The Complexity of Knapsack in Graph Groups

Markus Lohrey[1] **and Georg Zetzsche**[*2]

1 **Universität Siegen, Germany**
`lohrey@eti.uni-siegen.de`
2 **LSV, CNRS & ENS Cachan, Université Paris-Saclay, France**
`zetzsche@lsv.fr`

Abstract

Myasnikov et al. have introduced the knapsack problem for arbitrary finitely generated groups. In [19] the authors proved that for each graph group, the knapsack problem can be solved in NP. Here, we determine the exact complexity of the problem for every graph group. While the problem is TC^0-complete for complete graphs, it is LogCFL-complete for each (non-complete) transitive forest. For every remaining graph, the problem is NP-complete.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases knapsack, subset sum, graph groups, decision problems in group theory

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

1 Introduction

In their paper [21], Myasnikov, Nikolaev, and Ushakov started the investigation of discrete optimization problems, which are classically formulated over the integers, for arbitrary (possibly non-commutative) groups. The general goal of this line of research is to study to what extent results from the classical commutative setting can be transferred to the non-commutative setting. Among other problems, Myasnikov et al. introduced for a finitely generated group G the *knapsack problem* and the *subset sum problem*. The input for the knapsack problem is a sequence of group elements $g_1, \ldots, g_k, g \in G$ (specified by finite words over the generators of G) and it is asked whether there exists a solution $(x_1, \ldots, x_k) \in \mathbb{N}^k$ of the equation $g_1^{x_1} \cdots g_k^{x_k} = g$. For the subset sum problem one restricts the solution to $\{0,1\}^k$. For the particular case $G = \mathbb{Z}$ (where the additive notation $x_1 \cdot g_1 + \cdots + x_k \cdot g_k = g$ is usually preferred) these problems are NP-complete if the numbers $g_1, \ldots, g_k, g$ are encoded in binary representation. For subset sum, this is a classical result from Karp's seminal paper [15] on NP-completeness. Knapsack for integers is usually formulated in a more general form in the literature; NP-completeness of the above form (for binary encoded integers) was shown in [11], where the problem was called MULTISUBSET SUM.[1] Interestingly, if we consider subset sum for the group $G = \mathbb{Z}$, but encode the input numbers $g_1, \ldots, g_k, g$ in unary notation, then the problem is in DLOGTIME-uniform TC^0 (a small subclass of polynomial time and even of logarithmic space that captures the complexity of multiplication of binary encoded numbers; see e.g. the book [25] for more details) [7], and the same holds for knapsack (see Theorem 1). Related results can be found in [13]. Implicitly, the knapsack problem was also

* This author is supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

[1] Note that if we ask for a solution $(x_1, \ldots, x_k)$ in $\mathbb{Z}^k$, then knapsack can be solved in polynomial time (even for binary encoded integers) by checking whether $\gcd(g_1, \ldots, g_k)$ divides g.

studied by Babai et al. [3], where it is shown that knapsack for commutative matrix groups over algebraic number fields can be solved in polynomial time.

In [21] the authors encode elements of the finitely generated group G by words over the group generators and their inverses, which corresponds to the unary encoding of integers. Another, more succinct encoding of group elements uses *straight-line programs* (SLP). These are context-free grammars that produce a single word. Over a unary alphabet, one can achieve for every word exponential compression with SLPs: The word a^n can be produced by an SLP of size $O(\log n)$. This shows that knapsack and subset sum for the group $\mathbb{Z}$ with SLP-compressed group elements correspond to the classical knapsack and subset sum problem with binary encoded numbers. To distinguish between the two variants, we will speak in this introduction of uncompressed knapsack (resp., subset sum) if the input group elements are given explicitly by words over the generators. On the other hand, if these words are represented by SLPs, we will speak of SLP-compressed knapsack (resp., subset sum). Later in this paper, we will only use the uncompressed versions, and denote these simply with knapsack and subset sum, respectively.

In our recent paper [19], we started to investigate knapsack and subset sum for graph groups, which are also known as right-angled Artin groups in group theory. A graph group is specified by a finite simple graph Γ and denoted with $\mathbb{G}(\Gamma)$. The vertices are the generators of the group, and two generators a and b are allowed to commute if and only if a and b are adjacent in Γ. Graph groups interpolate between free groups and free abelian groups and can be seen as a group counterpart of trace monoids (free partially commutative monoids), which have been used for the specification of concurrent behavior. In combinatorial group theory, graph groups are currently an active area of research, mainly because of their rich subgroup structure, see e.g. [4, 5, 9].

Contribution. In [19] we proved that for every graph group, SLP-compressed knapsack (resp., subset sum) is NP-complete. This result generalizes the classical result for knapsack with binary encoded integers. Moreover, we proved that uncompressed knapsack and subset sum are NP-complete for the group $F_2 \times F_2$ (F_2 is the free group on two generators). The group $F_2 \times F_2$ is the graph group $\mathbb{G}(\Gamma)$, where the graph Γ is a cycle on four nodes. This result leaves open the complexity of uncompressed knapsack and subset sum for graph groups, whose underlying graph does not contain an induced cycle on four nodes. In this paper, we completely settle this open problem for knapsack by showing the following results:

(i) Uncompressed knapsack and subset sum for $\mathbb{G}(\Gamma)$ are complete for TC^0 if Γ is a complete graph (and thus $\mathbb{G}(\Gamma)$ is a free abelian group).[2]
(ii) Uncompressed knapsack and subset sum for $\mathbb{G}(\Gamma)$ are LogCFL-complete if Γ is not a complete graph and neither contains an induced cycle on four nodes (C4) nor an induced path on four nodes (P4).
(iii) Uncompressed knapsack for $\mathbb{G}(\Gamma)$ is NP-complete if Γ contains an induced C4 or an induced P4.

Overview of the proofs. The result (i) is a straightforward extension of the corresponding result for $\mathbb{Z}$ [7]. The statements in (ii) and (iii) are less obvious. Recall that LogCFL is the closure of the context-free languages under logspace reductions; it is contained in NC^2.

[2] In the following, TC^0 always refers to its DLOGTIME-uniform version.

To show the upper bound in (ii), we use the fact that the graph groups $\mathbb{G}(\Gamma)$, where Γ neither contains an induced $\mathsf{C4}$ nor an induced $\mathsf{P4}$ (these graphs are the so called transitive forests), are exactly those groups that can be built up from $\mathbb{Z}$ using the operations of free product and direct product with $\mathbb{Z}$. We then construct inductively over these operations a logspace-bounded auxiliary pushdown automaton working in polynomial time (these machines accept exactly the languages in LogCFL) that checks whether an acyclic finite automaton accepts a word that is trivial in the graph group. In order to apply this result to knapsack, we finally show that every solvable knapsack instance over a graph group $\mathbb{G}(\Gamma)$ with Γ a transitive forest has a solution with polynomially bounded exponents. This result might be of independent interest.

For the lower bound in (ii), it suffices to consider the group F_2 (the free group on two generators). Our proof is based on the fact that the context-free languages are exactly those languages that can be accepted by valence automata over F_2. This is a reinterpretation of the classical theorem of Chomsky and Schützenberger. To the authors' knowledge, the result (ii) is the first completeness result for LogCFL in the area of combinatorial group theory.

Finally, for the result (iii) it suffices to show NP-hardness of knapsack for the graph group $\mathbb{G}(\mathsf{P4})$ (the NP upper bound and the lower bound for $\mathsf{C4}$ is shown in [19]). We apply a technique that was first used in a paper by Aalbersberg and Hoogeboom [1] to show that the intersection non-emptiness problem for regular trace languages is undecidable for $\mathsf{P4}$.

Full proofs can be found in the long version [18].

