\ 0 & 1 \end{pmatrix}^z = \begin{pmatrix} 1 & 3^k+1 \\ 0 & 1 \end{pmatrix} \tag{4.1}$$

in $\mathrm{BS}(1,2)$. In the top-left entry, it implies $2^x 2^{-y} = 1$. Therefore, we must have $x = y$ in every solution. In this case, the left-hand side of eq. (4.1) is

$$\begin{pmatrix} 2^x & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 2^{-x} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & -z3^k \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 2^x - z \cdot 3^k \\ 0 & 1 \end{pmatrix}.$$

Therefore, eq. (4.1) is equivalent to $x = y$ and $2^x - z \cdot 3^k = 3^k + 1$. Since some non-zero power of 2 is congruent to 1 modulo $3^k$, eq. (4.1) has a solution. Moreover, any solution must satisfy $2^x \equiv 1 \pmod{3^k}$. Since 2 is a primitive root modulo $3^k$, $x$ must be a multiple of $|(\mathbb{Z}/3^k\mathbb{Z})^*| = \varphi(3^k) = 2 \cdot 3^{k-1}$ (here, $\varphi$ is Euler's phi-function). Moreover, $x$ must be non-zero, because $1 - z \cdot 3^k = 3^k + 1$ is not possible for $z \in \mathbb{N}$. We obtain $x \geq 2 \cdot 3^{k-1}$. Since $2^x - z \cdot 3^k = 3^k + 1$, this yields $z = (2^x - 3^k - 1)/3^k \geq (2^{2 \cdot 3^{k-1}} - 1)/3^k - 1$.

The knapsack expression $E_k$ from the lemma is of course obtained by bringing the right-hand side matrix in (4.1) to the left-hand side. To see that $|E_k| = \Theta(k)$, it suffices to show that the matrices

$$\begin{pmatrix} 1 & -3^k \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 3^k+1 \\ 0 & 1 \end{pmatrix}$$

can be produced by words of length $\Theta(k)$ over the generators of $\mathrm{BS}(1,2)$. This follows from the fact that $\log_2 3^k = \log_2(3) \cdot k = \Theta(k)$ (see also the proof of Theorem 2.4). □

**Remark 4.3.** Subject to Artin's conjecture on primitive roots [27], a similar doubly-exponential lower bound results for every $\mathrm{BS}(1,q)$ where $q \geq 2$ is not a perfect square. Moreover, Theorem 4.2 holds even if the variables $x, y, z$ range over $\mathbb{Z}$. For this, one replaces $3^k + 1$ with the inverse of 2 in $(\mathbb{Z}/3^k\mathbb{Z})^*$ in eq. (4.1).

Theorem 4.2 rules out a simple guess-and-verify strategy to show Theorem 4.1. If one has an exponential upper bound (in terms of input length) on the size of a smallest solution of a knapsack equation, then one can guess the binary representation of a solution and verify, using the power word problem, whether the guess is indeed a solution. The second step (verification of a solution using the power word problem) would work for $\mathrm{BS}(1,q)$ in polynomial time due to Theorem 3.1, but the first step (guessing a binary encoded candidate for a solution) does not work for $\mathrm{BS}(1,2)$ due to Theorem 4.2.

Our main tool for the proof of Theorem 4.1 is a recent result from [22] concerning the existential fragment of Büchi arithmetic.

### 4.1. *Büchi arithmetic*

*Büchi arithmetic* [10] is the first-order theory of the structure $(\mathbb{Z}, +, \geq, 0, 1, V_q)$. Here, $V_q$ is the function that maps $n \in \mathbb{Z}$ to the largest power of $q$ that divides $n$. It is well-known that Büchi arithmetic is decidable (this was first claimed in [10]; a correct proof was given in [8]). Here, we will only deal with the *existential fragment* of Büchi arithmetic, i.e., the set of all formulas in Büchi arithmetic of the form $\exists x_1 \exists x_2 \cdots \exists x_n : \phi(x_1, x_2, \ldots, x_n)$, where $\phi$ is quantifier-free. We will rely on the following recent result of Guépin, Haase, and Worrell [22]:

