Knapsack in Graph Groups, HNN-Extensions and Amalgamated Products

Markus Lohrey[1] and Georg Zetzsche[2]

1 Universität Siegen, Germany
lohrey@eti.uni-siegen.de
2 Technische Universität Kaiserslautern, Germany
zetzsche@cs.uni-kl.de

Abstract

It is shown that the knapsack problem, which was introduced by Myasnikov et al. for arbitrary finitely generated groups, can be solved in NP for graph groups. This result even holds if the group elements are represented in a compressed form by SLPs, which generalizes the classical NP-completeness result of the integer knapsack problem. We also prove general transfer results: NP-membership of the knapsack problem is passed on to finite extensions, HNN-extensions over finite associated subgroups, and amalgamated products with finite identified subgroups.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Graph groups, HNN-extensions, amalgamated products, knapsack

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

In their paper [36], Myasnikov, Nikolaev, and Ushakov started the investigation of classical discrete optimization problems, which are classically formulated over the integers, for arbitrary in general non-commutative groups. Among other problems, they 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 prefered) 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 [24] 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 [17], 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) [14], and the same holds for knapsack, since the instance $x_1 \cdot g_1 + \cdots + x_k \cdot g_k = g$ has a solution if and only if it has a solution with $x_i \leq k \cdot (\max\{g_1, \ldots, g_k, g\})^3$ [37]. This allows to reduce unary knapsack to unary subset sum. See [21] for related results.

[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.

Conference title on which this volume is based on.
Editors: Billy Editor and Bill Editors; pp. 1–14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In [36] the authors encode elements of the finitely generated group G by words over the group generators and their inverses. For $G = \mathbb{Z}$ this representation corresponds to the unary encoding of integers. Among others, the following results were shown in [36]:

- Subset sum and knapsack can be solved in polynomial time for every hyperbolic group.
- Subset sum for a virtually nilpotent group (a finite extension of a nilpotent group) can be solved in polynomial time.
- For the following groups, subset sum is NP-complete (whereas the word problem can be solved in polynomial time): free metabelian non-abelian groups of finite rank, the wreath product $\mathbb{Z} \wr \mathbb{Z}$, Thompson's group F, and the Baumslag-Solitar group $\mathrm{BS}(1,2)$.

Further results on knapsack and subset sum have been recently obtained in [27]:

- For a virtually nilpotent group, subset sum belongs to NL (nondeterministic logspace).
- There is a nilpotent group of class 2 (in fact, a direct product of sufficiently many copies of the discrete Heisenberg group $H_3(\mathbb{Z})$), for which knapsack is undecidable.
- The knapsack problem for the discrete Heisenberg group $H_3(\mathbb{Z})$ is decidable. In particular, together with the previous point it follows that decidability of knapsack is not preserved under direct products.
- There is a polycyclic group with an NP-complete subset sum problem.
- The knapsack problem is decidable for every co-context-free group.

The focus of this paper will be on the knapsack problem. We will prove that this problem can be solved in NP for every *graph group*. Graph groups are also known as right-angled Artin groups or free partially commutative groups. A graph group is specified by a finite simple graph. 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. Graph groups somehow interpolate between free groups and free abelian groups and can be seen as a group counterpart of trace monoids (free partially commutative monoids) that 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. [5, 8, 16].

To prove that knapsack belongs to NP for a graph group, we proceed in two steps: We first show that if an instance $g_1^{x_1} \cdots g_k^{x_k} = g$ has a solution in a graph group, then it has a solution, where every x_i is bounded exponentially in the input length (the total length of all words representing the group elements $g_1, \ldots, g_k, g$). We then guess the binary encodings of numbers $n_1, \ldots, n_k$ that are bounded by the exponential bound from the previous point and verify in polynomial time the identity $g_1^{n_1} \cdots g_k^{n_k} = g$. The latter problem is an instance of the so called *compressed word problem* for a graph group. This is the classical word problem, where the input group element is given succinctly by a so called *straight-line program* (SLP), which is a context-free grammar that produces a single word (here, a word over the group generators and their inverses). An SLP with n productions in Chomsky normal form can produce a string of length 2^n. Nevertheless, the compressed word problem for a fixed graph group can be solved in polynomial time, see [29] for more details.

In fact, our proof yields a stronger result: First, it yields an NP procedure for solving knapsack-like equations $h_0 g_1^{x_1} h_1 \cdots h_{k-1} g_k^{x_k} h_k = 1$, where some of the variables $x_1, \ldots, x_k$ are allowed to be identical. We call such an equation an *exponent equation*. Hence, we prove that solvability of exponent equations over a graph group belongs to NP.

Second, we show that the latter result even holds when the group elements $g_1, \ldots, g_k$ and $h_0, \ldots, h_k$ are given succinctly by SLPs; we speak of *solvability of compressed exponent equations*. This is interesting since the SLP-encoding of group elements corresponds in the case $G = \mathbb{Z}$ to the binary encoding of integers. Hence, membership in NP for solvability of compressed exponent equations over a graph group generalizes the classical NP membership for knapsack (over $\mathbb{Z}$) to a much wider class of groups.

Furthermore, we extend the class of groups for which solvability of knapsack (resp. compressed exponent equations) is in NP by proving general transfer results. Our first transfer result states that if H is a finite extension of G and solvability of compressed exponent equations (or knapsack) is in NP for G, then the same holds for H. This provides such algorithms for the abundant class of *virtually special groups*. These are finite extensions of subgroups of graph groups. Virtually special groups recently played a major role in a spectacular breakthrough in three-dimensional topology, namely the solution of the virtual Haken conjecture [1]. In the course of this development it turned out that the class of virtually special groups is very rich: It contains Coxeter groups [18], one-relator groups with torsion [41], fully residually free groups [41], and fundamental groups of hyperbolic 3-manifolds [1].

We also prove transfer results for HNN-extensions and amalgamated products with finite associated (resp. identified) subgroups in the case of the knapsack problem. These two constructions are of fundamental importance in combinatorial group theory [34]. Examples include Stallings' decomposition of groups with infinitely many ends [38] or the construction of virtually free groups [9]. Moreover, these constructions are known to preserve a wide range of important structural and algorithmic properties [2, 6, 19, 22, 23, 25, 26, 30, 31, 35].

A side product of our proof is that the set of all solutions $(x_1, \ldots, x_k) \in \mathbb{N}^k$ of an exponent equation $g_1^{x_1} \cdots g_k^{x_k} = g$ over a graph group is semilinear, and a semilinear representation can be produced effectively. This seems to be true for many groups, e.g., for all co-context-free groups [27]. On the other hand, for the discrete Heisenberg group $H_3(\mathbb{Z})$ solvability of exponent equations is decidable but the set of all solutions of an exponent equation is not semilinear; it is defined by a single quadratic Diophantine equation [27].