2 Knapsack and Exponent Equations

We assume that the reader has some basic knowledge concerning (finitely generated) groups (see e.g. [20] for further details). Let G be a finitely generated group, and let A be a finite generating set for G. Then, elements of G can be represented by finite words over the alphabet $A^{\pm 1} = A \cup A^{-1}$. An *exponent equation* over G is an equation of the form

$$h_0 g_1^{\mathsf{x}_1} h_1 g_2^{\mathsf{x}_2} h_2 \cdots g_k^{\mathsf{x}_k} h_k = 1$$

where $g_1, g_2, \ldots, g_k, h_0, h_1, \ldots, h_k \in G$ are group elements that are given by finite words over the alphabet $A^{\pm 1}$ and $\mathsf{x}_1, \mathsf{x}_2, \ldots, \mathsf{x}_k$ are not necessarily distinct variables. Such an exponent equation is *solvable* if there exists a mapping $\sigma\colon \{\mathsf{x}_1, \ldots, \mathsf{x}_k\} \to \mathbb{N}$ such that $h_0 g_1^{\sigma(\mathsf{x}_1)} h_1 g_1^{\sigma(\mathsf{x}_2)} h_2 \cdots g_k^{\sigma(\mathsf{x}_k)} h_k = 1$ in the group G. The *size* of an equation is $\sum_{i=0}^{k} |h_i| + \sum_{i=1}^{k} |g_i|$, where $|g|$ denotes the length of the shortest word $w \in (A^{\pm 1})^*$ representing g. *Solvability of exponent equations over* G is the following computational problem:

Input: An exponent equation E over G (with elements of G specified by words over $A^{\pm 1}$).
Question: Is E solvable?

The *knapsack problem* for the group G is the restriction of solvability of exponent equations over G to exponent equations of the form $g_1^{\mathsf{x}_1} g_2^{\mathsf{x}_2} \cdots g_k^{\mathsf{x}_k} g^{-1} = 1$ or, equivalently, $g_1^{\mathsf{x}_1} g_2^{\mathsf{x}_2} \cdots g_k^{\mathsf{x}_k} = g$ where the exponent variables $\mathsf{x}_1, \ldots, \mathsf{x}_k$ have to be pairwise different. The *subset sum problem* for the group G is defined in the same way as the knapsack problem, but the exponent variables $\mathsf{x}_1, \ldots, \mathsf{x}_k$ have to take values in $\{0, 1\}$.

It is a simple observation that the decidability and complexity of solvability of exponent equations over G as well as the knapsack problem and subset sum problem for G does not depend on the chosen finite generating set for the group G. Therefore, we do not have to mention the generating set explicitly in these problems.

▶ Remark. Since we are dealing with a group, one might also allow solution mappings $\sigma\colon \{\mathsf{x}_1, \ldots, \mathsf{x}_k\} \to \mathbb{Z}$ to the integers. This variant of solvability of exponent equations

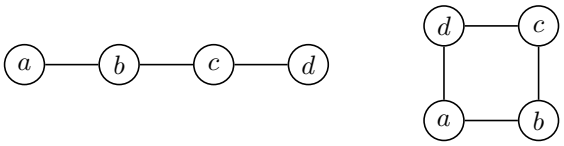

Figure 1 P4 and C4

(knapsack, respectively) can be reduced to the above version, where σ maps to $\mathbb{N}$, by simply replacing a power $g_i^{\mathsf{x}_i}$ by $g_i^{\mathsf{x}_i}(g_i^{-1})^{\mathsf{y}_i}$, where y_i is a fresh variable.

3 Traces and Graph Groups

Let (A, I) be a finite simple graph. In other words, the edge relation $I \subseteq A \times A$ is irreflexive and symmetric. It is also called the *independence relation*, and (A, I) is called an *independence alphabet*. We say that $a \in A$ *depends on* $b \in A$ if $(a, b) \notin I$. We consider the monoid $\mathbb{M}(A, I) = A^*/\equiv_I$, where $\equiv_I$ is the smallest congruence relation on the free monoid A^* that contains all pairs (ab, ba) with $a, b \in A$ and $(a, b) \in I$. This monoid is called a *trace monoid* or *partially commutative free monoid*. Elements of $\mathbb{M}(A, I)$ are called *Mazurkiewicz traces* or simply *traces*. The trace represented by the word u is denoted by $[u]_I$, or simply $[u]$ if no confusion can arise. The empty trace $[\varepsilon]_I$ is the identity element of the monoid $\mathbb{M}(A, I)$ and is denoted by 1. For a language $L \subseteq A^*$ we denote with $[L]_I = \{[u]_I \in \mathbb{M}(A, I) \mid u \in L\}$ the set of traces represented by L. Figure 1 shows two important indepedence alphabets that we denote with P4 (path on four nodes) and C4 (cycle on four nodes). Note that $\mathbb{M}(\mathsf{C4}) = \{a, c\}^* \times \{b, d\}^*$. For more details on traces see [6].

With an independence alphabet (A, I) we associate the finitely presented group $\mathbb{G}(A, I) = \langle A \mid ab = ba \ ((a, b) \in I)\rangle$. More explicitly, this group can be defined as follows: Let $A^{-1} = \{a^{-1} \mid a \in A\}$ be a disjoint copy of the alphabet A. We extend the independence relation I to $A^{\pm 1} = A \cup A^{-1}$ by $(a^x, b^y) \in I$ for all $(a, b) \in I$ and $x, y \in \{-1, 1\}$. Then $\mathbb{G}(A, I)$ is the quotient monoid $(A^{\pm 1})^*/\sim_I$, where $\sim_I$ is the smallest congruence relation that contains (i) all pairs (ab, ba) for $a, b \in A^{\pm 1}$ with $(a, b) \in I$ and (ii) all pairs (aa^{-1}, ε) and $(a^{-1}a, \varepsilon)$ for $a \in A$.

A group $\mathbb{G}(A, I)$ is called a *graph group*, or *right-angled Artin group*[3], or *free partially commutative group*. Here, we use the term graph group. Graph groups received a lot of attention in group theory during the last years, mainly due to their rich subgroup structure [4, 5, 9], and their relationship to low dimensional topology [2, 12, 26].

4 TC^0- and LogCFL-completeness

Let us first consider free abelian groups $\mathbb{Z}^m$. Note that $\mathbb{Z}^m$ is isomorphic to the graph group $\mathbb{G}(A, I)$ where (A, I) is the complete graph on m nodes. Our first result is a simple combination of known results [7, 22].

▶ **Theorem 1.** *For every fixed $m \geq 1$, knapsack and subset sum for the free abelian group $\mathbb{Z}^m$ are complete for TC^0. Hence, knapsack and subset sum for $\mathbb{G}(A, I)$ are complete for TC^0 if (A, I) is a non-empty complete graph.*

[3] This term comes from the fact that right-angled Artin groups are exactly the Artin groups corresponding to right-angled Coxeter groups.

We now characterize those graph groups where knapsack for $\mathbb{G}(A, I)$ is LogCFL-complete. The class LogCFL consists of all problems that are logspace reducible to a context-free language. It is included in the parallel complexity class NC^2 and has several alternative characterizations (see e.g. [24, 25]).

The *comparability graph* of a rooted tree t is the simple graph with the same vertices as t, but has an edge between two vertices whenever one is a descendent of the other in t. A graph (A, I) is a *transitive forest* if it is a disjoint union of comparability graphs of trees.

