Compressed word problems in HNN-extensions and amalgamated products

Niko Haubold and Markus Lohrey

Institut für Informatik, Universität Leipzig
{haubold,lohrey}@informatik.uni-leipzig.de

Abstract. It is shown that the compressed word problem for an HNN-extension $\langle H, t \mid t^{-1}at = \varphi(a)(a \in A)\rangle$ with A finite is polynomial time Turing-reducible to the compressed word problem for the base group H. An analogous result for amalgamated free products is shown as well.

1 Introduction

Since it was introduced by Dehn in 1910, the *word problem* for groups has emerged to a fundamental computational problem linking group theory, topology, mathematical logic, and computer science. The word problem for a finitely generated group G asks, whether a given word over the generators of G represents the identity of G, see Section 2 for more details. Dehn proved the decidability of the word problem for surface groups. On the other hand, 50 years after the appearance of Dehn's work, Novikov and independently Boone proved the existence of a finitely presented group with undecidable word problem, see [10] for references. However, many natural classes of groups with decidable word problem are known, as for instance finitely generated linear groups, automatic groups and one-relator groups. With the rise of computational complexity theory, also the complexity of the word problem became an active research area. This development has gained further attention by potential applications of combinatorial group theory for secure cryptographic systems [11].

In order to prove upper bounds on the complexity of the word problem for a group G, a "compressed" variant of the word problem for G was introduced in [6, 7, 14]. In the *compressed word problem* for G, the input word over the generators is not given explicitly but succinctly via a *straight-line program* (SLP for short). This is a context free grammar that generates exactly one word, see Section 2. Since the length of this word may grow exponentially with the size (number of productions) of the SLP, SLPs can be seen indeed as a succinct string representation. SLPs turned out to be a very flexible compressed representation of strings, which are well suited for studying algorithms for compressed data. In [7, 14] it was shown that the word problem for the automorphism group $\mathrm{Aut}(G)$ of G can be reduced in polynomial time to the *compressed* word problem for G. In [6], it was shown that the compressed word problem for a finitely generated free group F can be solved in polynomial time. Hence, the word problem for

$\mathrm{Aut}(F)$ turned out to be solvable in polynomial time [14], which solved an open problem from [5]. Generalizations of this result can be found in [7].

In this paper, we prove a transfer theorem for the compressed word problem of *HNN-extensions* [2]. For a *base group* H, two isomorphic subgroups $A, B \leq H$, and an isomorphism $\varphi : A \to B$, the corresponding HNN-extension is the group

$$G = \langle H, t \mid t^{-1}at = \varphi(a) \; (a \in A) \rangle. \tag{1}$$

Intuitively, it is obtained by adding to H a new generator t (the *stable letter*) in such a way that conjugation of A by t realizes φ. The subgroups A and B are also called the *associated subgroups*. A related operation is that of the *amalgamated free product* of two groups H_1 and H_2 with isomorphic subgroups $A_1 \leq H_1$, $A_2 \leq H_2$ and an isomorphism $\varphi : A_1 \to A_2$. The corresponding amalgamated free product is the group $\langle H_1 * H_2 \mid a = \varphi(a) \; (a \in A_1) \rangle$. Intuitively, it results from the free product $H_1 * H_2$ by identifying every element $a \in A_1$ with $\varphi(a) \in A_2$. The subgroups A_1 and A_2 are also called the *amalgamated subgroups*.

HNN-extensions were introduced by Higman, Neumann, and Neumann in 1949 [2]. They proved that H embeds into the group G from (1). Modern proofs of the above mentioned Novikov-Boone theorem use HNN-extensions as the main tool for constructing finitely presented groups with an undecidable word problem [10]. In particular, arbitrary HNN-extensions do not preserve good algorithmic properties of groups like decidability of the word problem. In this paper, we restrict to HNN-extensions (resp. amalgamated products) with finite associated (resp. amalgamated) subgroups, which is an important subcase. Stallings proved [15] that a group has more than one end if and only if it is either an HNN-extension with finite associated subgoups or an amalgamated free product with finite amalgamated subgroups. Moreover, a group is virtually-free (i.e., has a free subgroup of finite index) if and only if it can be built up from finite groups using amalgamated products with finite amalgamated subgoups and HNN-extensions with finite associated subgroups [1].