Finally, we complement our upper bounds by a new lower bound: Knapsack and subset sum are both NP-complete for a direct product of two free groups of rank two ($F_2 \times F_2$). This group is the graph group corresponding to a cycle of length four. NP-hardness already holds for the case that the input group elements are specified by words over the generators (for SLP-compressed words, NP-hardness already holds for $\mathbb{Z}$) and the exponent variables are allowed to take values in $\mathbb{Z}$ (instead of $\mathbb{N}$). NP-completeness of subset sum for $F_2 \times F_2$ solves an open problem from [15].

A full version of this work can be found in the arXiv [33].

Related work. The knapsack problem is a special case of the more general *rational subset membership problem*. A rational subset of a finitely generated monoid M is the homomorphic image in M of a regular language over the generators of M. In the rational subset membership problem for M the input consists of a rational subset $L \subseteq M$ (specified by a finite automaton) and an element $m \in M$ and it is asked whether $m \in L$. It was shown in [32] that the rational subset membership problem for a graph group G is decidable if and only if the corresponding graph has (i) no induced cycle on four nodes (C4) and (ii) no induced path on four nodes (P4). For the decidable cases, the precise complexity is open.

Knapsack for G can be also viewed as the question, whether a word equation $z_1 z_2 \cdots z_n = 1$, where $z_1, \ldots, z_n$ are variables, together with constraints of the form $\{g^n \mid n \geq 0\}$ for the variables has a solution in G. Such a solution is a mapping $\varphi : \{z_1, \ldots, z_n\} \to G$ such that $\varphi(z_1 z_2 \cdots z_n)$ evaluates to 1 in G and all constraints are satisfied. For another class of constraints (so called normalized rational constraints, which do not cover constraints of the form $\{g^n \mid n \geq 0\}$), solvability of general word equations was shown to be decidable

(PSPACE-complete) for graph groups by Diekert and Muscholl [12]. This result was extended in [11] to a transfer theorem for graph products. A graph product is specified by a finite simple graph, where every node is labelled with a group. The associated group is obtained from the free product of all vertex groups by allowing elements from adjacent groups to commute. Note that decidability of knapsack is not preserved under graph products. It is even not preserved under direct products, see the above mentioned results from [27].

2 Words and Straight-Line Programs

For a word w we denote with $\mathrm{alph}(w)$ the set of symbols occuring in w. The length of the word w is $|w|$. A *straight-line program*, briefly *SLP*, is basically a context-free grammar that produces exactly one string. To ensure this, the grammar has to be acyclic and deterministic (every variable has a unique production where it occurs on the left-hand side). Formally, an SLP is a tuple $\mathcal{G} = (V, \Sigma, \mathrm{rhs}, S)$, where V is a finite set of variables (or nonterminals), Σ is the terminal alphabet, $S \in V$ is the start variable, and rhs maps every variable to a right-hand side $\mathrm{rhs}(A) \in (V \cup \Sigma)^*$. We require that there is a linear order $<$ on V such that $B < A$, whenever $B \in N \cap \mathrm{alph}(\mathrm{rhs}(A))$. Every variable $A \in V$ derives to a unique string $\mathrm{val}_{\mathcal{G}}(A)$ by iteratively replacing variables by the corresponding right-hand sides, starting with A. Finally, the string derived by $\mathcal{G}$ is $\mathrm{val}(\mathcal{G}) = \mathrm{val}_{\mathcal{G}}(S)$.

Let $\mathcal{G} = (V, \Sigma, \mathrm{rhs}, S)$ be an SLP. The *size* of $\mathcal{G}$ is $|\mathcal{G}| = \sum_{A \in V} |\mathrm{rhs}(A)|$, i.e., the total length of all right-hand sides. A simple induction shows that for every SLP $\mathcal{G}$ of size m one has $|\mathrm{val}(\mathcal{G})| \leq \mathcal{O}(3^{m/3}) \subseteq 2^{\mathcal{O}(n)}$ [7, proof of Lemma 1]. On the other hand, it is straightforward to define an SLP $\mathcal{H}$ of size $2n$ such that $|\mathrm{val}(\mathcal{H})| \geq 2^n$. This justifies to see an SLP $\mathcal{G}$ as a compressed representation of the string $\mathrm{val}(\mathcal{G})$, and exponential compression rates can be achieved in this way. More details on SLPs can be found in [29].

3 Knapsack and Exponent Equations

We assume that the reader has some basic knowledge concerning (finitely generated) groups, see e.g. [34] 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

$$v_0 u_1^{x_1} v_1 u_2^{x_2} v_2 \cdots u_n^{x_n} v_n = 1,$$

where $u_1, u_2, \ldots, u_n, v_0, v_1, \ldots, v_n \in G$ are group elements that are given by finite words over the alphabet $A^{\pm 1}$ and $x_1, x_2, \ldots, x_n$ are not necessarily distinct variables. Such an exponent equation is solvable if there exists a mapping $\sigma : \{x_1, \ldots, x_n\} \to \mathbb{N}$ such that $v_0 u_1^{\sigma(x_1)} v_1 u_1^{\sigma(x_2)} v_2 \cdots u_n^{\sigma(x_n)} v_n = 1$ in the group G. *Solvability of exponent equations over* G is the following computational problem:

Input: An exponent equation E over G (where elements of G are specified by words over the group generators and their inverses).
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 $u_1^{x_1} u_2^{x_2} \cdots u_n^{x_n} u^{-1} = 1$, or, equivalently $u_1^{x_1} u_2^{x_2} \cdots u_n^{x_n} = u$, where the exponent variables $x_1, \ldots, x_n$ have to be pairwise different.

We will also study a compressed version of exponent equations over G, where elements of G are given by SLPs over $A^{\pm 1}$. A *compressed exponent equation* is an exponent equation $v_0 u_1^{x_1} v_1 u_2^{x_2} v_2 \cdots u_n^{x_n} v_n = 1$, where the group elements $u_1, u_2, \ldots, u_n, v_0, v_1, \ldots, v_n \in G$ are given by SLPs over the terminal alphabet $A^{\pm 1}$. The sum of the sizes of these SLPs is the size of the compressed exponent equation. Let us define *solvability of compressed exponent equations over G* as the following computational problem:

Input: A compressed exponent equation E over G.
Question: Is E solvable?

The *compressed knapsack problem* for G is defined analogously. Note that with this terminology, the classical knapsack problem for binary encoded integers is the compressed knapsack problem for the group $\mathbb{Z}$. The binary encoding of an integer can be easily transformed into an SLP over the alphabet $\{a, a^{-1}\}$ (where a is a generator of $\mathbb{Z}$) and vice versa. Thereby the number of bits in the binary encoding and the size of the SLP are linearly related.

It is a simple observation that the decidability and complexity of solvability of (compressed) exponent equations over G as well as the (compressed) knapsack 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 1. Since we are dealing with a group, one might also allow solution mappings $\sigma : \{x_1, \ldots, x_n\} \to \mathbb{Z}$ to the integers. But this variant of solvability of (compressed) exponent equations (knapsack, respectively) can be reduced to the above version, where σ maps to $\mathbb{N}$, by simply replacing a power $u_i^{x_i}$ by $u_i^{x_i}(u_i^{-1})^{y_i}$, where y_i is a fresh variable.