▶ **Theorem 2.** *If (A, I) is a transitive forest and not complete, then knapsack and subset sum for $\mathbb{G}(A, I)$ are* LogCFL*-complete.*

If the graph (A, I) is the disjoint union of graphs Γ_0 and Γ_1, then by definition, we have $\mathbb{G}(A, I) \cong \mathbb{G}(\Gamma_0) * \mathbb{G}(\Gamma_1)$. If one vertex v of (A, I) is adjacent to every other vertex and removing v from (A, I) results in the graph Γ_0, then $\mathbb{G}(A, I) \cong \mathbb{G}(\Gamma_0) \times \mathbb{Z}$. Therefore, we have the following *inductive characterization* of the graph groups $\mathbb{G}(A, I)$ for transitive forests (A, I): It is the smallest class of groups containing the trivial group that is closed under taking (i) free products and (ii) direct products with $\mathbb{Z}$.

Acyclic Automata In both the upper and the lower bound proof for Theorem 2, we employ the membership problem for acyclic automata, which has already been studied in connection with the knapsack and subset sum problem [8, 16].

We define a *finite automaton* as a tuple $\mathcal{A} = (Q, \Sigma, \Delta, q_0, q_f)$, where Q is a finite set of states, Σ is the *input alphabet*, $q_0 \in Q$ is the *initial state*, $q_f \in Q$ is the *final state*, and $\Delta \subseteq Q \times \Sigma^* \times Q$ is a finite set of *transitions*. The language accepted by $\mathcal{A}$ is denoted with $L(\mathcal{A})$. An *acyclic automaton* is a finite automaton $\mathcal{A} = (Q, \Sigma, \Delta, q_0, q_f)$ such that the relation $\{(p, q) \mid \exists w \in \Sigma^* : (p, w, q) \in \Delta\}$ is acyclic. For a graph group $\mathbb{G}(A, I)$ the *membership problem for acyclic automata* is the following computational problem:

Input: An acyclic automaton $\mathcal{A}$ over the input alphabet $A \cup A^{-1}$.
Question: Is there a word $w \in L(\mathcal{A})$ such that $w = 1$ in $\mathbb{G}(A, I)$?

In order to show the upper bound in Theorem 2, we reduce knapsack for $\mathbb{G}(A, I)$ with (A, I) a transitive forest to the membership problem for acyclic automata for $\mathbb{G}(A, I)$ (note that for subset sum this reduction is obvious). Then, we apply Proposition 3 below. From work of Frenkel, Nikolaev, and Ushakov [8], it follows that the membership problem for acyclic automata is in P. We strengthen this to LogCFL by constructing inductively over the operations of free product and direct product with $\mathbb{Z}$ a logspace-bounded auxiliary pushdown automaton working in polynomial time (these machines accept exactly the languages in LogCFL) that checks whether an acyclic automaton accepts a word that is trivial in $\mathbb{G}(A, I)$.

▶ **Proposition 3.** *If (A, I) is a transitive forest, then the membership problem for acyclic automata over $\mathbb{G}(A, I)$ is in* LogCFL.

4.1 Bounds on knapsack solutions

As mentioned above, we reduce for graph groups $\mathbb{G}(A, I)$ with (A, I) a transitive forest the knapsack problem to the membership problem for acyclic automata. To this end, we show that every solvable knapsack instance has a solution where all exponents are bounded polynomially in the size of the instance. The latter is the most involved proof in our paper.

Frenkel, Nikolaev, and Ushakov [8] call groups with this property *polynomially bounded knapsack groups* and show that this class is closed under taking free products. However, it is not clear if direct products with $\mathbb{Z}$ also inherit this property and we leave this question open.

Hence, we are looking for a property that yields polynomial size solutions and is passed on to free products and to direct products with $\mathbb{Z}$. It is known that the solution sets are always semilinear. If (A, I) is a transitive forest, this follows from a more general semilinearity property of rational sets [17] and for arbitrary graph groups, this was shown in [19]. Note that it is not true that the solution sets always have polynomial size semilinear representations. This already fails in the case of $\mathbb{Z}$: The equation $\mathsf{x}_1 + \cdots + \mathsf{x}_k = k$ has $\binom{2k-1}{k} \geq 2^k$ solutions. We will show here that the solution sets have semilinear representations where every occuring number is bounded by a polynomial.

For a vector $x = (x_1, \ldots, x_k) \in \mathbb{Z}^k$, we define the norm $\|x\| = \max\{|x_i| \mid i \in [1, k]\}$. For a subset $T \subseteq \mathbb{N}^k$, we write $T^{\oplus}$ for the smallest subset of $\mathbb{N}^k$ that contains $\{0\} \cup T$ and is closed under addition. A subset $S \subseteq \mathbb{N}^k$ is called *linear* if there is a vector $x \in \mathbb{N}^k$ and a finite set $F \subseteq \mathbb{N}^k$ such that $S = x + F^{\oplus}$. Note that a set is linear if and only if it can be written as $x + A\mathbb{N}^t$ for some $x \in \mathbb{N}^k$ and some matrix $A \in \mathbb{N}^{k \times t}$. Here, $A\mathbb{N}^t$ denotes the set of all vectors Ay for $y \in \mathbb{N}^t$. A *semilinear set* is a finite union of linear sets. If $S = \bigcup_{i=1}^{n} x_i + F_i^{\oplus}$ for $x_1, \ldots, x_n \in \mathbb{N}^k$ and finite sets $F_1, \ldots, F_n \subseteq \mathbb{N}^k$, then the tuple $(x_1, F_1, \ldots, x_n, F_n)$ is a *semilinear representation* of S and the *magnitude* of this representation is defined as the maximum of $\|y\|$, where y ranges over all elements of $\bigcup_{i=1}^{n} \{x_i\} \cup F_i$. The *magnitude* of a semilinear set S is the smallest magnitude of a semilinear representation for S.

▶ **Definition 4.** A group G is called *knapsack tame* if there is a polynomial p such that for every exponent equation $h_0 g_1^{\mathsf{x}_1} h_1 g_2^{\mathsf{x}_2} h_2 \cdots g_n^{\mathsf{x}_k} h_k = 1$ of size n with pairwise distinct variables $\mathsf{x}_1, \ldots, \mathsf{x}_k$, the set $S \subseteq \mathbb{N}^k$ of solutions is semilinear of magnitude at most $p(n)$.

Observe that although the size of an exponent equation may depend on the chosen generating set of G, changing the generating set increases the size only by a constant factor. Thus, whether or not a group is knapsack tame is independent of the chosen generating set.

▶ **Theorem 5.** *If (A, I) is a transitive forest, then $\mathbb{G}(A, I)$ is knapsack tame.*

Note that Theorem 5 implies in particular that every solvable exponent equation has a polynomially bounded solution. Theorem 5 and Proposition 3 easily yield the upper bound in Theorem 2.

We prove Theorem 5 by showing that knapsack tameness transfers from groups G to $G \times \mathbb{Z}$ (Proposition 6) and from G and H to $G * H$ (Proposition 10). Since the trivial group is obviously knapsack tame, the inductive characterization of groups $\mathbb{G}(A, I)$ for transitive forests (A, I) immediately yields Theorem 5.

4.2 Tameness of direct products with $\mathbb{Z}$

In this section, we sketch a proof of the following result.