It is not hard to see that the word problem for an HNN-extension (1) with A finite can be reduced in polynomial time to the word problem of the base group H. The main result of this paper extends this transfer theorem to the compressed setting: the compressed word problem for (1) with A finite can be reduced in polynomial time to the compressed word problem for H. In fact, we prove a slightly more general result, which deals with HNN-extensions with several stable letters $t_1, \ldots, t_n$, where the number n is part of the input. For each stable letter t_i the input contains a *partial* isomorphism φ_i from the fixed finite subgroup $A \leq H$ to the fixed finite subgroup $B \leq H$ and we consider the multiple HNN-extension $G = \langle H, t_1, \ldots, t_n \mid t_i^{-1}at_i = \varphi_i(a) \; (1 \leq i \leq n, a \in \mathrm{dom}(\varphi_i)) \rangle$. Our polynomial time reduction consists of a sequence of polynomial time reductions. In a first step, we reduce the compressed word problem for G to the same problem for *reduced sequences*. These are strings (over the generators of H and the symbols $t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}$) that do not contain a substring of the form $t_i^{-1}wt_i$ (resp. $t_iwt_i^{-1}$), where the string w represents a group element from the domain (resp. range) of φ_i. In a second step, we reduce the number n of stable letters to a constant δ, which only depends on the size of the fixed subgroup A. The main step of the paper reduces the compressed word problem for reduced sequences over an HNN-extension with $k \leq \delta$ many stable letters (and associated partial isomorphisms from A to B) into two simpler problems: (i) the same problem but with only $k-1$ many stable letters and (ii) the same problem (with at most δ many stable letters) but with associated subgroups that are strictly smaller than A. By iterating this procedure, we arrive after a constant number of iterations (where each iteration is a polynomial time reduction) at a compressed word problem for which we directly know the existence of a polynomial time reduction to the compressed word problem for the base group H. Since the composition of a constant number of polynomial time reductions is again a polynomial time reduction, our main result follows.

The main reduction step in our algorithm uses techniques similar to those from [8], where a transfer theorem for solving equations over HNN-extensions with finite associated subgroups was shown.

From the close relationship of HNN-extensions with amalgamated free products, a polynomial time reduction from the compressed problem for an amalgamated free product $\langle H_1 * H_2 \mid a = \varphi(a)\,(a \in A_1)\rangle$ (with A_1 finite) to the compressed word problems of H_1 and H_2 is deduced in the final Section 4.

A full version of this paper can be found at [3].

2 Preliminaries

Groups and the word problem For background in combinatorial group theory see [10]. For a group G and two elements $x, y \in G$ we denote with $x^y = y^{-1}xy$ the conjugation of x by y. Let G be a *finitely generated group* and let Σ be a finite *group generating set* for G. Hence, $\Sigma^{\pm 1} = \Sigma \cup \Sigma^{-1}$ is a finite *monoid generating set* for G and there exists a canonical monoid homomorphism $h : (\Sigma^{\pm 1})^* \to G$, which maps a word $w \in (\Sigma^{\pm 1})^*$ to the group element represented by w. For $u, v \in (\Sigma^{\pm 1})^*$ we will also say that $u = v$ in G in case $h(u) = h(v)$. The *word problem for G w.r.t. Σ* is the following decision problem:

INPUT: A word $w \in (\Sigma^{\pm 1})^*$.
QUESTION: $w = 1$ in G?

It is well known that if Γ is another finite generating set for G, then the word problem for G w.r.t. Σ is logspace many-one reducible to the word problem for G w.r.t. Γ. This justifies one to speak just of the word problem for the group G.

The *free group* $F(\Sigma)$ generated by Σ can be defined as the quotient monoid $F(\Sigma) = (\Sigma^{\pm 1})^*/\{aa^{-1} = \varepsilon \mid a \in \Sigma^{\pm 1}\}$, where ε denotes the empty word. A *group presentation* is a pair (Σ, R), where Σ is an alphabet of symbols and R is a set of *relations* of the form $u = v$, where $u, v \in (\Sigma^{\pm 1})^*$. The group defined by this presentation is denoted by $\langle \Sigma \mid R\rangle$. It is defined as the quotient $F(\Sigma)/N(R)$, where $N(R)$ is the smallest normal subgroup of the free group $F(\Sigma)$ that contains all elements uv^{-1} with $(u = v) \in R$. In particular $F(\Sigma) = \langle \Sigma \mid \emptyset\rangle$. Of course, one can assume that all relations are of the form $r = 1$. In fact, usually the set of relations is given by a set of *relators* $R \subseteq (\Sigma^{\pm 1})^*$, which corresponds to the set $\{r = 1 \mid r \in R\}$ of relations.

The *free product* of two groups G_1 and G_2 is denoted by $G_1 * G_2$. If $G_i \simeq \langle \Sigma_i \mid R_i \rangle$ for $i \in \{1, 2\}$ with $\Sigma_1 \cap \Sigma_2 = \emptyset$, then $G_1 * G_2 \simeq \langle \Sigma_1 \cup \Sigma_2 \mid R_1 \cup R_2 \rangle$.