The goal of this paper is to prove the decidability of solvability of exponent equations for so called graph groups. We actually prove that solvability of compressed exponent equations for a graph group belongs to NP. Graph groups will be introduced in the next section.

4 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 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*; it is cancellative, i.e., $xy = xz$ or $yx = zx$ implies $y = z$. 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. For a language $L \subseteq A^*$ we denote with $[L]_I = \{u \in A^* \mid \exists v \in L : u \equiv_I v\}$ its *partially commutative closure*. The length of the trace $[u]_I$ is $|[u]_I| = |u|$ and its alphabet is $\mathrm{alph}([u]_I) = \mathrm{alph}(u)$. It is easy to see that these definition do not depend on the concrete word that represents the trace $[u]_I$. For subsets $B, C \subseteq A$ we write BIC for $B \times C \subseteq I$. If $B = \{a\}$ we simply write aIC. For traces s, t we write sIt for $\mathrm{alph}(s) I \mathrm{alph}(t)$. The empty trace $[\varepsilon]_I$ is the identity element of the monoid $\mathbb{M}(A, I)$ and is denoted by 1. A trace t is connected if we cannot factorize t as $t = uv$ with $u \neq 1 \neq v$ and uIv. For a trace $t \in \mathbb{M}(A, I)$ let $\rho(t)$ be the number of prefixes of t. We will use the following statement from [4].

▶ **Lemma 2.** *Let $t \in \mathbb{M}(A, I)$ be a trace of length n. Then $\rho(t) \in \mathcal{O}(n^\alpha)$, where α is the size of a largest clique of the complementary graph $(A, I)^c = (A, (A \times A) \setminus I)$.*

We define the group $\mathbb{G}(A, I) = \langle A \mid ab = ba \ ((a, b) \in I) \rangle$. Such a group is called a *graph group*, or *right-angled Artin group*, or *free partially commutative group*. Here, we use the term graph group. We represent elements of $\mathbb{G}(A, I)$ by traces over an extended independence alphabet. For this, let $A^{-1} = \{a^{-1} \mid a \in A\}$ be a disjoint copy of the alphabet A, and let $A^{\pm 1} = A \cup A^{-1}$. We define $(a^{-1})^{-1} = a$ and for a word $w = a_1 a_2 \cdots a_n$

with $a_i \in A^{\pm 1}$ we define $w^{-1} = a_n^{-1} \cdots a_2^{-1} a_1^{-1}$. This defines an involution (without fixed points) on $(A^{\pm 1})^*$. We extend the independence relation I to $A^{\pm 1}$ by $(a^x, b^y) \in I$ for all $(a, b) \in I$ and $x, y \in \{-1, 1\}$. For a trace $t = [u]_I$ ($u \in (A^{\pm 1})^*$) we can then define $t^{-1} = [u^{-1}]_I$. This is well-defined, since $u \equiv_I v$ implies $u^{-1} \equiv_I v^{-1}$. There is a canonical surjective morphism $h : \mathbb{M}(A^{\pm 1}, I) \to \mathbb{G}(A, I)$ that maps every symbol $a \in A^{\pm 1}$ to the corresponding group element. Of course, h is not injective, but we can easily define a subset $\mathrm{IRR}(A^{\pm 1}, I) \subseteq \mathbb{M}(A^{\pm 1}, I)$ of *irreducible traces* such that h restricted to $\mathrm{IRR}(A^{\pm 1}, I)$ is bijective. The set $\mathrm{IRR}(A^{\pm 1}, I)$ consists of all traces $t \in \mathbb{M}(A^{\pm 1}, I)$ such that t does not contain a factor $[aa^{-1}]_I$ with $a \in A^{\pm 1}$, i.e., there do not exist $u, v \in \mathbb{M}(A^{\pm 1}, I)$ and $a \in A^{\pm 1}$ such that in $\mathbb{M}(A^{\pm 1}, I)$ we have a factorization $t = u[aa^{-1}]_I v$. For every trace t there exists a corresponding *irreducible normal form* that is obtained by removing from t factors $[aa^{-1}]_I$ with $a \in A^{\pm 1}$ as long as possible. It can be shown that this reduction process is terminating (which is trivial since it reduces the length) and confluent (in [28] a more general confluence lemma for graph products of monoids is shown). Hence, the irreducible normal form of t does not depend on the concrete order of reduction steps. For a group element $g \in \mathbb{G}(A, I)$ we denote with $|g|$ the length of the unique trace $t \in \mathrm{IRR}(A^{\pm 1}, I)$ such that $h(t) = g$.

5 Three Auxiliary Results

Based on Levi's lemma for traces (see e.g. [13, p. 74]) one can show the following factorization result for powers of a connected trace.

▶ **Lemma 3.** *Let $u \in \mathbb{M}(A, I) \setminus \{1\}$ be a connected trace and $m \in \mathbb{N}$, $m \geq 2$. Then, for all $x \in \mathbb{N}$ and traces $y_1, \ldots, y_m$ we have: $u^x = y_1 y_2 \cdots y_m$ if and only if there exist traces $p_{i,j}$ $(1 \leq j < i \leq m)$, s_i $(1 \leq i \leq m)$ and $x_i, c_j \in \mathbb{N}$ $(1 \leq i \leq m,\ 1 \leq j \leq m-1)$ such that:*

- *$y_i = (\prod_{j=1}^{i-1} p_{i,j}) u^{x_i} s_i$ for all $1 \leq i \leq m$,*
- *$p_{i,j} I p_{k,l}$ if $j < l < k < i$ and $p_{i,j} I (u^{x_k} s_k)$ if $j < k < i$*
- *$s_m = 1$ and for all $1 \leq j < m$, $s_j \prod_{i=j+1}^{m} p_{i,j} = u^{c_j}$*
- *$c_j \leq |A|$ for all $1 \leq j \leq m-1$,*
- *$x = \sum_{i=1}^{m} x_i + \sum_{i=1}^{m-1} c_i$.*

▶ Remark 4. In Section 6 we will apply Lemma 3 in order to replace an equation $u^x = y_1 y_2 \cdots y_m$, (where $x, y_1, \ldots, y_m$ are variables und u is a concrete connected trace) by an equivalent disjunction. Note that the length of all factors $p_{i,j}$ and s_i above is bounded by $|A| \cdot |u|$. Hence, one can guess these traces as well as the numbers $c_j \leq |A|$ (the guess results in a disjunction). We can also guess which of the numbers x_i are zero and which are greater than zero. After these guesses we can verify the independences $p_{i,j} I p_{k,l}$ $(j < l < k < i)$ and $p_{i,j} I (u^{x_k} s_k)$ $(j < k < i)$, and the identities $s_m = 1$, $s_j \prod_{i=j+1}^{m} p_{i,j} = u^{c_j}$ $(1 \leq j < m)$. If one of them does not hold, the specific guess does not contribute to the disjunction. In this way, we can replace the equation $u^x = y_1 y_2 \cdots y_m$ by a disjunction of formulas of the form

$$\exists x_i > 0\ (i \in K) : x = \sum_{i \in K}^{m} x_i + c \wedge \bigwedge_{i \in K} y_i = p_i u^{x_i} s_i \wedge \bigwedge_{i \in [1,m] \setminus K} y_i = p_i s_i,$$

where $K \subseteq [1, m]$, $c \leq |A| \cdot (m-1)$ and the p_i, s_i are concrete traces of length at most $|A| \cdot (m-1) \cdot |u|$. The number of disjuncts in the disjunction is not important for our purpose.