▶ **Proposition 6.** *If G is knapsack tame, then so is $G \times \mathbb{Z}$.*

Linear Diophantine equations We employ a result of Pottier [23], which bounds the norm of minimal non-negative solutions to a linear Diophantine equation. Let $A \in \mathbb{Z}^{k \times m}$ be an integer matrix where a_{ij} is the entry of A at row i and column j. We will use the norms $\|A\|_{1,\infty} = \max_{i \in [1,k]}(\sum_{j \in [1,m]} |a_{ij}|)$, $\|A\|_{\infty,1} = \max_{j \in [1,m]}(\sum_{i \in [1,k]} |a_{ij}|)$ and $\|A\|_{\infty} = \max_{i \in [1,k], j \in [1,m]} |a_{ij}|$ for matrices and $\|x\|_1 = \sum_{i=1}^{m} |x_i|$ for vectors $x \in \mathbb{Z}^m$. Recall that $\|x\| = \max_{i \in [1,m]} |x_i|$. A solution $x \in \mathbb{N}^m \setminus \{0\}$ to the equation $Ax = 0$ is *minimal* if there is no $y \in \mathbb{N}^m \setminus \{0\}$ with $Ay = 0$ and $y \leq x$, $y \neq x$. The set of all solutions clearly forms a submonoid of $\mathbb{N}^m$. Let r be the rank of A.

▶ **Theorem 7** (Pottier [23]). *Each non-trival minimal solution $x \in \mathbb{N}^m$ to $Ax = 0$ satisfies $\|x\|_1 \leq (1 + \|A\|_{1,\infty})^r$.*

By applying Theorem 7 to the matrix $(A \mid -b)$, it is easy to deduce that for each $x \in \mathbb{N}^m$ with $Ax = b$, there is a $y \in \mathbb{N}^m$ with $Ay = b$, $y \leq x$, and $\|y\|_1 \leq (1 + \|(A \mid -b)\|_{1,\infty})^{r+1}$. We reformulate Theorem 7 as follows.

▶ **Lemma 8.** *If $B \in \mathbb{Z}^{\ell \times k}$ has rank r and $b \in \mathbb{Z}^\ell$, then there exist $c_1, \ldots, c_s \in \mathbb{N}^k$, $C \in \mathbb{N}^{k \times t}$ with $\|c_i\|_1, \|C\|_{\infty,1} \leq (1+\|B\|_{1,\infty}+\|b\|)^{r+1}$ such that $\{x \in \mathbb{N}^k \mid Bx = b\} = \{c_1, \ldots, c_s\}+C\mathbb{N}^t$.*

We want to apply Lemma 8 in a situation where we have no bound on $\|B\|_{1,\infty}$, but only one on $\|B\|_\infty$. However, we will know that $\ell = 1$, which allows us to bound magnitudes in terms of $\|B\|_\infty$ in the following lemma. Then, Proposition 6 is straightforward to show.

▶ **Lemma 9.** *If $B \in \mathbb{Z}^{1 \times k}$ and $b \in \mathbb{Z}$ with $\|B\|_\infty, |b| \leq M$, then we have a decomposition $\{x \in \mathbb{N}^k \mid Bx = b\} = \{c_1, \ldots, c_s\} + C\mathbb{N}^t$ where $\|c_i\|_1$ and $\|C\|_{\infty,1}$ are at most $(M + 1)^4$.*

4.3 Tameness of free products

This section is devoted to the proof of the following Proposition.

▶ **Proposition 10.** *If G_0 and G_1 are knapsack tame, then so is $G_0 * G_1$.*

Let $G = G_0 * G_1$. Suppose that for $i \in \{0, 1\}$ the group G_i is generated by A_i, where $A_i^{-1} = A_i$ and let $A = A_0 \uplus A_1$. Recall that every $g \in G$ can be written uniquely as $g = g_1 \cdots g_n$ where $g_i \in (G_0 \setminus \{1\}) \cup (G_1 \setminus \{1\})$ for each $i \in [1, n]$ and where $g_j \in G_t$ iff $g_{j+1} \in G_{1-t}$ for $j \in [1, n-1]$. We call g *cyclically reduced* if either $n \in \{0, 1\}$ or $n \geq 2$ and for some $t \in \{0, 1\}$, either $g_1 \in G_t$ and $g_n \in G_{1-t}$ or $g_1, g_n \in G_t$ and $g_n g_1 \neq 1$.

Every word $w \in A^*$ has a (possibly empty) unique factorization into maximal factors from $A_0^+ \cup A_1^+$, which we call *syllables*. By $\|w\|$, we denote the number of syllables of w. The word w is *reduced* if none of its syllables represents 1 (in G_0 resp. G_1). We define the maps $\lambda, \rho \colon A^+ \to A^+$ ("rotate left/right"), where for each word $w \in A^+$ with its factorization $w = w_1 \cdots w_m$ into syllables, we set $\lambda(w) = w_2 \cdots w_m w_1$ and $\rho(w) = w_m w_1 w_2 \cdots w_{m-1}$.

Consider a word $w = w_1 \cdots w_m \in A^*$, where for each $i \in [1, m]$, we have $w_i \in A_j^+$ for some $j \in \{0, 1\}$ (we allow $w_i, w_{i+1} \in A_j^+$). A *cancellation* is a subset $C \subseteq 2^{[1,m]}$ that is

- *a partition*: $\bigcup_{I \in C} I = [1, m]$ and $I \cap J = \emptyset$ for any $I, J \in C$ with $I \neq J$.
- *consistent*: for each $I \in C$, there is an $i \in \{0, 1\}$ such that $w_j \in A_i^*$ for all $j \in I$.
- *cancelling*: if $\{i_1, \ldots, i_\ell\} \in C$ with $i_1 < \cdots < i_\ell$, then $w_{i_1} \cdots w_{i_\ell}$ represents 1 in G.
- *well-nested*: there are no $I, J \in C$ with $i_1, i_2 \in I$, $j_1, j_2 \in J$ and $i_1 < j_1 < i_2 < j_2$.
- *maximal*: if $w_i, w_{i+1} \in A_j^+$ for $j \in \{0, 1\}$ then there is an $I \in C$ with $i, i + 1 \in I$.

Since C can be regarded as a hypergraph on $[1, m]$, the elements of C will be called *edges*. It is not hard to show that a word w admits a cancellation if and only if $w = 1$ in G.

Consider now an exponent equation

$$h_0 g_1^{x_1} h_1 \cdots g_k^{x_k} h_k = 1, \tag{1}$$

of size n, where g_i is represented by $u_i \in A^*$ for $i \in [1, k]$ and h_i is represented by $v_i \in A^*$ for $i \in [0, k]$. Then clearly $\sum_{i=0}^k |v_i| + \sum_{i=1}^k |u_i| \leq n$. Let $S \subseteq \mathbb{N}^k$ be the set of all solutions to (1). Of course, when showing that S has a polynomial magnitude, we may assume that $g_i \neq 1$ for any $i \in [1, k]$. Moreover, we lose no generality by assuming that all words u_i, $i \in [1, k]$ and v_i, $i \in [0, k]$ are reduced. Furthermore, we may assume that each g_i is cyclically reduced.

Indeed, if some g_i is not cyclically reduced, we can write $g_i = f^{-1}gf$ for some cyclically reduced g and replace h_{i-1}, g_i, and h_i by $h_{i-1}f^{-1}$, $g = fg_if^{-1}$, and fh_i, respectively. This does not change the solution set because $h_{i-1}f^{-1}(fg_if^{-1})^{x_i}fh_i = h_{i-1}g_i^{x_i}h_i$. Moreover, if we do this replacement for each g_i that is not cyclically reduced, we increase the size of the instance by at most $2|g_1| + \cdots + 2|g_k| \leq 2n$ (note that $|g| = |g_i|$). Applying this argument again, we may even assume that $u_i \in A_0^+ \cup A_1^+ \cup A_0^+A^*A_1^+ \cup A_1^+A^*A_0^+$ for every $i \in [1,k]$. Note that λ and ρ are bijections on words of this form.