**Theorem 4.4 (c.f. [22]).** *The existential fragment of Büchi arithmetic belongs to* NP. [g]

We will also make use of the following simple lemma:

**Lemma 4.5.** *Given the $q$-ary representation of a number $r \in \mathbb{Z}[1/q]$ we can construct in polynomial time an existential Presburger formula over $(\mathbb{Z}, +, 0, 1)$ of size $\mathcal{O}(\|r\|_q)$ which expresses $y = r \cdot x$ for $x, y \in \mathbb{Z}$.*

**Proof.** Let $r = \sum_{-k \leq i \leq \ell} a_i q^i$ with $k, \ell \geq 0$ and $0 \leq a_i < q$ for $-k \leq i \leq \ell$. We have $y = rx$ if and only if $q^k y = r'x$ for $r' = \sum_{i=0}^{k+\ell} a_{i-k} q^i \in \mathbb{Z}$. Using iterated multiplication with the constant $q$ (which can be replaced by addition) we can easily define from $x$ and $y$ the integers $q^k y$ and $r'x$ by Presburger formulas of size $\mathcal{O}(k + \ell) = \mathcal{O}(\|r\|_q)$. □

### 4.2. *Proof of Theorem 4.1*

We start with the lower bound. The *multisubset sum problem* asks for integers $a_1, \ldots, a_d, b \in \mathbb{Z}$ given in binary, whether there exist natural numbers $x_1, \ldots, x_d \geq 0$ with $x_1 a_1 + \cdots + x_d a_d = b$. It is known to be NP-complete [25]. Since the knapsack equation

$$\begin{pmatrix} 1 & a_1 \\ 0 & 1 \end{pmatrix}^{x_1} \cdots \begin{pmatrix} 1 & a_d \\ 0 & 1 \end{pmatrix}^{x_d} = \begin{pmatrix} 1 & b \\ 0 & 1 \end{pmatrix}$$

is equivalent to $x_1 a_1 + \cdots + x_d a_d = b$, we obtain NP-hardness of knapsack over $\mathrm{BS}(1, q)$. Note that computing the $q$-ary representation of $a_i$ from the binary representation is possible in logspace (even in $\mathsf{TC}^0$).

For the upper bound we reduce Knapsack($\mathrm{BS}(1, q)$) to the existential fragment of Büchi arithmetic, which belongs to NP by Theorem 4.4. We proceed in three steps.

[g]The paper [22] shows an NP upper bound for the structure $(\mathbb{N}, +, 0, 1, V_q)$, but an existential sentence over the structure $(\mathbb{Z}, +, \geq, 0, 1, V_q)$ easily translates into one over $(\mathbb{N}, +, 0, 1, V_q)$.

**Step 1:** Expressing $M_g$ and $M_g^*$ using $S_\ell$. We first express a particular set of binary relations using existential first-order formulas over $(\mathbb{Z}, +, \geq, 0, 1, V_q, (S_\ell)_{\ell\in\mathbb{Z}})$. Here, for $\ell \in \mathbb{Z}$, $S_\ell$ is the binary predicate with

$$x\, S_\ell\, y \iff \exists r \in \mathbb{N}\ \exists s \in \mathbb{N}\colon x = q^r \wedge y = q^{r+\ell\cdot s}.$$

Let $T_{\mathbb{Z}}(q)$ denote the subset of matrices in $T(q)$ that have entries in $\mathbb{Z}$. We represent the matrix $\left(\begin{smallmatrix} m & n \\ 0 & 1 \end{smallmatrix}\right) \in T_{\mathbb{Z}}(q)$ by the pair $(m, n) \in \mathbb{Z} \times \mathbb{Z}$ (note that we must have $m \in \mathbb{N}$). Observe that we can define in the structure $(\mathbb{Z}, +, \geq, 0, 1, V_q, (S_\ell)_{\ell\in\mathbb{Z}})$ the set of pairs $(m, n) \in \mathbb{Z}$ such that $\left(\begin{smallmatrix} m & n \\ 0 & 1 \end{smallmatrix}\right) \in T_{\mathbb{Z}}(q)$, because this is equivalent to $m$ being a power of $q$, which is expressed by $1\, S_1\, m$.