The second auxiliary result that we need is (recall that $\rho(t)$ is the number of prefixes of the trace t):

▶ **Lemma 5.** *Let $p, q, u, v, s, t \in \mathbb{M}(A, I)$ with $u \neq 1$ and $v \neq 1$ connected. Let $m = \max\{\rho(p), \rho(q), \rho(s), \rho(t)\}$ and $n = \max\{\rho(u), \rho(v)\}$. Then the set $L(p, u, s, q, v, t) := \{(x, y) \in \mathbb{N} \times \mathbb{N} \mid pu^x s = qv^y t\}$ is semilinear and is a union of $\mathcal{O}(m^8 \cdot n^{4|A|})$ many linear sets of the form $\{(a + bz, c + dz) \mid z \in \mathbb{N}\}$ with $a, b, c, d \in \mathcal{O}(m^8 \cdot n^{4|A|})$.*

The proof of Lemma 5 applies the theory of recognizable trace languages. We construct an automaton for the language $L = [pu^x s]_I \cap [qv^y t]_I$ with at most $4m^4 \cdot n^{2\cdot|A|}$ states. Then, we analyze the set of all lengths of words from L using results on unary finite automata [39].

Finally, we need a bound on the norm of a smallest vector in a certain kind of semilinear sets. We easily obtain this bound from a result by zur Gathen and Sieveking [40].

▶ **Lemma 6.** *Let $A \in \mathbb{Z}^{n\times m}$, $\overline{a} \in \mathbb{Z}^n$, $C \in \mathbb{N}^{k\times m}$, $\overline{c} \in \mathbb{N}^k$. Let β be an upper bound for the absolute value of all entries in A, $\overline{a}$, C, $\overline{c}$. The set $L = \{C\overline{z} + \overline{c} \mid \overline{z} \in \mathbb{N}^m, A\overline{z} = \overline{a}\} \subseteq \mathbb{N}^k$ is semilinear. Moreover, if $L \neq \emptyset$ then L contains a vector with all entries bounded by $\beta + n! \cdot m \cdot (m + 1) \cdot \beta^{n+1}$.*

6 Exponent Equations in Graph Groups

In this section, we prove the following two statements, where G is a fixed graph group:

- The set of solutions of an exponent equation over G is (effectively) semilinear.
- Solvability of compressed exponent equations over G belongs to NP.

In the next section, we will extend these results to the larger class of virtually special groups. We start with some definitions. As usual, we fix an independence alphabet (A, I). In the following we will consider reduction rules on sequences of traces. For better readability we separate the consecutive traces in such a sequence by commas. Let $u_1, u_2, \ldots, u_n \in \mathrm{IRR}(A^{\pm 1}, I)$ be irreducible traces. The sequence $u_1, u_2, \ldots, u_n$ is *I-freely reducible* if the sequence $u_1, u_2, \ldots, u_n$ can be reduced to the empty sequence ε by the following rules:

- $u_i, u_j \to u_j, u_i$ if $u_i I u_j$,
- $u_i, u_j \to \varepsilon$ if $u_i = u_j^{-1}$ in $\mathbb{G}(A, I)$,
- $u_i \to \varepsilon$ if $u_i = \varepsilon$ (this rule deletes the empty trace ε from a sequence of traces).

A concrete sequence of these rewrite steps leading to the empty sequence is a *reduction* of the sequence $u_1, u_2, \ldots, u_n$. Such a reduction can be seen as a witness for the fact that $u_1 u_2 \cdots u_n = 1$ in $\mathbb{G}(A, I)$. On the other hand, $u_1 u_2 \cdots u_n = 1$ does not necessarily imply that $u_1, u_2, \ldots, u_n$ has a reduction. For instance, the sequence a^{-1}, ab, b^{-1} has no reduction. But we can show that every sequence which multiplies to 1 in G can be refined (by factorizing the elements of the sequence) such that the resulting refined sequence has a reduction. For getting an NP-algorithm, it is important to bound the length of the refined sequence exponentially in the length of the initial sequence.

▶ **Lemma 7.** *Let $n \geq 2$ and $u_1, u_2, \ldots, u_n \in \mathrm{IRR}(A^{\pm 1}, I)$. If $u_1 u_2 \cdots u_n = 1$ in $\mathbb{G}(A, I)$, then there exist factorizations $u_i = u_{i,1} \cdots u_{i,k_i}$ such that the sequence*

$$u_{1,1}, \ldots, u_{1,k_1},\ u_{2,1}, \ldots, u_{2,k_2},\ \ldots, u_{n,1}, \ldots, u_{n,k_n}$$

is I-freely reducible. Moreover, $\sum_{i=1}^{n} k_i \leq 2^n - 2$.

We now come to the main technical result of this paper:

▶ **Theorem 8.** *Let $u_1, u_2, \ldots, u_n \in \mathbb{G}(A, I) \setminus \{1\}$, $v_0, v_1, \ldots, v_n \in \mathbb{G}(A, I)$ and let $x_1, \ldots, x_n$ be variables (we may have $x_i = x_j$ for $i \neq j$) ranging over $\mathbb{N}$. Then, the set of solutions of the exponent equation*

$$v_0 u_1^{x_1} v_1 u_2^{x_2} v_2 \cdots u_n^{x_n} v_n = 1, \qquad (1)$$

is semilinear. Moreover, if there exists a solution, then there is a solution with $x_i \in \mathcal{O}((\alpha n)! \cdot 2^{2\alpha^2 n(n+3)} \cdot \mu^{8\alpha(n+1)} \cdot \nu^{8\alpha|A|(n+1)})$*, where*

- $\alpha \le |A|$ *is the size of a largest clique of the complementary graph* $(A, I)^c = (A, (A \times A) \setminus I)$,
- $\lambda = \max\{|u_1|, |u_2|, \dots, |u_n|, |v_0|, |v_1|, \dots, |v_n|\}$,
- $\mu \in \mathcal{O}(|A|^\alpha \cdot 2^{2\alpha^2 n} \cdot \lambda^\alpha)$*, and*
- $\nu \in \mathcal{O}(\lambda^\alpha)$.