Consider a solution $(x_1, \ldots, x_k)$ to (1). Then the word

$$w = v_0u_1^{x_1}v_1 \cdots u_k^{x_k}v_k \tag{2}$$

represents 1 in $G = G_0 * G_1$. We factorize each v_i, $i \in [0,k]$, and each u_i, $i \in [1,k]$, into its syllables. These factorizations define a factorization $w = w_1 \cdots w_m$ and we call this the *block factorization* of w. This is the coarsest refinement of the factorization $w = v_0u_1^{x_1}v_1 \cdots u_k^{x_k}v_k$ and of w's factorization into syllables. The numbers $1, 2, \ldots, m$ are the *blocks* of w. We fix this factorization $w = w_1 \cdots w_m$ for the rest of this section.

Certified solutions. In the representation $v_0u_1^{\mathsf{x}_1}v_1 \cdots u_k^{\mathsf{x}_k}v_k = 1$ of (1), the words $u_1, \ldots, u_k$ are called the *cycles*. If $u_i \in A_0^+ \cup A_1^+$, the cycle u_i is said to be *simple* and otherwise *mixed* (note that $u_i = \varepsilon$ cannot happen because $g_i \neq 1$). Let p be a block of w. If w_p is contained in some $u_i^{x_i}$ for a cycle u_i, then p is a *u_i-blocks* or *block from u_i*. If w_p is contained in some v_i, then p is a *v_i-block* or a *block from v_i*. A *certified solution* is a pair (x, C), where x is a solution to (1) and C is a cancellation of the word w as in (2). An edge $I \in C$ is called *standard* if $|I| = 2$ and the two blocks in I are from mixed cycles. Intuitively, the following tells us that in a cancellation, most edges are standard.

▶ **Lemma 11.** *Let C be a cancellation and u_i be a mixed cycle. Then there are at most $n + 3k + 1$ non-standard edges $I \in C$ containing a u_i-block.*

Mixed periods From now on, for each $i \in [1,k]$, we use e_i to denote the i-th unit vector in $\mathbb{N}^k$, i.e. the vector with 1 in the i-th coordinate and 0 otherwise. A *mixed period* is a vector $\pi \in \mathbb{N}^k$ of the form $\|u_j\| \cdot e_i + \|u_i\| \cdot e_j$, where u_i and u_j are mixed cycles. Let $\mathbb{P} \subseteq \mathbb{N}^k$ be the set of mixed periods. Note that $|\mathbb{P}| \leq k^2$.

We will need a condition that guarantees that a given period $\pi \in \mathbb{P}$ can be added to a solution x to obtain another solution. Suppose we have two blocks p and q for which we know that if we insert a string f_1 to the left of w_p and a string f_2 to the right of w_q and f_1f_2 cancels to 1 in G, then the whole word cancels to 1. Which string would we insert to the left of w_p and to the right of w_q if we build the solution $x + \pi$?

Suppose p is a u_i-block and q is a u_j-block. Moreover, let r be the first (left-most) u_i-block and let s be the last (right-most) u_j-block. If we add $\|u_j\| \cdot e_i$ to x, this inserts $\lambda^{p-r}(u_i^{\|u_j\|})$ to the left of w_p: Indeed, in the case $p = r$, we insert $u_i^{\|u_j\|}$; and when p moves one position to the right, the inserted string is rotated once to the left. Similarly, if we add $\|u_i\| \cdot e_j$ to x, we insert $\rho^{s-q}(u_j^{\|u_i\|})$ to the right of w_q: This is clear for $q = s$ and decrementing q means rotating the inserted string to the right. This motivates the following definition:

Let (x, C) be a certified solution and let u_i and u_j be mixed cycles with $i < j$. Moreover, let $r \in [1,m]$ be the left-most u_i-block and let $s \in [1,m]$ be the right-most u_j-block. Then the mixed period $\pi = \|u_j\| \cdot e_i + \|u_i\| \cdot e_j$ is *compatible with* (x, C) if there are a u_i-block p and a u_j-block q such that

$$\{p, q\} \in C \text{ and } \lambda^{p-r}(u_i^{\|u_j\|})\rho^{s-q}(u_j^{\|u_i\|}) \text{ represents 1 in } G. \tag{3}$$

With $\mathbb{P}(x, C)$, we denote the set of mixed periods that are compatible with (x, C). One might wonder why we require an edge $\{p, q\} \in C$. In order to guarantee that $\lambda^{p-r}(u_i^{\|u_j\|})$ and $\rho^{s-q}(u_j^{\|u_i\|})$ can cancel, it would be sufficient to merely forbid edges $I \in C$ that intersect $[p, q]$ and contain a block outside of $[p-1, q+1]$. However, this weaker condition can become false when we insert other mixed periods. Our stronger condition is preserved, which implies:

▶ **Lemma 12.** *Let (x, C) be a certified solution. Then every $x' \in x + \mathbb{P}(x, C)^{\oplus}$ is a solution.*

Let $M \subseteq [1, k]$ be the set of $i \in [1, k]$ such that u_i is a mixed cycle and $\|x\|_{\mathsf{m}} = \max_{i \in M} x_i$.

▶ **Lemma 13.** *There is a polynomial q such that the following holds. For every certified solution (x, C) with $\|x\|_{\mathsf{m}} > q(n)$, there exists a mixed period $\pi \in \mathbb{P}$ and a certified solution (x', C') such that $x = x' + \pi$, $\pi \in \mathbb{P}(x', C')$, and $\mathbb{P}(x, C) \subseteq \mathbb{P}(x', C')$.*

Proof. We show that the lemma holds if $q(n) \geq (n + 3k + 1) + kn^2$. (Recall that $k \leq n$.) Let (x, C) be a certified solution with $\|x\|_{\mathsf{m}} > q(n)$. Then there is a mixed cycle u_i such that $x_i > q(n)$ and hence $u_i^{x_i}$ consists of more than $q(n)$ blocks. Let $D \subseteq C$ be the set of all edges $I \in C$ that contain a block from u_i. It is not hard to show that an edge can contain at most one block per mixed cycle. Hence, we have $|D| > q(n)$ and, by Lemma 11, D contains more than kn^2 standard edges. Therefore, there must exist a mixed cycle u_j such that the set $E \subseteq D$ of standard edges $I \in D$ that consist of one block from u_i and one block from u_j satisfies $|E| > n^2$. Let B_i (resp., B_j) be the set of blocks from u_i (resp., u_j) contained in some edge $I \in E$. One can show that B_i and B_j are intervals of size more than n^2.

We only deal with the case $i < j$, the case $i > j$ can be done similarly. Let us take a subinterval $[p', p]$ of B_i such that $p - p' = \|u_i\| \cdot \|u_j\| \leq n^2$. By well-nestedness and since B_j is an interval, the neighbors (with respect to the edges from E) of $[p', p]$ form an interval $[q, q'] \subseteq B_j$ as well, and we have $p - p' = q' - q = \|u_i\| \cdot \|u_j\|$. Moreover, we have an edge $\{p - \ell, q + \ell\} \in E$ for each $\ell \in [0, p - p']$. In particular, $w_{p'} w_{p'+1} \cdots w_{p-1} w_{q+1} \cdots w_{q'}$ represents 1 in G.

Let r be the left-most u_i-block and let s be the right-most u_j-block. Then, as shown before the definition of compatibility, we have

$$\lambda^{p-r}(u_i^{\|u_j\|}) = w_{p'} w_{p'+1} \cdots w_{p-1} \text{ and } \rho^{s-q}(u_j^{\|u_i\|}) = w_{q+1} w_{q+1} \cdots w_{q'}.$$

Therefore, $\lambda^{p-r}(u_i^{\|u_j\|}) \rho^{s-q}(u_j^{\|u_i\|})$ represents 1 in G and $\{p, q\}$ witnesses compatibility of $\pi = \|u_j\| \cdot e_i + \|u_i\| \cdot e_j$ with (x, C). Hence, $\pi \in \mathbb{P}(x, C)$.