Straight-line programs We are using straight-line programs as a compressed representation of strings with reoccuring subpatterns [13]. A *straight-line program (SLP) over the alphabet Γ* is a context free grammar $\mathbb{A} = (V, \Gamma, S, P)$, where V is the set of *nonterminals*, Γ is the set of *terminals*, $S \in V$ is the *initial nonterminal*, and $P \subseteq V \times (V \cup \Gamma)^*$ is the set of *productions* such that (i) for every $X \in V$ there is exactly one $\alpha \in (V \cup \Gamma)^*$ with $(X, \alpha) \in P$ and (ii) there is no cycle in the relation $\{(X, Y) \in V \times V \mid \exists \alpha : (X, \alpha) \in P, Y \text{ occurs in } \alpha\}$. A production (X, α) is also written as $X \to \alpha$. The language generated by the SLP $\mathbb{A}$ contains exactly one word $\mathrm{val}(\mathbb{A})$. Moreover, every nonterminal $X \in V$ generates exactly one word that is denoted by $\mathrm{val}(\mathbb{A}, X)$, or briefly $\mathrm{val}(X)$, if $\mathbb{A}$ is clear from the context. The size of $\mathbb{A}$ is $|\mathbb{A}| = \sum_{(X,\alpha) \in P} |\alpha|$. It can be seen easily that an SLP can be transformed in polynomial time into an SLP in *Chomsky normal form*, which means that all productions have the form $A \to BC$ or $A \to a$ for $A, B, C \in V$ and $a \in \Gamma$.

Let G be a finitely generated group and Σ a finite generating set for G. The *compressed word problem* for G w.r.t. Σ is the following decision problem:

INPUT: An SLP $\mathbb{A}$ over the terminal alphabet $\Sigma^{\pm 1}$.
OUTPUT: Does $\mathrm{val}(\mathbb{A}) = 1$ hold in G?

In this problem, the input size is $|\mathbb{A}|$. As for the ordinary word problem, the complexity of the compressed word problem does not depend on the chosen generating set. This allows one to speak of the compressed word problem for the group G. The compressed word problem for G is also denoted by $\mathrm{CWP}(G)$.

Polynomial time Turing-reductions For two computational problems A and B, we write $A \leq_T^P B$ if A is polynomial time Turing-reducible to B. This means that A can be decided by a deterministic polynomial time Turing-machine that uses B as an oracle. Clearly, $\leq_T^P$ is transitive, and $A \leq_T^P B \in \mathsf{P}$ implies $A \in \mathsf{P}$. More generally, if $A, B_1, \ldots, B_n$ are computational problems, then we write $A \leq_T^P \{B_1, \ldots, B_n\}$ if $A \leq_T^P \bigcup_{i=1}^n (\{i\} \times B_i)$ (the set $\bigcup_{i=1}^n (\{i\} \times B_i)$ is basically the disjoint union of the B_i with every element from B_i marked by i).

HNN-extensions Let $H = \langle \Sigma \mid R \rangle$ be a *base group* with isomorphic subgroups $A_i, B_i \leq H$ $(1 \leq i \leq n)$ and isomorphisms $\varphi_i : A_i \to B_i$. Let $h : (\Sigma^{\pm 1})^* \to H$ be the canonical morphism, which maps a word $w \in (\Sigma^{\pm 1})^*$ to the element of H it represents. We consider the *HNN-extension*

$$G = \langle H, t_1, \ldots, t_n \mid a^{t_i} = \varphi_i(a) \ (1 \leq i \leq n, a \in A_i) \rangle. \tag{2}$$

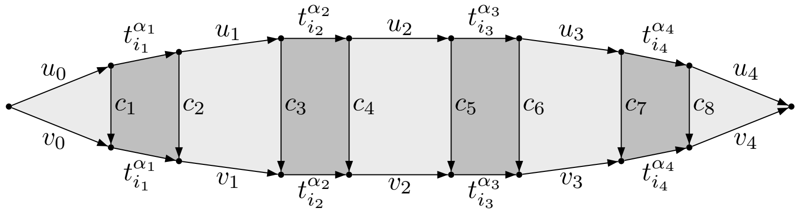
This means that $G = \langle \Sigma \cup \{t_1, \ldots, t_n\} \mid R \cup \{a^{t_i} = \varphi_i(a) \mid 1 \leq i \leq n, a \in A_i\} \rangle$. It is known that the base group H naturally embeds into G [2]. In this paper, we will be only concerned with the case that all groups $A_1, \ldots, A_n$ are finite and that Σ is finite. In this situation, we may assume that $\bigcup_{i=1}^n (A_i \cup B_i) \subseteq \Sigma$. We say that A_i and B_i are *associated subgroups* in the HNN-extension G. For the following, the notations $A_i(+1) = A_i$ and $A_i(-1) = B_i$ are useful. Note that $\varphi_i^\alpha : A_i(\alpha) \to A_i(-\alpha)$ for $\alpha \in \{+1, -1\}$.