Proof. Let us choose irreducible traces for $u_1, u_2, \dots, u_n, v_0, v_1, \dots, v_n$; we denote these traces with the same letters as the group elements. A trace u is called cyclically reduced if there do not exist $a \in A^{\pm 1}$ and v such that $u = ava^{-1}$. For every trace u there exist unique traces p, w such that $u = pwp^{-1}$ and w is cyclically reduced (since the reduction relation $a^{-1}xa \to x$ is terminating and confluent [10]). These traces p and w can be computed in polynomial time. Note that for a cyclically reduced irreducible trace w, every power w^n is irreducible. Let $u_i = p_i w_i p_i^{-1}$ with w_i cyclically reduced. By replacing every $u_i^{x_i}$ by $p_i w_i^{x_i} p_i^{-1}$, we can assume that all u_i are cyclically reduced and irreducible. In case one of the traces u_i is not connected, we can write u_i as $u_i = u_{i,1} u_{i,2}$ with $u_{i,1} I u_{i,2}$ and $u_{i,1} \neq 1 \neq u_{i,2}$. Thus, we can replace the power $u_i^{x_i}$ by $u_{i,1}^{x_i} u_{i,2}^{x_i}$. Note that $u_{i,1}$ and $u_{i,2}$ are still irreducible and cyclically reduced. By doing this, the number n from the theorem multiplies by at most α (which is the maximal number of pairwise independent letters). In order to keep the notation simple we still use the letter n for the number of u_i, but at the end of the proof we have to multiply n by α in the derived bound. Hence, for the further proof we can assume that all u_i are connected, irreducible and cyclically reduced. Let λ be the maximal length of one of the traces $u_1, u_2, \dots, u_n, v_0, v_1, \dots, v_n$, which does not increase by the above preprocessing.

We now apply Lemma 7 to the equation (1), where every $u_i^{x_i}$ is viewed as a single factor. Note that by our preprocessing, all factors $u_1^{x_1}, u_2^{x_2}, \dots, u_n^{x_n}, v_0, \dots, v_n$ are irreducible (for all choices of the x_i). By taking the disjunction over (i) all possible factorizations of the $2n+1$ factors $u_1^{x_1}, u_2^{x_2}, \dots, u_n^{x_n}, v_0, \dots, v_n$ into totally at most $2^{2n+1} - 2$ factors and (ii) all possible reductions of the resulting refined factorization of $v_0 u_1^{x_1} v_1 u_2^{x_2} v_2 \cdots u_n^{x_n} v_n$, it follows that (1) is equivalent to a disjunction of statements of the following form: There exist traces $y_{i,1}, \dots, y_{i,k_i}$ $(1 \le i \le n)$ and $z_{i,1}, \dots, z_{i,l_i}$ $(0 \le i \le n)$ such that

(a) $u_i^{x_i} = y_{i,1} \cdots y_{i,k_i}$ $(1 \le i \le n)$
(b) $v_i = z_{i,1} \cdots z_{i,l_i}$ $(0 \le i \le n)$
(c) $y_{i,j} I y_{k,l}$ for all $(i,j,k,l) \in J_1$
(d) $y_{i,j} I z_{k,l}$ for all $(i,j,k,l) \in J_2$
(e) $z_{i,j} I z_{k,l}$ for all $(i,j,k,l) \in J_3$
(f) $y_{i,j} = y_{k,l}^{-1}$ for all $(i,j,k,l) \in M_1$
(g) $y_{i,j} = z_{k,l}^{-1}$ for all $(i,j,k,l) \in M_2$
(h) $z_{i,j} = z_{k,l}^{-1}$ for all $(i,j,k,l) \in M_3$

Here, the numbers k_i and l_i sum up to at most $2^{2n+1} - 2$ (hence, some k_i can be exponentially large, whereas l_i can be bound by the length of v_i, which is at most λ). The tuple sets J_1, J_2, J_3 collect all independences between the factors $y_{i,j}$, $z_{k,l}$ that are necessary to carry out the chosen reduction of the refined left-hand side in (1). Similarly, the tuple sets M_1, M_2, M_3 tell us which of the factors $y_{i,j}$, $z_{k,l}$ cancels against which of the factors $y_{i,j}$, $z_{k,l}$ in our chosen reduction of the refined left-hand side in (1). Note that every factor $y_{i,j}$ (resp., $z_{k,l}$) appears in exactly one of the identites (f), (g), (h) (since in the reduction every factor cancels against another unique factor).

Next, we simplify our statements. Since the v_i are concrete traces (of length at most λ), we can take a disjunction over all possible factorizations $v_i = v_{i,1} \cdots v_{i,l_i}$ $(1 \le i \le n+1)$. This allows to replace every variable $z_{i,j}$ by a concrete trace $v_{i,j}$. Statements of the form $v_{i,j} I v_{k,l}$ and $v_{i,j} = v_{k,l}^{-1}$ can, of course, be eliminated. Moreover, if there is an identity $y_{i,j} = v_{k,l}^{-1}$ then we can replace the variable $y_{i,j}$ by the concrete trace $v_{k,l}^{-1}$ (of length at most λ).

In the next step, we replace statements of the form $u_i^{x_i} = y_{i,1} \cdots y_{i,k_i}$ $(1 \le i \le n)$. Note that some of the variables $y_{i,j}$ might have been replaced by concrete traces of length at most λ. We apply to each of these equations Lemma 3, or better Remark 4. This allows us to replace every equation $u_i^{x_i} = y_{i,1} \cdots y_{i,k_i}$ $(1 \le i \le n)$ by a disjunction of statements of the following form: There exist numbers $x_{i,j} > 0$ $(1 \le i \le n,\ j \in K_i)$ such that

- $x_i = c_i + \sum_{j \in K_i} x_{i,j}$ for all $1 \le i \le n$,
- $y_{i,j} = p_{i,j} u_i^{x_{i,j}} s_{i,j}$ for all $1 \le i \le n$, $j \in K_i$,
- $y_{i,j} = p_{i,j} s_{i,j}$ for all $1 \le i \le n$, $j \in [1, k_i] \setminus K_i$.

Here, $K_i \subseteq [1, k_i]$, the c_i are concrete numbers with $c_i \le |A| \cdot (k_i - 1)$, and the $p_{i,j}, s_{i,j}$ are concrete traces of length at most $|A| \cdot (k_i - 1) \cdot |u_i| \le |A| \cdot (2^{2n+1} - 3) \cdot \lambda$. Hence, the lengths of these traces can be exponential in n.

Note that since $x_i > 0$, we know the alphabet of $y_{i,j} = p_{i,j} u_i^{x_{i,j}} s_{i,j}$ (resp., $y_{i,j} = p_{i,j} s_{i,j}$). This allows us to replace all independences of the form $y_{i,j} I y_{k,l}$ for $(i,j,k,l) \in J_1$ (see (c)) and $y_{i,j} I z_{k,l}$ for $(i,j,k,l) \in J_2$ (see (d)) by concrete truth values. Note that all variables $z_{k,l}$ have already been replaced by concrete traces. If $y_{i,j}$ was already replaced by a concrete trace, then we can determine from an equation $y_{i,j} = p_{i,j} u_i^{x_{i,j}} s_{i,j}$ the exponent $x_{i,j}$. Since $y_{i,j}$ was replaced by a trace of length at most λ (a small number), we get $x_{i,j} \le \lambda$, and we can replace $x_{i,j}$ in $x_i = \sum_{j \in K_i} x_{i,j} + c_i$ by a concrete number of size at most λ. Finally, if $y_{i,j}$ was replaced by a concrete trace, and we have an equation of the form $y_{i,j} = p_{i,j} s_{i,j}$, then the resulting identity is either true or false and can be eliminated.