A key trick is to express solvability of a knapsack equation $g_1^{x_1} \cdots g_d^{x_d} = g$ without introducing variables in the logic for $x_1, \ldots, x_d$. Instead, we employ the following binary relations $M_g$ and $M_g^*$ on $T_{\mathbb{Z}}(q)$, which allow us to express existence of powers implicitly. For $g \in T(q)$ and $x, y \in T_{\mathbb{Z}}(q)$, we have:

- $x\, M_g\, y \iff y = xg$,
- $x\, M_g^*\, y \iff \exists s \in \mathbb{N}\colon y = xg^s$.

We construct existential formulas of size polynomial in $\|g\|$ over the structure $(\mathbb{Z}, +, \geq, 0, 1, V_q, (S_\ell)_{\ell\in\mathbb{Z}})$, which define the relations $M_g$ and $M_g^*$. For the further consideration let

$$g = \begin{pmatrix} q^\ell & v \\ 0 & 1 \end{pmatrix}.$$

Note that the relation $M_g$ is easily expressible because we can express multiplication with $q^\ell$ and $v$ by existential Presburger formulas of length $\|g\|$, see Lemma 4.5.

We now focus on the relations $M_g^*$ and express

$$\begin{pmatrix} q^k & u \\ 0 & 1 \end{pmatrix} M_g^* \begin{pmatrix} q^m & w \\ 0 & 1 \end{pmatrix} \tag{4.2}$$

Observe that for $\ell \neq 0$, we have

$$\begin{aligned}\begin{pmatrix} q^k & u \\ 0 & 1 \end{pmatrix} \begin{pmatrix} q^\ell & v \\ 0 & 1 \end{pmatrix}^s &= \begin{pmatrix} q^k & u \\ 0 & 1 \end{pmatrix} \begin{pmatrix} q^{\ell s} & v + q^\ell v + \cdots + q^{(s-1)\ell} v \\ 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} q^k & u \\ 0 & 1 \end{pmatrix} \begin{pmatrix} q^{\ell s} & v\frac{q^{\ell s}-1}{q^\ell - 1} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} q^{k+\ell s} & u + v\frac{q^{k+\ell s}-q^k}{q^\ell-1} \\ 0 & 1 \end{pmatrix}.\end{aligned}$$

Therefore, eq. (4.2) is equivalent to

$$\exists x \in \mathbb{Z}\ \exists s \in \mathbb{N}\colon q^m = q^{k+\ell s} \ \wedge\ w = u + vx \ \wedge\ (q^\ell - 1)x = q^m - q^k.$$

Here, we can quantify $x$ over $\mathbb{Z}$, because

$$\frac{q^{k+\ell s} - q^k}{q^\ell - 1} = q^k + q^{k+\ell} + \cdots + q^{k+(s-1)\ell}$$

must be an integer ($k$ and $k + \ell s = m$ are non-negative). Note that since we can express multiplication with $v$ and $q^\ell$ by existential Presburger formulas of size

$\mathcal{O}(\|g\|)$ (Lemma 4.5), we can also express $w = u + vx$ and $(q^\ell - 1)x = q^m - q^k$ by existential Presburger formulas of size $\mathcal{O}(\|g\|)$. Finally, we can express $\exists s \in \mathbb{N}\colon q^m = q^{k+\ell s}$ using $q^k\, S_\ell\, q^m$.

It remains to express eq. (4.2) in the case $\ell = 0$. Note that

$$g^s = \begin{pmatrix} 1 & sv \\ 0 & 1 \end{pmatrix}$$

in this case. Therefore, eq. (4.2) is equivalent to

(i) there exists $s \in \mathbb{N}$ with $w = u + q^k \cdot s \cdot v$ and
(ii) $q^m = q^k$.