Let $x' = x - \pi$. We remove the factors $w_{p'} \cdots w_{p-1}$ and $w_{q+1} \cdots w_{q'}$ from w. Then, the remaining blocks spell $w' = v_0 u_1^{x_1'} v_1 \cdots u_k^{x_k'} v_k$. Indeed, recall that removing from a word y^t any factor of length $\ell \cdot |y|$ will result in the word $y^{t-\ell}$. Moreover, let C' be the set of edges that agree with C on the remaining blocks. By the choice of the removed blocks, it is clear that C' is a cancellation for w'. Hence, (x', C') is a certified solution.

It remains to verify $\mathbb{P}(x, C) \subseteq \mathbb{P}(x', C')$. First note that for every mixed cycle u_ℓ, all u_ℓ-blocks that remain in w' change their position relative to the left-most and the right-most u_ℓ-block by a difference that is divisible by $\|u_\ell\|$ (if $i \neq \ell \neq j$ then these relative positions do not change at all). Note that the expression $\lambda^{p-r}(u_i^{\|u_j\|})$ is not altered when $p - r$ changes by a difference divisible by $\|u_i\|$, and an analogous fact holds for $\rho^{s-q}(u_j^{\|u_i\|})$. Hence, the edge in C' that corresponds to the C-edge $\{p, q\}$ is a witness for $\pi \in \mathbb{P}(x', C')$. Moreover, for all other mixed periods $\pi' \in \mathbb{P}(x, C) \setminus \{\pi\}$ that are witnessed by an edge $\{t, u\} \in C$, the blocks t and u do not belong to $[p', p-1] \cup [q+1, q']$. Therefore, the corresponding edge in C' exists and serves as a witness for $\pi' \in \mathbb{P}(x', C')$. ◀

Repeated application of Lemma 13 now yields:

▶ **Lemma 14.** *There exists a polynomial q such that the following holds. For every solution $x \in \mathbb{N}^k$, there exists a certified solution (x', C') such that $\|x'\|_{\mathsf{m}} \leq q(n)$ and $x \in x' + \mathbb{P}(x', C')^{\oplus}$.*

We are now ready to prove Proposition 10 and thus Theorem 5.

Proof of Proposition 10. Suppose that p_0 and p_1 are the polynomials guaranteed by the knapsack tameness of G_0 and G_1, respectively. Recall that $S \subseteq \mathbb{N}^k$ is the set of solutions to (1). We prove that there exists a polynomial p such that for every $x \in S$ there is a semilinear set $S' \subseteq \mathbb{N}^k$ of magnitude at most $p(n)$ such that $x \in S' \subseteq S$. This clearly implies that S has magnitude at most $p(n)$. First, we apply Lemma 14. It yields a polynomial q and a certified solution (x', C') with $\|x'\|_{\mathsf{m}} \leq q(n)$ such that $x \in x' + \mathbb{P}(x', C')^{\oplus}$. Let $w' = v_0 u_1^{x'_1} v_1 \cdots u_k^{x'_k} v_k$ and consider w' decomposed into blocks as we did above with w.

Let $T \subseteq [1, k]$ be the set of all $i \in [1, k]$ for which the cycle u_i is simple. Since C' is maximal, for each $i \in T$, all u_i-blocks are contained in one edge $I_i \in C'$. Note that an edge may contain the blocks of more than one simple cycle. We partition T into sets $T = T_1 \uplus \cdots \uplus T_t$ so that $i \in T$ and $j \in T$ belong to the same part if and only if the u_i-blocks and the u_j-blocks belong to the same edge of C, i.e. $I_i = I_j$.

For a moment, let us fix an $\ell \in [1, t]$ and let $I \in C'$ be the edge containing all u_i-blocks for all the $i \in T_\ell$. Moreover, let $T_\ell = \{i_1, \ldots, i_r\}$. The words $\bar{v}_j$ for $j \in [0, r]$ will collect those blocks that belong to I but are not u_{i_s}-blocks for any $s \in [1, r]$. Formally, $\bar{v}_0$ consists of all blocks that belong to I that are to the left of all u_{i_1}-blocks. Similarly, $\bar{v}_r$ is the concatenation of all blocks belonging to I that are to the right of all u_{i_r}-blocks. Finally, for $j \in [1, r-1]$, $\bar{v}_j$ consists of all blocks that belong to I and are to the right of all u_{i_j}-blocks and to the left of all $u_{i_{j+1}}$-blocks. By consistency of C', for some $s \in \{0, 1\}$, all the words $\bar{v}_j$ for $j \in [0, r]$ and the words u_{i_j} for $j \in [1, r]$ belong to A_s^* and thus represent elements of G_s. Since G_s is knapsack tame, we know that the set

$$S_\ell = \{z \in \mathbb{N}^k \mid \bar{v}_0 u_{i_1}^{z_{i_1}} \bar{v}_1 u_{i_2}^{z_{i_2}} \bar{v}_2 \cdots u_{i_r}^{z_{i_r}} \bar{v}_r \text{ represents 1 in } G_s, \quad z_j = 0 \text{ for } j \notin T_\ell\}$$

has magnitude at most $p_s(n)$. Consider the vector $y \in \mathbb{N}^k$ with $y_i = 0$ for $i \in T$ and $y_i = x'_i$ for $i \in [1, k] \setminus T$ (i.e. when u_i is a mixed cycle). We claim that $S' = y + S_1 + \cdots S_t + \mathbb{P}(x', C')^{\oplus}$ has magnitude at most $q(n) + p_0(n) + p_1(n) + n$ and satisfies $x \in S' \subseteq S$.

First, since y and the members of $S_1, \ldots, S_t$ are non-zero on pairwise disjoint coordinates, the magnitude of $y + S_1 + \cdots + S_t$ is the maximum of $\|y\|$ and the maximal magnitude of $S_1, \ldots, S_\ell$. Hence, it is bounded by $q(n) + p_0(n) + p_1(n)$. The summand $\mathbb{P}(x', C')^{\oplus}$ contributes only periods, and their magnitude is bounded by n (recall that they are mixed periods). Thus, the magnitude of S' is at most $p(n) = q(n) + p_0(n) + p_1(n) + n$.

The cancelling property of (x', C') tells us that $x' - y$ is contained in $S_1 + \cdots + S_t$. By the choice of (x', C'), we have $x \in x' + \mathbb{P}(x', C')^{\oplus}$. Together, this means $x \in S'$. Hence, it remains to show $S' \subseteq S$. To this end, consider a vector $x'' \in y + S_1 + \cdots + S_t$. It differs from x' only in the exponents at simple cycles. Therefore, we can apply essentially the same cancellation to x'' as to x': we just need to adjust the edges containing the blocks of simple cycles. It is therefore clear that the resulting cancellation C'' has the same compatible mixed periods as C': $\mathbb{P}(x'', C'') = \mathbb{P}(x', C')$. Thus, by Lemma 12, we have $x'' + \mathbb{P}(x', C')^{\oplus} \subseteq S$. This proves $S' = y + S_1 + \cdots + S_t + \mathbb{P}(x', C')^{\oplus} \subseteq S$ and hence Proposition 10. ◀

4.4 LogCFL-hardness

It remains to show the lower bound in Theorem 2. If (A, I) is not complete, then (A, I) contains two non-adjacent vertices and thus $\mathbb{G}(A, I)$ contains an isomorphic copy of F_2, the

free group of rank two. Hence, we will show that knapsack and subset sum for F_2 are LogCFL-hard. Let $\{a, b\}$ be a generating set for F_2. Let $\theta\colon \{a, b, a^{-1}, b^{-1}\}^* \to F_2$ be the morphism that maps a word w to the group element represented by w. A *valence automaton* over a group G is a tuple $\mathcal{A} = (Q, \Sigma, \Delta, q_0, q_f)$ where Q, Σ, q_0, q_f are as in a finite automaton and Δ is a finite subset of $Q \times \Sigma^* \times G \times Q$. The *language accepted by* $\mathcal{A}$ is denoted $L(\mathcal{A})$ and consists of all words $w_1 \cdots w_n$ such that there is a computation $p_0 \xrightarrow{w_1,g_1} p_1 \to \cdots \to p_{n-1} \xrightarrow{w_n,g_n} p_n$ such that $(p_{i-1}, w_i, g_i, p_i) \in \Delta$ for $i \in [1, n]$ and $p_0 = q_0$, $p_n = q_f$, and $g_1 \cdots g_n = 1$ in G.