After this step, we obtain a disjunction of statements of the following form: There exist numbers $x_{i,j} > 0$ $(1 \le i \le n,\ j \in K'_i)$ such that

(a') $x_i = c_i + \sum_{j \in K'_i} x_{i,j}$ for all $1 \le i \le n$, and
(b') $p_{i,j} u_i^{x_{i,j}} s_{i,j} = s_{k,l}^{-1} (u_k^{-1})^{x_{k,l}} p_{k,l}^{-1}$ for all $(i,j,k,l) \in M$.

Here, $K'_i \subseteq K_i$ is a set of size at most $k_i \le 2^{2n+1} - 2$, $c_i \le |A| \cdot (k_i - 1) + \lambda \cdot k_i < (|A| + \lambda) \cdot (2^{2n+1} - 2)$, and the $p_{i,j}, s_{i,j}$ are concrete traces of length at most $|A| \cdot (2^{2n+1} - 3) \cdot \lambda$. The set M specifies a matching in the sense that for every exponent $x_{a,b}$ $(1 \le a \le n,\ b \in K'_i)$ there is a unique $(i,j,k,l) \in M$ such that $(i,j) = (a,b)$ or $(k,l) = (a,b)$. Note that

$$|M| = \frac{1}{2} \sum_{i=1}^{n} |K'_i| \le \frac{1}{2} \sum_{i=1}^{n} k_i \le \frac{1}{2} (2^{2n+1} - 2) = 2^{2n} - 1.$$

We now apply Lemma 5 to the identities $p_{i,j} u_i^{x_{i,j}} s_{i,j} = s_{k,l}^{-1} (u_k^{-1})^{x_{k,l}} p_{k,l}^{-1}$. Each such identity can be replaced by a disjunction of constraints

$$(x_{i,j}, x_{k,l}) \in \{(a_{i,j,k,l} + b_{i,j,k,l} \cdot z_{i,j,k,l},\ c_{i,j,k,l} + d_{i,j,k,l} \cdot z_{i,j,k,l}) \mid z_{i,j,k,l} \in \mathbb{N}\}.$$

For the numbers $a_{i,j,k,l}, b_{i,j,k,l}, c_{i,j,k,l}, d_{i,j,k,l}$ we obtain the bound

$$a_{i,j,k,l}, b_{i,j,k,l}, c_{i,j,k,l}, d_{i,j,k,l} \in \mathcal{O}(\mu^8 \cdot \nu^{8|A|})$$

(the alphabet of the traces is $A^{\pm 1}$ which has size $2|A|$, therefore, we have to multiply in Lemma 5 $|A|$ by 2), where, by Lemma 2,

$$\mu = \max\{\rho(p_{i,j}), \rho(p_{k,l}), \rho(s_{i,j}), \rho(s_{k,l})\} \in \mathcal{O}(|A|^\alpha \cdot 2^{2\alpha n} \cdot \lambda^\alpha) \text{ and} \tag{2}$$

$$\nu = \max\{\rho(u_i), \rho(u_k)\} \in \mathcal{O}(\lambda^\alpha). \tag{3}$$

Note that $\rho(t) = \rho(t^{-1})$ for every trace t. The above condition (a') for x_i can be written as

$$x_i = c_i + \sum_{(i,j,k,l) \in M} (a_{i,j,k,l} + b_{i,j,k,l} \cdot z_{i,j,k,l}) + \sum_{(k,l,i,j) \in M} (c_{k,l,i,j} + d_{k,l,i,j} \cdot z_{k,l,i,j}).$$

Note that the two sums in this equation contain in total $|K'_i| \leq 2^{2n+1}$ many summands (since for every $j \in K'_i$ there is a unique pair (k,l) with $(i,j,k,l) \in M$ or $(k,l,i,j) \in M$).

Hence, after a renaming of symbols, the initial equation (1) becomes equivalent to a finite disjunction of statements of the form: There exist $z_1, \dots, z_m \in \mathbb{N}$ (these z_i are the above $z_{i,j,k,l}$ and $m = |M|$) such that

$$x_i = a_i + \sum_{j=1}^{m} a_{i,j} z_j \text{ for all } 1 \leq i \leq n. \tag{4}$$

Moreover, we have the following size bounds:

- $m = |M| \leq 2^{2n} - 1$,
- $a_i \in \mathcal{O}(c_i + |K'_i| \cdot \mu^8 \cdot \nu^{8|A|}) \subseteq \mathcal{O}(2^{2n}(|A| + \lambda + \mu^8 \cdot \nu^{8|A|})) \subseteq \mathcal{O}(2^{2n} \cdot \mu^8 \cdot \nu^{8|A|})$
- $a_{i,j} \in \mathcal{O}(\mu^8 \cdot \nu^{8|A|})$

Recall that some of the variables x_i can be identical. W.l.o.g. assume that $x_1, \dots, x_k$ are pairwise different and for all $k+1 \leq i \leq n$, $x_i = x_{f(i)}$, where $f : [k+1, n] \to [1, k]$. Then, the system of equations (4) is equivalent to

$$x_i = a_i + \sum_{j=1}^{m} a_{i,j} z_j \ (1 \leq i \leq k) \quad \text{and} \quad a_i - a_{f(i)} = \sum_{j=1}^{m} (a_{f(i),j} - a_{i,j}) z_j \ (k+1 \leq i \leq n).$$

The set of all $(x_1, \dots, x_k) \in \mathbb{N}^k$ for which there exist $z_1, \dots, z_m \in \mathbb{N}$ satisfying these equalities is semilinear by Lemma 6, and if it is non-empty then it contains $(x_1, \dots, x_k) \in \mathbb{N}^k$ such that $x_i \in \mathcal{O}(n! \cdot m^2 \cdot 2^{2n(n+1)} \cdot \mu^{8(n+1)} \cdot \nu^{8|A|(n+1)}) \subseteq \mathcal{O}(n! \cdot 2^{2n(n+3)} \cdot \mu^{8(n+1)} \cdot \nu^{8|A|(n+1)})$. Recall that in this bound we have to replace n by $\alpha \cdot n$ due to the initial preprocessing. This proves the theorem. ◀

▶ **Theorem 9.** *Let (A, I) be a fixed independence alphabet. Solvability of compressed exponent equations over the graph group $\mathbb{G}(A, I)$ is in* NP.