A word $u \in (\Sigma^{\pm 1} \cup \{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\})^*$ is *reduced* if u does not contain a factor of the form $t_i^{-\alpha} w t_i^\alpha$ for $\alpha \in \{1, -1\}$, $w \in (\Sigma^{\pm 1})^*$ and $h(w) \in A_i(\alpha)$. With $\mathrm{Red}(H, \varphi_1, \ldots, \varphi_n)$ we denote the set of all reduced words. For a word $u \in (\Sigma^{\pm 1} \cup \{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\})^*$ let us denote with $\pi_t(u)$ the projection of u to the alphabet $\{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\}$. The following Lemma provides a necessary and sufficient condition for equality of reduced strings in the group (2) [9]:

Lemma 2.1. *Let $u = u_0 t_{i_1}^{\alpha_1} u_1 \cdots t_{i_\ell}^{\alpha_\ell} u_\ell$ and $v = v_0 t_{j_1}^{\beta_1} v_1 \cdots t_{j_m}^{\beta_m} v_m$ be reduced words with $u_0, \ldots, u_\ell, v_0, \ldots, v_m \in (\Sigma^{\pm 1})^*$, $\alpha_1, \ldots, \alpha_\ell, \beta_1, \ldots, \beta_m \in \{1, -1\}$, and $i_1, \ldots, i_\ell, j_1, \ldots, j_m \in \{1, \ldots, n\}$. Then $u = v$ in the HNN-extension G from (2) if and only if the following hold:*

(a) $\pi_t(u) = \pi_t(v)$ (i.e., $\ell = m$, $i_k = j_k$, and $\alpha_k = \beta_k$ for $1 \le k \le \ell$)
(b) there exist $c_1, \ldots, c_{2m} \in \bigcup_{k=1}^n (A_k \cup B_k)$ such that:
  - *$u_k c_{2k+1} = c_{2k} v_k$ in H for $0 \le k \le \ell$ (here we set $c_0 = c_{2\ell+1} = 1$)*
  - *$c_{2k-1} \in A_{i_k}(\alpha_k)$ and $c_{2k} = \varphi_{i_k}^{\alpha_k}(c_{2k-1}) \in A_{i_k}(-\alpha_k)$ for $1 \le k \le \ell$.*

Condition (b) of the lemma can be visualized by a diagram of the following form (also called a Van Kampen diagram, see [10] for more details), where $\ell = m = 4$. Light-shaded (resp. dark-shaded) faces represent relations in H (resp. relations of the form $ct_i^\alpha = t_i^\alpha \varphi_i^\alpha(c)$ with $c \in A_i(\alpha)$).



Some simple compressed word problems Plandowski [12] has shown that for two SLPs $\mathbb{A}$ and $\mathbb{B}$ it can be checked in polynomial time whether $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$. In other words: the compressed word problem for a free monoid can be solved in polynomial time. In [6], this result was extended to free groups. A further generalization to free products $G_1 * G_2$ was shown in [7]:

Theorem 2.2. $\mathrm{CWP}(G_1 * G_2) \le_T^P \{\mathrm{CWP}(G_1), \mathrm{CWP}(G_2)\}$.

For our reduction of the compressed word problem of an HNN-extension to the compressed word problem of the base group, we need the special case that in (2) we have $H = A_1 = \cdots = A_n = B_1 = \cdots = B_n$ (in particular, H is finite). In this case, we can even assume that the finite group H (represented by its multiplication table) is part of the input:

Lemma 2.3. *The following problem can be solved in polynomial time:*

INPUT: A finite group H, automorphisms $\varphi_i : H \to H$ $(1 \leq i \leq n)$, and an SLP $\mathbb{A}$ over the alphabet $H \cup \{t_1, t_1^{-1}, \dots t_n, t_n^{-1}\}$.
QUESTION: $\mathrm{val}(\mathbb{A}) = 1$ *in* $\langle H, t_1, \dots, t_n \mid h^{t_i} = \varphi_i(h)\ (1 \leq i \leq n, h \in H)\rangle$*?*

Note that the group $\langle H, t_1, \dots, t_n \mid h^{t_i} = \varphi_i(h)\ (1 \leq i \leq n, h \in H)\rangle$ is the semidirect product $H \rtimes_\varphi F$, where $F = F(t_1, \dots, t_n)$ is the free group generated by $t_1, \dots, t_n$ and the homomorphism $\varphi : F \to \mathrm{Aut}(H)$ is defined by $\varphi(t_i) = \varphi_i$.

3 Compressed word problem of an HNN-extension

In this section we show that the compressed word problem for an HNN-extension of the form (1) is polynomial time Turing-reducible to the compressed word problem for H. In fact, we prove the existence of such a reduction for a slightly more general problem, which we introduce below.