Fix a context-free language $L \subseteq \Sigma^*$ with a LogCFL-complete membership problem; such languages exist [10]. The Chomsky-Schützenberger theorem implies that there exists a valence automaton $\mathcal{A}$ over F_2 such that $L = L(\mathcal{A})$. Moreover, analyzing the proof of the Chomsky-Schützenberger theorem from [14] shows that there exists a constant c such that for every $w \in \Sigma^*$ we have: $w \in L(\mathcal{A}) = L$ if and only if there exists an accepting run of $\mathcal{A}$ for w of length at most $c \cdot |w|$. Given the word $w \in \Sigma$, it is easy to convert the valence automaton $\mathcal{A}$ into an acyclic automaton over $\{a, b, a^{-1}, b^{-1}\}^*$ that exhausts all computations of $\mathcal{A}$ of length at most $c \cdot |w|$. This yields the following:

▶ **Proposition 15.** *For F_2, the membership problem for acyclic automata is* LogCFL*-hard.*

▶ **Proposition 16.** *For F_2, knapsack and subset sum are* LogCFL*-hard.*

Proof. Let $\mathcal{A} = (Q, \{a, b, a^{-1}, b^{-1}\}, \Delta, q_0, q_f)$ be an acyclic automaton. We construct words $w, w_1, \ldots, w_m \in \{a, b, a^{-1}, b^{-1}\}$ such that $1 \in \theta(L(\mathcal{A})$ if and only if $\theta(w) \in \theta(w_1^* w_2^* \cdots w_m^*)$ if and only if $\theta(w) \in \theta(w_1^{e_1} w_2^{e_2} \cdots w_m^{e_m})$ for some $e_1, e_2, \ldots, e_m \in \{0, 1\}$. W.l.o.g. assume that $Q = \{1, \ldots, n\}$, where 1 is the initial state and n is the unique final state of $\mathcal{A}$.

Let $\alpha_i = a^i b a^{-i}$ for $i \in [1, n+2]$. It is well known that the α_i generate a free subgroup of rank $n+2$ in F_2 [20, Proposition 3.1]. Define the embedding $\varphi\colon F_2 \to F_2$ by $\varphi(a) = \alpha_{n+1}$ and $\varphi(b) = \alpha_{n+2}$. For a transition $t = (p, w, q) \in \Delta$ let $\tilde{t} = \alpha_p \varphi(w) \alpha_q^{-1}$. Let $\Delta = \{t_1, \ldots, t_m\}$ such that $t_i = (p, a, q)$ and $t_j = (q, b, r)$ implies $i < j$. Since $\mathcal{A}$ is acyclic, such an enumeration must exist. Together with the fact that the α_i generate a free group, it follows that $1 \in \theta(L(\mathcal{A})$ if and only if $\theta(\alpha_1 \alpha_n^{-1}) \in \theta(\tilde{t}_1^* \tilde{t}_2^* \cdots \tilde{t}_m^*)$ if and only if $\theta(\alpha_1 \alpha_n^{-1}) \in \theta(\tilde{t}_1^{e_1} \tilde{t}_2^{e_2} \cdots \tilde{t}_m^{e_m})$ for some $e_1, e_2, \ldots, e_m \in \{0, 1\}$. ◀

5 NP-completeness

In [19], the authors proved that knapsack for the graph group $\mathbb{G}(\mathsf{C4}) \cong F_2 \times F_2$ is NP-complete. Here we extend this result to all graph groups $\mathbb{G}(A, I)$ where (A, I) is not a transitive forest. An *acyclic loop automaton* is a finite automaton $\mathcal{A} = (Q, \Sigma, \Delta, q_0, q_f)$ such that there exists a linear order $\preceq$ on Δ having the property that for all $(p, u, q), (q, v, r) \in \Delta$ it holds $(p, a, q) \preceq (q, b, r)$. Thus, an acyclic loop automaton is obtained from an acyclic automaton by attaching to some of the states a unique loop. For a trace monoid $\mathbb{M}(A, I)$, *intersection nonemptiness for acyclic loop automata* is the following computational problem:

Input: Two acyclic loop automata $\mathcal{A}_1$, $\mathcal{A}_2$ over the input alphabet A.
Question: Does $[L(\mathcal{A}_1)]_I \cap [L(\mathcal{A}_2)]_I \neq \emptyset$ hold?

Aalbersberg and Hoogeboom [1] proved that for the trace monoid $\mathbb{M}(\mathsf{P4})$, intersection nonemptiness for arbitrary finite automata is undecidable. We use their technique to show:

▶ **Lemma 17.** *For $\mathbb{M}(\mathsf{P4})$, intersection nonemptiness for acyclic loop automata is* NP*-hard.*

Proof. We give a reduction from 3SAT. Let $\varphi = \bigwedge_{i=1}^m C_i$ where for every $i \in [1, m]$, $C_i = (L_{i,1} \vee L_{i,2} \vee L_{i,3})$ is a clause consisting of three literals. Let $x_1, \ldots, x_n$ be the boolean variables that occur in φ. Every literal $L_{i,j}$ belongs to $\{x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$.

Let $p_1, p_2, \ldots, p_n$ be a list of the first n prime numbers. So, for each boolean variable x_i we have the corresponding prime number p_i. We encode a valuation $\beta\colon \{x_1, \ldots, x_n\} \to \{0,1\}$ by any natural number N such that $N \equiv 0 \bmod p_i$ if and only if $\beta(x_i) = 1$. For a positive literal x_i let $S(x_i) = \{p_i \cdot n \mid n \in \mathbb{N}\}$ and for a negative literal $\neg x_i$ let $S(\neg x_i) = \{p_i \cdot n + r \mid n \in \mathbb{N}, r \in [1, p_i - 1]\}$. Moreover, for every $i \in [1,m]$ let $S_i = S(L_{i,1}) \cup S(L_{i,2}) \cup S(L_{i,3})$. Thus, S_i is the set of all numbers that encode a valuation that makes the clause C_i true. Hence, the set $S = \bigcap_{i=1}^{n} S_i$ encodes the set of all valuations that make φ true.

We first construct an acyclic loop automaton $\mathcal{A}_1$ with $L(\mathcal{A}_1) = \prod_{i=1}^{m} \{a(bc)^{N_i} d \mid N_i \in S_i\}$. Note that φ is satisfiable iff $[L(\mathcal{A}_1)]_I$ contains a trace from $[\{(a(bc)^N d)^m \mid N \in \mathbb{N}\}]_I$. We will ensure this property with a second acyclic loop automaton $\mathcal{A}_2$ that satisfies the equality $L(\mathcal{A}_2) = b^*(ad(bc)^*)^{m-1}adc^*$. We claim that $[L(\mathcal{A}_1)]_I \cap [L(\mathcal{A}_2)]_I = [\{(a(bc)^N d)^m \mid N \in S\}]_I$.