Note that condition (i) is equivalent to

$$w = u \ \vee \ \exists t \in \mathbb{N}\colon V_q(t) \geq q^k \wedge w = u + v \cdot t.$$

This is because choosing $t = q^k \cdot s$ yields (i). By Lemma 4.5, $w = u + v \cdot t$ can be expressed by an existential Presburger formula of size $\mathcal{O}(\|g\|)$.

**Step 2:** Expressing $S_\ell$ using $V_q$. In our second step, we show that the binary relations $M_g$ and $M_g^*$ can be expressed using existential formulas over $(\mathbb{Z}, +, \geq, 0, 1, V_q)$ of size $\mathrm{poly}(\|g\|)$. As shown above, for this it suffices to define $S_\ell$ by an existential formula over $(\mathbb{Z}, +, \geq, 0, 1, V_q)$ of size $\mathrm{poly}(\ell)$ (note that the relations $S_\ell$ occur only positively in the formulas from Step 1). For $m \in \mathbb{N}$, let $P_m$ be the predicate where $P_m(x)$ states that $x$ is a power of $m$. We first claim that for each $\ell \geq 0$, we can express $P_{q^\ell}$ using an existential formula of size polynomial in $\ell$ over $(\mathbb{Z}, +, \geq, 0, 1, V_q)$. The case $\ell = 0$ is clear. For the case $\ell \geq 1$ we use the following observation from the proof of Proposition 7.1 in [9]. Note that $P_q(x)$ is just $V_q(x) = x$.

Fact 4.6. For all $\ell \geq 1$, $P_{q^\ell}(x)$ if and only if $P_q(x)$ and $q^\ell - 1$ divides $x - 1$.

**Proof.** If $x$ is a power of $q^\ell$, then $x = q^{\ell \cdot s}$ for some $s \geq 0$. So, $x$ is a power of $q$. Moreover,

$$\frac{x-1}{q^\ell - 1} = \frac{q^{\ell \cdot s} - 1}{q^\ell - 1} = \sum_{i=0}^{s-1} q^{i\ell}$$

is an integer.

Conversely, suppose $x$ is a power of $q$ and $q^\ell - 1$ divides $x - 1$. Write $x = q^{\ell \cdot s + r}$ with $0 \leq r < \ell$. Observe that

$$x - 1 = q^{s\ell + r} - 1 = q^r(q^{s\ell} - 1) + (q^r - 1).$$

Since $q^\ell - 1$ divides $x - 1$ as well as $q^{s\ell} - 1$, we conclude that $q^\ell - 1$ divides $q^r - 1$. As $0 \leq r < \ell$, this is only possible with $r = 0$. This shows the above fact. □

Using the predicates $P_{q^\ell}$, we can now express $S_\ell$. Note that for $\ell \geq 0$, we have $x\, S_\ell\, y$ if and only if

$$y \geq x \wedge \bigvee_{i=0}^{\ell-1} P_{q^\ell}(q^i x) \wedge P_{q^\ell}(q^i y).$$

Furthermore, for $\ell < 0$, we have $x\, S_\ell\, y$ if and only if $y\, S_{|\ell|}\, x$. Therefore, we can express each $S_\ell$ using an existential formula of size polynomial in $\ell$ over $(\mathbb{Z}, +, \geq, 0, 1, V_q)$. Hence, we can express $M_g$ and $M_g^*$ using existential formulas of size $\mathrm{poly}(\|g\|)$ over $(\mathbb{Z}, +, \geq, 0, 1, V_q)$.

**Step 3:** Expressing solvability of knapsack. In the last step, we express solvability of a knapsack equation by an existential first-order sentence over $(\mathbb{Z}, +, \geq, 0, 1, V_q)$, using the predicates $M_g$ and $M_g^*$. We claim that $g_1^{x_1} \cdots g_d^{x_d} = g$ has a solution $(x_1, \ldots, x_d) \in \mathbb{N}^d$ if and only if there exist $h_0, \ldots, h_d \in T_{\mathbb{Z}}(q)$ with

$$h_0 M_{g_1}^* h_1 \ \wedge\ h_1 M_{g_2}^* h_2 \ \wedge \cdots \wedge\ h_{d-1} M_{g_d}^* h_d \ \wedge\ h_0 M_g h_d. \tag{4.3}$$

This can be stated by an existential sentence over $(\mathbb{Z}, +, \geq, 0, 1, V_q)$ of size polynomial in $\|g\| + \sum_{i=1}^d \|g_i\|$.