For the further consideration, let us fix the group H together with the finite subgroups A and B. Let Σ be a finite generating set for H. These data are fixed, i.e., they will not belong to the input of computational problems. In the following, when writing down a multiple HNN-extension

$$\langle H, t_1, \dots, t_n \mid a^{t_i} = \varphi_i(a)\ (1 \leq i \leq n, a \in A)\rangle, \tag{3}$$

we allow implicitly that every φ_i is only partially defined on A. Thus, (3) is in fact an abbreviation for $\langle H, t_1, \dots, t_n \mid a^{t_i} = \varphi_i(a)\ (1 \leq i \leq n, a \in \mathrm{dom}(\varphi_i))\rangle$. Note that there is only a fixed number of partial isomorphisms from A to B, but we allow $\varphi_i = \varphi_j$ for $i \neq j$ in (3).

Let us introduce several restrictions and extensions of $\mathrm{CWP}(G)$. Our most general problem is the following computational problem $\mathrm{UCWP}(H, A, B)$ (the letter "U" stands for "uniform", meaning that a list of partial isomorphisms from A to B is part of the input):

INPUT: Partial isomorphisms $\varphi_i : A \to B$ $(1 \leq i \leq n)$ and an SLP $\mathbb{A}$ over the alphabet $\Sigma^{\pm 1} \cup \{t_1, t_1^{-1}, \dots, t_n, t_n^{-1}\}$.
QUESTION: $\mathrm{val}(\mathbb{A}) = 1$ in $\langle H, t_1, \dots, t_n \mid a^{t_i} = \varphi_i(a)\ (1 \leq i \leq n, a \in A)\rangle$?

The restriction of this problem $\mathrm{UCWP}(H, A, B)$ to reduced input strings is denoted by $\mathrm{RUCWP}(H, A, B)$. It is formally defined as the following problem:

INPUT: Partial isomorphisms $\varphi_i : A \to B$ $(1 \leq i \leq n)$ and SLPs $\mathbb{A}, \mathbb{B}$ over the alphabet $\Sigma^{\pm 1} \cup \{t_1, t_1^{-1}, \dots, t_n, t_n^{-1}\}$ such that $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(H, \varphi_1, \dots, \varphi_n)$.
QUESTION: $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ in $\langle H, t_1, \dots, t_n \mid a^{t_i} = \varphi_i(a)\ (1 \leq i \leq n, a \in A)\rangle$?

Let us now consider a fixed list of partial isomorphisms $\varphi_1, \dots, \varphi_n : A \to B$. Then $\mathrm{RCWP}(H, A, B, \varphi_1, \dots, \varphi_n)$ is the following computational problem:

INPUT: Two SLPs $\mathbb{A}$ and $\mathbb{B}$ over the alphabet $\Sigma^{\pm 1} \cup \{t_1, t_1^{-1}, \dots, t_n, t_n^{-1}\}$ such that $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(H, \varphi_1, \dots, \varphi_n)$.
QUESTION: $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ in $\langle H, t_1, \dots, t_n \mid a^{t_i} = \varphi_i(a)\ (1 \leq i \leq n, a \in A)\rangle$?

Our main result is:

Theorem 3.1. $\mathrm{UCWP}(H, A, B) \leq_P^T \mathrm{CWP}(H)$.

The rest of Section 3 sketches the main steps of our proof of Theorem 3.1. First, we state that we may restrict ourselves to SLPs that evaluate to reduced strings:

Lemma 3.2. $\mathrm{UCWP}(H, A, B) \leq_P^T \mathrm{RUCWP}(H, A, B)$. *More precisely, there is a polynomial time Turing-reduction from* $\mathrm{UCWP}(H, A, B)$ *to* $\mathrm{RUCWP}(H, A, B)$ *that on input* $(\varphi_1, \ldots, \varphi_n, \mathbb{A})$ *only asks* $\mathrm{RUCWP}(H, A, B)$*-queries of the form* $(\varphi_1, \ldots, \varphi_n, \mathbb{A}', \mathbb{B}')$ *(thus, the list of partial isomorphisms is not changed).*

Lemma 3.3. *Let* $\varphi_1, \ldots, \varphi_n : A \to B$ *be fixed partial isomorphisms. Then* $\mathrm{CWP}(\langle H, t_1, \ldots, t_n \mid a^{t_i} = \varphi_i(a)\ (1 \leq i \leq n, a \in A)\rangle)$ *is polynomial time Turing-reducible to* $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_n)$.

In a second step we show that the number of different stable letters can be reduced to a constant. For this, it is important to note that the associated subgroups $A, B \leq H$ do not belong to the input; so their size is a fixed constant.

Fix the constant $\delta = 2 \cdot |A|! \cdot 2^{|A|}$. Note that the number of HNN-extensions of the form $\langle H, t_1, \ldots, t_k \mid a^{t_i} = \psi_i(a)\ (1 \leq i \leq k, a \in A)\rangle$ with $k \leq \delta$ is constant. The following lemma says that $\mathrm{RUCWP}(H, A, B)$ can be reduced in polynomial time to one of the problems $\mathrm{RCWP}(H, A, B, \psi_1, \ldots, \psi_k)$. Moreover, we can determine in polynomial time, which of these problems arises.