First assume that $w \equiv_I (a(bc)^N d)^m$ for some $N \in S$. We have $w \equiv_I (a(bc)^N d)^m \equiv_I b^N (ad(bc)^N)^{m-1} adc^N$ and thus $[w]_I \in [L(\mathcal{A}_2)]_I$. Moreover, since $N \in S$ we get $[w]_I \in [L(\mathcal{A}_1)]_I$. For the other direction, let $[w]_I \in [L(\mathcal{A}_1)]_I \cap [L(\mathcal{A}_2)]_I$. Thus

$$w \equiv_I \prod_{i=1}^{m} (a(bc)^{N_i} d) \equiv_I b^{N_1} \Bigg(\prod_{i=1}^{m-1} adc^{N_i} b^{N_{i+1}} \Bigg) adc^{N_m},$$

where $N_i \in S_i$ for $i \in [1,m]$. Moreover, $[w]_I \in [L(\mathcal{A}_2)]_I$ yields $k_0, k_1, \ldots, k_{m-1}, k_m \geq 0$ with

$$b^{N_1} \Bigg(\prod_{i=1}^{m-1} adc^{N_i} b^{N_{i+1}} \Bigg) adc^{N_m} \equiv_I b^{k_0} \Bigg(\prod_{i=1}^{m-1} ad(bc)^{k_i} \Bigg) adc^{k_m} \equiv_I b^{k_0} \Bigg(\prod_{i=1}^{m-1} (adb^{k_i} c^{k_i}) \Bigg) adc^{k_m}.$$

Since every symbol depends on a or on d, this identity implies $N_i = N_{i+1}$ for $i \in [1, m-1]$. Thus, $[w]_I \in [\{(a(bc)^N d)^m \mid N \in S\}]_I$. ◀

For a graph group $\mathbb{G}(A, I)$ the *membership problem for acyclic loop automata* is the following computational problem:

Input: An acyclic loop automaton $\mathcal{A}$ over the input alphabet $A \cup A^{-1}$.
Question: Is there a word $w \in L(\mathcal{A})$ such that $w = 1$ in $\mathbb{G}(A, I)$?

It is straightforward to reduce the intersection emptiness problem for acyclic loop automata over $\mathbb{M}(A, I)$ to the membership problem for acyclic loop automata over $\mathbb{G}(A, I)$.

▶ **Lemma 18.** *For $\mathbb{G}(\mathsf{P4})$, the membership problem for acyclic loop automata is* NP*-hard.*

We can now use a construction from [17] to reduce membership for acyclic loop automata over $\mathbb{G}(\mathsf{P4})$ to knapsack for $\mathbb{G}(\mathsf{P4})$.

▶ **Lemma 19.** *Knapsack for the graph group $\mathbb{G}(\mathsf{P4})$ is* NP*-hard.*

▶ **Theorem 20.** *If (A, I) is an independence alphabet, which is not a transitive forest, then knapsack for the graph group $\mathbb{G}(A, I)$ is* NP*-complete.*

Proof. If (A, I) is not a transitive forest, then $\mathsf{P4}$ or $\mathsf{C4}$ is an induced subgraph of (A, I) [27]. Thus, $\mathbb{G}(\mathsf{P4})$ or $\mathbb{G}(\mathsf{C4}) \cong F_2 \times F_2$ is a subgroup of $\mathbb{G}(A, I)$. Hence, NP-hardness of knapsack for $\mathbb{G}(A, I)$ follows from [19] or Lemma 19. ◀

6 An open problem

In [19] the authors proved that (uncompressed) subset sum for $\mathbb{G}(\mathsf{C4})$ is NP-complete as well. It remains open whether subset sum is NP-hard also for $\mathbb{G}(\mathsf{P4})$. Our proof for the NP-hardness of knapsack for $\mathbb{G}(\mathsf{P4})$ makes essential use of exponentially large exponents and hence cannot be used for subset sum.

References

1 I. J. Aalbersberg and H. J. Hoogeboom. Characterizations of the decidability of some problems for regular trace languages. *Mathematical Systems Theory*, 22:1–19, 1989.

2 I. Agol. The virtual Haken conjecture. With an appendix by Agol, Daniel Groves, and Jason Manning. *Documenta Mathematica*, 18:1045–1087, 2013.

3 L. Babai, R. Beals, J. Cai, G. Ivanyos, and E. M.Luks. Multiplicative equations over commuting matrices. In *Proceedings of SODA 1996*, pages 498–507. ACM/SIAM, 1996.

4 M. Bestvina and N. Brady. Morse theory and finiteness properties of groups. *Inventiones Mathematicae*, 129(3):445–470, 1997.

5 J. Crisp and B. Wiest. Embeddings of graph braid and surface groups in right-angled Artin groups and braid groups. *Algebraic & Geometric Topology*, 4:439–472, 2004.

6 V. Diekert. *Combinatorics on Traces*, volume 454 of *Lecture Notes in Computer Science*. Springer-Verlag, 1990.

7 M. Elberfeld, A. Jakoby, and T. Tantau. Algorithmic meta theorems for circuit classes of constant and logarithmic depth. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:128, 2011.

8 E. Frenkel, A. Nikolaev, and A. Ushakov. Knapsack problems in products of groups. *Journal of Symbolic Computation*, 74:96–108, 2016.

9 R. Ghrist and V. Peterson. The geometry and topology of reconfiguration. *Advances in Applied Mathematics*, 38(3):302–323, 2007.

10 S. Greibach. The hardest context-free language. *SIAM Journal on Computing*, 2(4):304–310, 1973.

11 C. Haase. *On the complexity of model checking counter automata*. PhD thesis, University of Oxford, St Catherine's College, 2011.

12 F. Haglund and D. T. Wise. Coxeter groups are virtually special. *Advances in Mathematics*, 224(5):1890–1903, 2010.

13 B. Jenner. Knapsack problems for NL. *Information Processing Letters*, 54(3):169–174, 1995.

14 M. Kambites. Formal languages and groups as memory. *Communications in Algebra*, 37:193–208, 2009.

15 R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

16 D. König, M. Lohrey, and G. 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.

17 M. Lohrey and B. Steinberg. The submonoid and rational subset membership problems for graph groups. *Journal of Algebra*, 320(2):728–755, 2008.

18 M. Lohrey and G. Zetzsche. The complexity of knapsack in graph groups. Technical report, arXiv.org, 2015. `https://arxiv.org/abs/1610.00373`.

19 M. Lohrey and G. Zetzsche. Knapsack in graph groups, HNN-extensions and amalgamated products. In *Proceedings of STACS 2016*, volume 47 of *LIPIcs*, pages 50:1–50:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

20 R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory*. Springer-Verlag, 1977.

21 A. Myasnikov, A. Nikolaev, and A. Ushakov. Knapsack problems in groups. *Mathematics of Computation*, 84:987–1016, 2015.

22 C. H. Papadimitriou. On the complexity of integer programming. *Journal of the Association for Computing Machinery*, 28(4):765–768, 1981.

23 L. Pottier. Minimal solutions of linear diophantine systems : bounds and algorithms. In *Proceedings of RTA 1991*, volume 488 of *Lecture Notes in Computer Science*, pages 162–173. Springer-Verlag, 1991.

24 I. H. Sudborough. On the tape complexity of deterministic context–free languages. *Journal of the ACM*, 25(3):405–414, 1978.

25 H. Vollmer. *Introduction to Circuit Complexity*. Springer-Verlag, 1999.

26 D. T. Wise. Research announcement: the structure of groups with a quasiconvex hierarchy. *Electronic Research Announcements in Mathematical Sciences*, 16:44–55, 2009.

27 E. S. Wolk. A note on "The comparability graph of a tree". *Proceedings of the American Mathematical Society*, 16:17–20, 1965.