If such $h_0, \ldots, h_d$ exist, then for some $x_1, \ldots, x_d \in \mathbb{N}$, we have $h_i = h_{i-1} g_i^{x_i}$ for all $i \in [1, d]$ and $h_d = h_0 g$, which implies $g_1^{x_1} \cdots g_d^{x_d} = g$. For the converse, we observe that for each matrix $A \in T(q)$, there is some large enough $k \in \mathbb{N}$ such that $\left(\begin{smallmatrix} q^k & 0 \\ 0 & 1 \end{smallmatrix}\right) A \in T_{\mathbb{Z}}(q)$. Therefore, if $g_1^{x_1} \cdots g_d^{x_d} = g$, then there is some large enough $k \in \mathbb{N}$ so that for every $i \in [1, d]$, the matrix $\left(\begin{smallmatrix} q^k & 0 \\ 0 & 1 \end{smallmatrix}\right) g_1^{x_1} \cdots g_i^{x_i}$ has integer entries. With this, we set $h_0 = \left(\begin{smallmatrix} q^k & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and $h_i = h_{i-1} g_i^{x_i}$ for $i \in [1, d]$. Then we have $h_0, \ldots, h_d \in T_{\mathbb{Z}}(q)$ and eq. (4.3) is satisfied. □

## 5. Systems of exponent equations over BS(1, q)

Our algorithm for the knapsack problem in $\mathrm{BS}(1, q)$ cannot be extended to solvability of systems of exponent equations (not even to solvability of a single exponent equation). If we allow systems of exponent equations, we can show undecidability:

**Theorem 5.1.** *For every $q \in \mathbb{N}$ with $q \geq 2$, $\mathrm{ExpEq}(\mathrm{BS}(1, q))$ is undecidable.*

**Proof.** Consider the function $(x, y) \mapsto x \cdot 2^y$ on the natural numbers. Büchi and Senger [11, Corollary 5] have shown that the existential fragment of the first-order theory of $(\mathbb{N}, +, x \cdot 2^y)$ is undecidable. The proof generalizes to every function $(x, y) \mapsto x \cdot q^y$ for $q \in \mathbb{N}$, $q \geq 2$. We reduce this fragment to $\mathrm{ExpEq}(\mathrm{BS}(1, q))$. For this it suffices to consider an existentially quantified conjunction of formulas of the following form: $x \cdot q^y = z$, $x + y = z$, and $x < y$ (the latter allow to express inequalities). We replace each of these formulas by an equivalent exponent equation over $\mathrm{BS}(1, q)$. For this we use the two generators $a$ and $t$ from (2.4) (for $\alpha = q$).

The formula $x+y=z$ is clearly equivalent to $a^x a^y = a^z$, i.e., $a^x a^y a^{-z} = 1$. The formula $x < y$ is equivalent $\exists z \in \mathbb{N}\colon a^x a^z a\, a^{-y} = 1$. Finally, $x \cdot q^y = z$ is equivalent to $t^y a^x t^{-y} a^{-z} = 1$. □

Theorem 5.1 implies that there is no algorithm that computes from a given knapsack expression $E$ an existential formula of Büchi arithmetic that defines the set $\mathrm{sol}(\mathrm{BS}(1,q), E)$. This is because with such an algorithm one could also construct an existential formula of Büchi arithmetic for the set of solutions of a given systems of exponent equations. This has an interesting consequence for the proof of Theorem 4.1. There, we remarked that in the constructed formula (of Büchi arithmetic) the exponent variables $x_1, \ldots, x_d$ do not appear. This is indeed unavoidable.