Lemma 3.4. *There exists a polynomial time algorithm for the following:*

INPUT: Partial isomorphisms $\varphi_1, \ldots, \varphi_n : A \to B$ *and SLPs* $\mathbb{A}, \mathbb{B}$ *over the alphabet* $\Sigma^{\pm 1} \cup \{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\}$ *such that* $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(H, \varphi_1, \ldots, \varphi_n)$.

OUTPUT: Partial isomorphisms $\psi_1, \ldots, \psi_k : A \to B$ *where* $k \leq \delta$ *and SLPs* $\mathbb{A}'$, $\mathbb{B}'$ *over the alphabet* $\Sigma^{\pm 1} \cup \{t_1, t_1^{-1}, \ldots, t_k, t_k^{-1}\}$ *such that:*

- *For every* $1 \leq i \leq k$ *there exists* $1 \leq j \leq n$ *with* $\psi_i = \varphi_j$.
- $\mathrm{val}(\mathbb{A}'), \mathrm{val}(\mathbb{B}') \in \mathrm{Red}(H, \psi_1, \ldots, \psi_k)$
- $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ *in* $\langle H, t_1, \ldots, t_n \mid a^{t_i} = \varphi_i(a)\ (1 \leq i \leq n, a \in A)\rangle$ *if and only if* $\mathrm{val}(\mathbb{A}') = \mathrm{val}(\mathbb{B}')$ *in* $\langle H, t_1, \ldots, t_k \mid a^{t_i} = \psi_i(a)\ (1 \leq i \leq k, a \in A)\rangle$.

Due to Lemma 3.4 it suffices to concentrate our effort on problems of the form $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$, where $k \leq \delta$. We have to check whether for two given SLP-compressed reduced strings u and v conditions (a) and (b) from Lemma 2.1 are satisfied. Condition (a) can be easily checked by computing SLPs for $\pi_t(u)$ and $\pi_t(v)$ and then checking for equality using Plandowski's algorithm [12]. The whole difficulty lies in checking condition (b) from Lemma 2.1. Let

$$G_0 = \langle H, t_1, \ldots, t_k \mid a^{t_i} = \varphi_i(a)\ (1 \leq i \leq k, a \in A)\rangle \tag{4}$$

and let us choose $i \in \{1, \ldots, k\}$ such that $|\mathrm{dom}(\varphi_i)|$ is maximal. W.l.o.g. assume that $i = 1$. Let $\mathrm{dom}(\varphi_1) = A_1 \leq A$ and $\mathrm{ran}(\varphi_1) = B_1 \leq B$. We write t for t_1 in the following and define $\Gamma = \Sigma \cup \{t_2, \ldots, t_k\}$. We can write our HNN-extension G_0 from (4) as

$$G_0 = \langle K, t \mid a^t = \varphi_1(a)\ (a \in A_1)\rangle, \text{ where} \tag{5}$$

$$K = \langle H, t_2, \ldots, t_k \mid a^{t_i} = \varphi_i(a)\ (2 \leq i \leq k, a \in A)\rangle. \tag{6}$$

The latter group K is generated by Γ. The main reduction step in our algorithm is expressed in the following lemma:

Lemma 3.5. *$\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$ is polynomial time Turing-reducible to the problems $\mathrm{RCWP}(H, A, B, \varphi_2, \ldots, \varphi_k)$ and $\mathrm{RUCWP}(A_1, A_1, A_1)$.*

Let us briefly sketch the proof of Lemma 3.5: Let $(\mathbb{A}, \mathbb{B})$ be an input for the problem $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$ with $k \leq \delta$. Thus, $\mathbb{A}$ and $\mathbb{B}$ are SLPs over the alphabet $\Sigma^{\pm 1} \cup \{t_1, t_1^{-1}, \ldots, t_k, t_k^{-1}\} = \Gamma^{\pm 1} \cup \{t, t^{-1}\}$ with $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(H, \varphi_1, \ldots, \varphi_k)$. Hence, we also have $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(K, \varphi_1)$. W.l.o.g. we may assume that $\pi_t(\mathrm{val}(\mathbb{A})) = \pi_t(\mathrm{val}(\mathbb{B}))$. This property can be checked in polynomial time using Plandowski's algorithm [12], and if it is not satisfied then we have $\mathrm{val}(\mathbb{A}) \neq \mathrm{val}(\mathbb{B})$ in G_0.