Proof. Consider a compressed exponent equation $E = (v_0 u_1^{x_1} v_1 u_2^{x_2} v_2 \cdots u_n^{x_n} v_n = 1)$, where $u_i = \mathrm{val}(\mathcal{G}_i)$ and $v_i = \mathrm{val}(\mathcal{H}_i)$ for SLPs $\mathcal{G}_1, \dots, \mathcal{G}_n, \mathcal{H}_0, \dots, \mathcal{H}_n$, which form the input. Let $m = \max\{|\mathcal{G}_1|, \dots, |\mathcal{G}_n|, |\mathcal{H}_0|, \dots, |\mathcal{H}_n|\}$, $\lambda = \max\{|u_1|, |u_2|, \dots, |u_n|, |v_0|, |v_1|, \dots, |v_n|\} \in 2^{\mathcal{O}(m)}$. By Thm. 8 we know that if there exists a solution for E then there exists a solution σ with $\sigma(x_i) \in \mathcal{O}((\alpha n)! \cdot 2^{2\alpha^2 n(n+3)} \cdot \mu^{8\alpha(n+1)} \cdot \nu^{8\alpha|A|(n+1)})$, where $\mu \in \mathcal{O}(|A|^\alpha \cdot 2^{2\alpha^2 n} \cdot \lambda^\alpha)$, $\nu \in \mathcal{O}(\lambda^\alpha)$, and $\alpha \leq |A|$. Note that the bound on the $\sigma(x_i)$ is exponential in the input length (the sum of the sizes of all $\mathcal{G}_i$ and $\mathcal{H}_i$). Hence, we can guess in polynomial time the binary encodings of numbers $k_i \in \mathcal{O}((\alpha n)! \cdot 2^{2\alpha^2 n(n+3)} \cdot \mu^{8\alpha(n+1)} \cdot \nu^{8\alpha|A|(n+1)})$ (where $k_i = k_j$ if $x_i = x_j$). It remains to verify the identity $v_0 u_1^{k_1} v_1 u_2^{k_2} v_2 \cdots u_n^{k_n} v_n = 1$ in $\mathbb{G}(A, I)$, where all u_i and v_i are given by SLPs. This is an instance of the so called *compressed word problem* for $\mathbb{G}(A, I)$, where the input consists of an SLP $\mathcal{G}$ over the alphabet $A^{\pm 1}$ and it is asked whether $\mathrm{val}(\mathcal{G}) = 1$ in $\mathbb{G}(A, I)$. Note that the big powers $\mathrm{val}(\mathcal{G}_i)^{k_i}$ can be produced with the productions of $\mathcal{G}_i$ and additional $\lceil \log k_i \rceil$ many productions (using iterated squaring). Since the compressed word problem for a graph group can be solved in polynomial time [29] (NP would suffice), the theorem follows. For the last step, it is important that (A, I) is fixed. ◀

▶ Remark 10. Note that the bound on the exponents $\sigma(x_i)$ in the previous proof is still exponential in the input length if the independence alphabet (A, I) is part of the input as well. The problem is that we do not know whether the *uniform compressed word problem* for graph groups (where the input is an independence alphabet (A, I) together with an SLP over the terminal alphabet $A^{\pm 1}$) can be solved in polynomial time or at least in NP. The latter would suffice to get an NP-algorithm for solvability of compressed exponent equations over a graph group that is part of the input.

7 Transfer Results

Here, we show that the property of having an NP-algorithm for the knapsack problem (or compressed exponent equations) is preserved by certain group constructions.

Finite extensions and virtually special groups. Our first transfer result concerns finite extensions. Together with our result on graph groups, this will provide an abundant class of groups with an NP-algorithm for compressed exponent equations. A group G is called *virtually special* if it is a finite extension of a subgroup of a graph group. Recently, this class of groups turned out to be very rich. It includes all Coxeter groups [18], one-relator groups with torsion [41], fully residually free groups [41], and fundamental groups of hyperbolic 3-manifolds [1].

The following is our transfer theorem for finite extensions. The idea is to guess the cosets that occur on the left-hand side of an exponent equation. Given a collection of cosets, one can then reduce to exponent equations over G.

▶ **Theorem 11.** *Let G and H be finitely generated groups such that H is a finite extension of G. If knapsack (resp. solvability of compressed exponent equations) belongs to* NP *for G, then the same holds for H.*

From Theorem 9 it follows that solvability of compressed exponent equations belongs to NP for every subgroup of a graph group. Therefore, our transfer theorem implies:

▶ **Theorem 12.** *Solvability of compressed exponent equations belongs to* NP *for every virtually special group. In particular, solvability of compressed exponent equations belongs to* NP *for Coxeter groups, one-relator groups with torsion, fully residually free groups, and fundamental groups of hyperbolic 3-manifolds.*

HNN-extensions and amalgamated products. The remaining transfer results concern two constructions that are of fundamental importance in combinatorial group theory [34], namely HNN-extensions with finite associated subgroups and amalgamated products with finite identified subgroups. These constructions are known to preserve a variety of important structural and algorithmic properties [2, 6, 19, 22, 23, 25, 26, 30, 31, 35].

Suppose G is a finitely generated group that has two isomorphic subgroups A and B with an isomorphism $\varphi\colon A \to B$. Then the corresponding *HNN-extension* is the group $H = \langle G, t \mid t^{-1}at = \varphi(a)\ (a \in A)\rangle$, where t is a new letter not contained in G. Intuitively, H is obtained from G by adding a new element t such that conjugating elements of A with t applies the isomorphism φ. Here, t is called the *stable letter* and the groups A and B are the *associated subgroups*. A basic fact about HNN-extensions is that the group G embeds naturally into H [20].

Our algorithm for knapsack in HNN-extensions is an adaptation of the saturation algorithm of Benois [3] for the membership problem for rational subsets of free groups. Here, for each path spelling aa^{-1}, one adds a parallel edge labeled with the empty word. Since knapsack is a special case of this problem, we need to choose a suitable subclass of automata that is preserved by our saturation and corresponds to the knapsack problem. This subclass is the class of *knapsack automata*, where each strongly connected component is a singleton or an induced (directed) cycle.

With respect to NP membership, one can show that the knapsack problem is equivalent to the membership problem for knapsack automata (see [33]). Therefore, it suffices to show that NP membership of the latter problem is preserved by HNN-extensions, which we prove using a saturation procedure. Here, the basic idea is to successively guess states p and q and check (using the algorithm for G) whether there is a path from p to q reading a word $u = t^{-1}wt$ (resp. $u = twt^{-1}$) such that w represents an element $a \in A$ (resp. $b \in B$). If this is the case, u represents $\varphi(a)$ (resp. $\varphi^{-1}(b)$) and we add an edge $p \xrightarrow{\varphi(a)} q$ (resp. $p \xrightarrow{\varphi^{-1}(b)} q$), which is termed *shortcut edge*.

This, however, it is not possible if p and q lie on a cycle, as that would create connected components that are not cycles. In this case, we replace the cycle segment between p and q with the shortcut edge and glue in new paths to make up for lost edges that entered or left the cycle between p and q. In the end, we show that every element accepted by the automaton has an accepting run that avoids factors $t^{-1}wt$ and twt^{-1} as above. This allows us then to apply the algorithm for G to decide the membership problem.