## 6. Open problems

Several open problems arise from our work:

- What is the complexity/decidability status of the power word/knapsack problem for Baumslag-Solitar groups $\mathrm{BS}(p,q) = \langle a, t \mid ta^p t^{-1} = a^q \rangle$ for $p, q \geq 2$? Decidability of knapsack in case $\gcd(p,q) = 1$ was shown in [14], but the complexity as well as the decidability in case $\gcd(p,q) > 1$ are open. Since the word problem for $\mathrm{BS}(p,q)$ can be solved in logspace [45], one can easily show that the power word problem for $\mathrm{BS}(p,q)$ belongs to $\mathsf{PSPACE}$. By using techniques from [35] one might try to find a logspace reduction from the power word problem for $\mathrm{BS}(p,q)$ to the word problem for $\mathrm{BS}(p,q)$ (the same was done for a free group in [35]); this would show that the power word problem for $\mathrm{BS}(p,q)$ can be solved in logspace.
- Baumslag-Solitar groups $\mathrm{BS}(1,q)$ are examples of f.g. solvable linear groups. In [28] it was shown that for every f.g. solvable linear group the word problem can be solved in $\mathsf{TC}^0$. This leads to the question whether for every f.g. solvable linear group the power word problem belongs to $\mathsf{TC}^0$.
- The power word problem is a restriction of the compressed word problem, where it is asked whether the word produced by a so-called straight-line program (a context-free grammar that produces a single word) represents the group identity; see [33]. The compressed word problem for $\mathrm{BS}(1,q)$ belongs to $\mathsf{coRP}$ (the complement of randomized polynomial time); this holds in fact for every f.g. linear group [33]. No better complexity bound is known for the compressed word problem for $\mathrm{BS}(1,q)$.
- Is the knapsack problem decidable for every matrix group $T(\alpha)$ with $\alpha \in \mathbb{C} \setminus \{0\}$?

## Acknowledgements

Markus Lohrey has been supported by the DFG research project Lo748/12-1.

## References

[1] Eric Allender, Nikhil Balaji, and Samir Datta. Low-depth uniform threshold circuits and the bit-complexity of straight line programs. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science 2014, MFCS 2014*, volume 8635 of *Lecture Notes in Computer Science*, pages 13–24. Springer, 2014. doi:10.1007/978-3-662-44465-8_2.

[2] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach*. Cambridge University Press, 2009. URL: http://www.cambridge.org/catalogue/catalogue.asp?isbn=9780521424264.

[3] Lázló Babai, Robert Beals, Jin yi Cai, Gábor Ivanyos, and Eugene M.Luks. Multiplicative equations over commuting matrices. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 1996*, pages 498–507. ACM/SIAM, 1996. URL: http://dl.acm.org/citation.cfm?id=313852.314109.

[4] Laurent Bartholdi, Michael Figelius, Markus Lohrey, and Armin Weiß. Groups with ALOGTIME-hard word problems and PSPACE-complete circuit value problems. In *Proceedings of the 35th Computational Complexity Conference, CCC 2020*, volume 169 of *LIPIcs*, pages 29:1–29:29. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.29.

[5] Gilbert Baumslag. Wreath products and finitely presented groups. *Mathematische Zeitschrift*, 75(1):22–28, 1961. doi:10.1007/BF01211007.

[6] Pascal Bergsträßer, Moses Ganardi, and Georg Zetzsche. A characterization of wreath products where knapsack is decidable. In *Proceedings of the 38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021*, volume 187 of *LIPIcs*, pages 11:1–11:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.STACS.2021.11.

[7] Russell J. Bradford and James H. Davenport. Effective tests for cyclotonic polynomials. In *Proceedings of the 47th International Symposium on Symbolic and Algebraic Computation, ISSAC 1988*, volume 358 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1989. doi:10.1007/3-540-51084-2_22.

[8] Véronique Bruyère. Entiers et automates finis. Mémoire de fin d'études, Université de Mons, 1985.

[9] Véronique Bruyère, Georges Hansel, Christian Michaux, and Roger Villemaire. Logic and $p$-recognizable sets of integers. *Bulletin of the Belgian Mathematical Society*, 1:191–238, 1994. doi:10.36045/bbms/1103408547.

[10] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960. doi:10.1002/malq.19600060105.