In a first step, we modify the SLPs $\mathbb{A}$ and $\mathbb{B}$ in such a way that in a first step they generate strings of the form $X_0 t^{\alpha_1} X_1 \cdots t^{\alpha_m} X_m$ and $Y_0 t^{\alpha_1} Y_1 \cdots t^{\alpha_m} Y_m$, respectively. Here the X_i and Y_j are nonterminals that generate in a second phase strings over the alphabet $\Gamma^{\pm 1}$. This is possible in polynomial time. Then, we transform our $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$-instance $(\mathbb{A}, \mathbb{B})$ into a compressed word problem for a new group G_1 that is generated by the stable letter t and the symbols $X_1, \ldots, X_m, Y_1, \ldots, Y_m$. Here, the idea is to abstract as far as possible from the concrete structure of the original base group K. In some sense, we only keep those K-relations that are necessary to prove (or disprove) that $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ in the group G_0. These K-relations are translated into relations on the "generic" symbols $X_1, \ldots, X_m, Y_1, \ldots, Y_m$. In order to compute these relations, we need oracle access to $\mathrm{CWP}(K)$ or alternatively (by Lemma 3.3) to $\mathrm{RCWP}(H, A, B, \varphi_2, \ldots, \varphi_k)$. Using Tietze transformations [10], our new group G_1 is finally transformed into an HNN-extension with base group A_1 — this gives us the $\mathrm{RUCWP}(A_1, A_1, A_1)$-instance in Lemma 3.5.

We now apply Lemma 3.4 to the problem $\mathrm{RUCWP}(A_1, A_1, A_1)$ (one of the two target problems in Lemma 3.5). An input for this problem can be reduced in polynomial time to an instance of a problem $\mathrm{RCWP}(A_1, A_1, A_1, \psi_1, \ldots, \psi_k)$, where $\psi_1, \ldots, \psi_k : A_1 \to A_1$ are partial automorphisms and $k \leq \delta$ (we have $k \leq 2|A_1|! \cdot 2^{|A_1|} \leq 2|A|! \cdot 2^{|A|} = \delta$). Hence, we are faced with an HNN-extension of the form $G_2 = \langle A_1, t_1, \ldots, t_k \mid a^{t_i} = \psi_i(a)\, (1 \leq i \leq k, a \in \mathrm{dom}(\psi_i))\rangle$. Next, we separate the (constantly many) stable letters $t_1, \ldots, t_k$ that occur in the $\mathrm{RCWP}(A_1, A_1, A_1, \psi_1, \ldots, \psi_k)$-instance into two sets: $\{t_1, \ldots, t_k\} = S_1 \cup S_2$ where $S_1 = \{t_i \mid \mathrm{dom}(\psi_i) = A_1\}$ and $S_2 = \{t_1, \ldots, t_k\} \setminus S_1$. W.l.o.g. assume that $S_2 = \{t_1, \ldots, t_\ell\}$. Then we can write our HNN-extension G_2 as

$$G_2 = \langle H', t_1, \ldots, t_\ell \mid a^{t_i} = \psi_i(a)\ (1 \leq i \leq \ell, a \in \mathrm{dom}(\psi_i))\rangle, \tag{7}$$

where $H' = \langle A_1, t_{\ell+1}, \ldots, t_k \mid a^{t_i} = \psi_i(a)\ (\ell + 1 \leq i \leq k, a \in A_1)\rangle$. Note that $|\mathrm{dom}(\psi_i)| < |A_1|$ for every $1 \leq i \leq \ell$ and that $A_1 = \mathrm{dom}(\psi_i)$ for every $\ell + 1 \leq i \leq k$. By Lemma 2.3, $\mathrm{CWP}(H')$ can be solved in polynomial time; H' is in fact the semidirect product $A_1 \rtimes_\varphi F(t_{\ell+1}, \ldots, t_k)$, where $\varphi : F(t_{\ell+1}, \ldots, t_k) \to \mathrm{Aut}(A_1)$ is defined by $\varphi(t_i) = \psi_i$. Recall also that A_1 was chosen to be of maximal cardinality among the domains of all partial isomorphisms $\varphi_1, \ldots, \varphi_k$. The following proposition summarizes what we have shown so far:

Proposition 3.6. *Let $\varphi_1, \ldots, \varphi_k : A \to B$ be partial isomorphisms, where $k \leq \delta$, $A_1 = \mathrm{dom}(\varphi_1)$, and w.l.o.g $|A_1| \geq |\mathrm{dom}(\varphi_i)|$ for $1 \leq i \leq k$. From an instance $(\mathbb{A}, \mathbb{B})$ of the problem* $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$ *we can compute in polynomial time with oracle access to the problem* $\mathrm{RCWP}(H, A, B, \varphi_2, \ldots, \varphi_k)$

(1) a semidirect product $A_1 \rtimes_\varphi F$, where F is a free group of rank at most δ,
(2) partial automorphisms $\psi_1, \ldots, \psi_\ell : A_1 \to A_1$ with $\ell \leq \delta$ and $|\mathrm{dom}(\psi_i)| < |A_1|$ for all $1 \leq i \leq \ell$, and
(3) an $\mathrm{RCWP}(A_1 \rtimes_\varphi F, A_1, A_1, \psi_1, \ldots, \psi_\ell)$*-instance, which is positive if and only if the initial* $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$*-instance* $(\mathbb{A}, \mathbb{B})$ *is positive.*

Note that in (1) there are only constantly many semidirect products of the form $A_1 \rtimes_\varphi F$ and that $\mathrm{CWP}(A_1 \rtimes_\varphi F)$ can be solved in polynomial time by Lemma 2.3. We are now ready to prove the main theorem of this paper.