▶ **Theorem 13.** *Let H be an HNN-extension of the finitely generated group G with finite associated subgroups. If knapsack for G belongs to* NP*, then the same holds for H.*

In our last transfer theorem, we consider amalgamated free products. For $i \in \{0, 1\}$, let $G_i = \langle \Sigma_i \mid R_i \rangle$ be a finitely generated group and let F be a finite group that is embedded in each G_i via injective morphisms $\varphi_i \colon F \to G_i$ for $i \in \{0, 1\}$. Then, the *free product with amalgamation with identified subgroup F* is defined as $G_0 *_F G_1 = \langle G_0 * G_1 \mid \varphi_0(f) = \varphi_1(f) \ (f \in F)\rangle$. Here, $G_0 * G_1$ denotes the free product $G_0 * G_1 = \langle \Sigma_0 \uplus \Sigma_1 \mid R_0 \uplus R_1 \rangle$. Intuitively, $G_0 *_F G_1$ consists of alternating sequences of elements of G_0 and G_1 where the elements of $\varphi_0(F)$ and $\varphi_1(F)$ are identified.

The transfer theorem states that taking amalgamated products with finite identified subgroups preserves NP membership of knapsack. Since $G_0 *_F G_1$ can be embedded into the HNN-extension $\langle G_0 * G_1, t \mid t^{-1}\varphi_0(f)t = \varphi_1(f) \ (f \in F)\rangle$, it suffices to prove the theorem for a free product $G_0 * G_1$. As above, we work with the membership problem for knapsack automata and use a saturation procedure. Note that $G_0 * G_1$ is generated by $\Sigma_0 \cup \Sigma_1$. We guess states p and q and $i \in \{0, 1\}$ and then check, using the NP algorithm for G_i, whether there is a path from p to q that reads a representative of $1 \in G_i$ in $(\Sigma_i^{\pm 1})^*$. If so, we add a shortcut edge (p, ε, q). Again, the case that p and q lie on a cycle is somewhat more involved.

▶ **Theorem 14.** *Let G_0 and G_1 be finitely generated groups with a common finite subgroup F. If knapsack for G_0 and for G_1 belongs to* NP*, then the same holds for $G_0 *_F G_1$.*

8 Hardness Results

Since knapsack for binary encoded integers is NP-complete, it follows that the compressed knapsack problem is NP-hard for every group that contains an element of infinite order. Our final result states that (uncompressed) knapsack and subset sum are NP-complete for a direct product of two free groups of rank two. Since this group is a graph group, we obtain a matching lower bound for Thm. 9. Moreover, we solve an open problem from [15].

▶ **Theorem 15.** *The subset sum problem and the knapsack problem are* NP*-complete for $F_2 \times F_2$, where F_2 is the free group of rank two. For knapsack* NP*-hardness already holds for the variant, where the exponent variables are allowed to take values from $\mathbb{Z}$ (see Remark 1).*

References

1 I. Agol. The virtual Haken conjecture. Technical report, arXiv.org, 2012. `http://arxiv.org/abs/1204.2810`.

2 R. B. J. T. Allenby and R. J. Gregorac. On locally extended residually finite groups. In *Conference on Group Theory (Univ. Wisconsin-Parkside, Kenosha, Wis., 1972)*, number 319 in Lecture Notes in Mathematics, pages 9–17. Springer, 1973.

3 M. Benois. Parties rationnelles du groupe libre. *C. R. Acad. Sci. Paris*, Sér. A, 269:1188–1190, 1969.

4 A. Bertoni, G. Mauri, and N. Sabadini. Membership problems for regular and context free trace languages. *Information and Computation*, 82:135–150, 1989.

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

6 V. N. Bezverkhniĭ. On the intersection of subgroups in HNN-groups. *Fundamentalnaya i Prikladnaya Matematika*, 4(1):199–222, 1998.

7 M. Charikar, E. Lehman, A. Lehman, D. Liu, R. Panigrahy, M. Prabhakaran, A. Sahai, and A. Shelat. The smallest grammar problem. *IEEE Transactions on Information Theory*, 51(7):2554–2576, 2005.

8 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.

9 W. Dicks and M. J. Dunwoody. *Groups Acting on Graphs*. Cambridge University Press, 1989.

10 V. Diekert and G.Rozenberg, editors. *The Book of Traces*. World Scientific, 1995.

11 V. Diekert and J. Kausch. Logspace computations in graph products. *Journal of Symbolic Computation*, 2015. to appear.

12 V. Diekert and M. Lohrey. Word equations over graph products. *International Journal of Algebra and Computation*, 18(3):493–533, 2008.

13 V. Diekert and A. Muscholl. Solvability of equations in graph groups is decidable. *International Journal of Algebra and Computation*, 16(6):1047–1069, 2006.

14 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.

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

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

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

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

19 N. Haubold and M. Lohrey. Compressed word problems in HNN-extensions and amalgamated products. *Theory of Computing Systems*, 49(2):283–305, 2011.

20 G. Higman, B. H. Neumann, and H. Neumann. Embedding theorems for groups. *Journal of the London Mathematical Society. Second Series*, 24:247–254, 1949.

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

22 M. Kambites, P. V. Silva, and B. Steinberg. On the rational subset problem for groups. *Journal of Algebra*, 309(2):622–639, 2007.

23 I. Kapovich, R. Weidmann, and A. Myasnikov. Foldings, graphs of groups and the membership problem. *International Journal of Algebra and Computation*, 15(1):95–128, 2005.

24 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.

25 A. Karrass and D. Solitar. The subgroups of a free product of two groups with an amalgamated subgroup. *Transactions of the American Mathematical Society*, 150:227–255, 1970.

26 A. Karrass and D. Solitar. Subgroups of HNN groups and groups with one defining relation. *Canadian Journal of Mathematics*, 23:627–643, 1971.

27 D. König, M. Lohrey, and G. Zetzsche. Knapsack and subset sum problems in nilpotent, polycyclic, and co-context-free groups. Technical report, arXiv.org, 2015. `http://arxiv.org/abs/1507.05145`.

28 D. Kuske and M. Lohrey. Logical aspects of Cayley-graphs: the monoid case. *International Journal of Algebra and Computation*, 16(2):307–340, 2006.

29 M. Lohrey. *The Compressed Word Problem for Groups.* SpringerBriefs in Mathematics. Springer, 2014.

30 M. Lohrey and G. Sénizergues. Theories of HNN-extensions and amalgamated products. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006)*, number 4052 in Lecture Notes in Computer Science, pages 504–515. Springer, 2006.

31 M. Lohrey and G. Sénizergues. Rational subsets in HNN-extensions and amalgamated products. *International Journal of Algebra and Computation*, 18(1):111–163, 2008.

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

33 M. Lohrey and G. Zetzsche. Knapsack in graph groups, HNN-extensions and amalgamated products. Technical report, arXiv.org, 2015. `http://arxiv.org/abs/1509.05957`.

34 R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory.* Springer, 1977.

35 V. Metaftsis and E. Raptis. Subgroup separability of graphs of abelian groups. *Proceedings of the American Mathematical Society*, 132:1873–1884, 2004.

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

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

38 J. R. Stallings. *Group Theory and Three-Dimensional Manifolds.* Number 4 in Yale Mathematical Monographs. Yale University Press, 1971.

39 A. W. To. Unary finite automata vs. arithmetic progressions. *Information Processing Letters*, 109(17):1010–1014, 2009.

40 J. von zur Gathen and M. Sieveking. A bound on solutions of linear integer equalities and inequalities. *Proceedings of the American Mathematical Society*, 72(1):155–158, 1978.

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