[11] J. Richard Büchi and Steven Senger. Definability in the existential theory of concatenation and undecidable extensions of this theory. *Mathematical Logic Quarterly*, 34(4):337–342, 1988. doi:10.1002/malq.19880340410.

[12] Jin-Yi Cai, Richard J. Lipton, and Yechezkel Zalcstein. The complexity of the A B C problem. *SIAM Journal on Computing*, 29(6):1878–1888, 2000. doi:10.1137/S0097539794276853.

[13] Ruiwen Dong. The identity problem in the special affine group of $\mathbb{Z}^2$, 2023. URL: https://arxiv.org/abs/2301.09502, doi:10.48550/ARXIV.2301.09502.

[14] Fedor Dudkin and Alexander Treyer. Knapsack problem for Baumslag–Solitar groups. *Siberian Journal of Pure and Applied Mathematics*, 18:43–55, 2018. doi:10.33048/pam.2018.18.404.

[15] Wayne Eberly. Very fast parallel polynomial arithmetic. *SIAM Journal on Computing*, 18(5):955–976, 1989. doi:10.1137/0218066.

[16] Michael Figelius, Moses Ganardi, Markus Lohrey, and Georg Zetzsche. The complex-

ity of knapsack problems in wreath products. In *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming, ICALP 2020*, volume 168 of *LIPIcs*, pages 126:1–126:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.126.

[17] Michael Figelius, Markus Lohrey, and Georg Zetzsche. Closure properties of knapsack semilinear groups. *Journal of Algebra*, 589(1):437–482, 2022. doi:10.1016/j.jalgebra.2021.08.016.

[18] Elizaveta Frenkel, Andrey Nikolaev, and Alexander Ushakov. Knapsack problems in products of groups. *Journal of Symbolic Computation*, 74:96–108, 2016. doi:10.1016/j.jsc.2015.05.006.

[19] Esther Galby, Joël Ouaknine, and James Worrell. On matrix powering in low dimensions. In *Proceedings of the 32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015*, volume 30 of *LIPIcs*, pages 329–340. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.STACS.2015.329.

[20] Moses Ganardi, Daniel König, Markus Lohrey, and Georg Zetzsche. Knapsack problems for wreath products. In *Proceedings of the 35th Symposium on Theoretical Aspects of Computer Science, STACS 2018*, volume 96 of *LIPIcs*, pages 32:1–32:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.STACS.2018.32.

[21] Guoqiang Ge. Testing equalities of multiplicative representations in polynomial time (extended abstract). In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science, FOCS 1993*, pages 422–426. IEEE Computer Society, 1993. doi:10.1109/SFCS.1993.366845.

[22] Florent Guépin, Christoph Haase, and James Worrell. On the existential theories of Büchi arithmetic and linear $p$-adic fields. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019*, pages 1–10. IEEE Computer Society, 2019. doi:10.1109/LICS.2019.8785681.

[23] Luc Guyot. Limits of metabelian groups. *International Journal of Algebra and Computation*, 22(4), 2012. doi:10.1142/S0218196712500312.

[24] Luc Guyot. Generators of split extensions of abelian groups by cyclic groups. *Groups, Geometry, and Dynamics*, 12(2):765–802, 2018. doi:10.4171/GGD/455.

[25] Christoph Haase. *On the complexity of model checking counter automata*. PhD thesis, University of Oxford, St Catherine's College, 2011.

[26] William Hesse, Eric Allender, and David A. Mix Barrington. Uniform constant-depth threshold circuits for division and iterated multiplication. *Journal of Computer and System Sciences*, 65(4):695–716, 2002. doi:10.1016/S0022-0000(02)00025-9.

[27] Christopher Hooley. On Artin's conjecture. *Journal für die reine und angewandte Mathematik*, 1967(225):209–220, 1967. doi:10.1515/crll.1967.225.209.

[28] Daniel König and Markus Lohrey. Evaluation of circuits over nilpotent and polycyclic groups. *Algorithmica*, 80(5):1459–1492, 2018. doi:10.1007/s00453-017-0343-z.

[29] Daniel König, Markus Lohrey, and Georg Zetzsche. Knapsack and subset sum problems in nilpotent, polycyclic, and co-context-free groups. In *Algebra and Computer Science*, volume 677 of *Contemporary Mathematics*, pages 138–153. American Mathematical Society, 2016. doi:10.1090/conm/677/13625.

[30] Serge Lang. *Algebra*. Graduate Texts in Mathematics. Springer New York, NY, 2002. Revised Third Edition. doi:10.1007/978-1-4613-0041-0.

[31] Klaus-Jörn Lange and Pierre McKenzie. On the complexity of free monoid morphisms. In *Proceedings of the 9th International Symposium on Algorithms and Computation, ISAAC 1998*, number 1533 in Lecture Notes in Computer Science, pages 247–256.

Springer, 1998. doi:10.1007/3-540-49381-6_27.

[32] H. W. Lenstra, Jr. Finding small degree factors of lacunary polynomials. In *Number Theory in Progress, vol. 1 Diophantine Problems and Polynomials*, pages 267–276. Walter de Gruyter, 1999.

[33] Markus Lohrey. *The Compressed Word Problem for Groups*. SpringerBriefs in Mathematics. Springer, 2014. doi:10.1007/978-1-4939-0748-9.

[34] Markus Lohrey. Knapsack in hyperbolic groups. *Journal of Algebra*, 545:390–415, 2020. doi:10.1016/j.jalgebra.2019.04.008.

[35] Markus Lohrey and Armin Weiß. The power word problem. In *Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019*, volume 138 of *LIPIcs*, pages 43:1–43:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.MFCS.2019.43.

[36] Markus Lohrey and Georg Zetzsche. Knapsack in graph groups. *Theory of Computing Systems*, 62(1):192–246, 2018. doi:10.1007/s00224-017-9808-3.

[37] Markus Lohrey and Georg Zetzsche. Knapsack and the power word problem in solvable Baumslag-Solitar groups. In *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020*, volume 170 of *LIPIcs*, pages 67:1–67:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.67.

[38] Alexei Miasnikov, Vitaly Roman'kov, Alexander Ushakov, and Anatoly Vershik. The word and geodesic problems in free solvable groups. *Transactions of the American Mathematical Society*, 362(9):4655–4682, 2010. doi:10.1090/S0002-9947-10-04959-7.

[39] Alexei Mishchenko and Alexander Treier. Knapsack problem for nilpotent groups. *Groups Complexity Cryptology*, 9(1):87, 2017. doi:10.1515/gcc-2017-0006.

[40] Alexei Myasnikov, Andrey Nikolaev, and Alexander Ushakov. Knapsack problems in groups. *Mathematics of Computation*, 84:987–1016, 2015. doi:10.1090/S0025-5718-2014-02880-9.

[41] Melvyn B. Nathanson. *Elementary Methods in Number Theory*. Springer, 2000. doi:10.1007/b98870.

[42] Florian Stober and Armin Weiß. The power word problem in graph products. In *Proceedings to the 26th International Conference on Developments in Language Theory, DLT 2022*, volume 13257 of *Lecture Notes in Computer Science*, pages 286–298. Springer, 2022. doi:10.1007/978-3-031-05578-2_23.

[43] Heribert Vollmer. *Introduction to Circuit Complexity*. Springer, 1999. doi:10.1007/978-3-662-03927-4.

[44] Armin Weiß. *On the complexity of conjugacy in amalgamated products and HNN extensions*. PhD thesis, University of Stuttgart, 2015. URL: http://elib.uni-stuttgart.de/opus/volltexte/2015/10018/.

[45] Armin Weiß. A logspace solution to the word and conjugacy problem of generalized Baumslag-Solitar groups. In *Algebra and Computer Science*, volume 677 of *Contemporary Mathematics*. American Mathematical Society, 2016. doi:https://doi.org/10.1090/conm/677/13628.

[46] Wolfgang Woess. *Random Walks on Infinite Graphs and Groups*. Cambridge University Press, 2000. doi:10.1017/CBO9780511470967.