Proof of Theorem 3.1. By Lemma 3.2 and Lemma 3.4 it suffices to solve a problem $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_\delta)$ in polynomial time. For this we apply Proposition 3.6 repeatedly. We obtain a computation tree, where the root is labeled with an $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_\delta)$-instance and every other node is labeled with an instance of a problem $\mathrm{RCWP}(C \rtimes_\varphi F, C, C, \theta_1, \ldots, \theta_p)$, where F is a free group of rank at most δ, C is a subgroup of our finite group A, and $p \leq \delta$. The number of these problems is bounded by some fixed constant. Since along each edge in the tree, either the number of stable letters reduces by one, or the maximal size of an associated subgroup becomes strictly smaller, the height of the tree is bounded by a constant (it is at most $|A| \cdot \delta = 2 \cdot |A| \cdot |A|! \cdot 2^{|A|}$). Moreover, along each tree edge, the size of a problem instance can grow only polynomially. Hence, each problem instance that appears in the computation tree has polynomial size w.r.t. the input size. Hence, the total running time is bounded polyomially. □

4 Amalgamated Products

Let H_1 and H_2 be two finitely generated groups. Let $A_1 \leq H_1$ and $A_2 \leq H_2$ be finite and $\varphi : A_1 \mapsto A_2$ an isomorphism. The *amalgamated free product of H_1 and H_2, amalgamating the subgroups A_1 and A_2 by the isomorphism φ*, is the group $G = \langle H_1 * H_2 \mid a = \varphi(a)\ (a \in A_1)\rangle$.

Theorem 4.1. *Let $G = \langle H_1 * H_2 \mid a = \varphi(a)\ (a \in A_1)\rangle$ be an amalgamated free product with A_1 finite. Then* $\mathrm{CWP}(G) \leq_T^P \{\mathrm{CWP}(H_1), \mathrm{CWP}(H_2)\}$.

Proof. It is well known [10, Theorem 2.6, p. 187] that G can be embedded into the HNN-extension $G' = \langle H_1 * H_2, t \mid a^t = \varphi(a)\ (a \in A_1)\rangle$ by the homomorphism Φ with $\Phi(x) = t^{-1}xt$ for $x \in H_1$ and $\Phi(x) = x$ for $x \in H_2$. Given an SLP $\mathbb{A}$ we can easily compute an SLP $\mathbb{B}$ with $\mathrm{val}(\mathbb{B}) = \Phi(\mathrm{val}(\mathbb{A}))$. We obtain: $\mathrm{val}(\mathbb{A}) = 1$ in $G \iff \Phi(\mathrm{val}(\mathbb{A})) = 1$ in $\Phi(G) \iff \mathrm{val}(\mathbb{B}) = 1$ in G'. By Theorem 3.1 and Theorem 2.2, $\mathrm{CWP}(G')$ can be solved in polynomial time with oracle access to $\mathrm{CWP}(H_1)$ and $\mathrm{CWP}(H_2)$. □

5 Open Problems

We have shown that the compressed word problem for an HNN-extension with finite associated subgroups is polynomial time Turing-reducible to the compressed word problem for the base group. Here, the base group and the associated subgroups are fixed, i.e. are not part of the input. One might also consider the *uniform* compressed word problem for HNN-extensions of the form $\langle H, t \mid a^t = \varphi(a)\ (a \in A)\rangle$, where H is a finite group that is part of the input. It is not clear, whether this problem can be solved in polynomial time. Finally, one might also consider the compressed word problem for HNN-extensions of semigroups [4].

References

1. W. Dicks and M. J. Dunwoody. *Groups Acting on Graphs.* Cambridge University Press, 1989.
2. G. Higman, B. H. Neumann, and H. Neumann. Embedding theorems for groups. *Journal of the London Mathematical Society. Second Series*, 24:247–254, 1949.
3. N. Haubold and M. Lohrey. Compressed word problems in HNN-extensions and amalgamated products. arXiv.org, 2008. `http://arxiv.org/abs/0811.3303`.
4. J. M. Howie. Embedding theorems for semigroups. *Quart. J. Math. Oxford Ser. (2)*, 14:254–258, 1963.
5. I. Kapovich, A. Myasnikov, P. Schupp, and V. Shpilrain. Generic-case complexity, decision problems in group theory, and random walks. *J. Algebra*, 264(2):665–694, 2003.
6. M. Lohrey. Word problems and membership problems on compressed words. *SIAM J. Comput.*, 35(5):1210 – 1240, 2006.
7. M. Lohrey and S. Schleimer. Efficient computation in groups via compression. In *Proc. CSR 2007*, LNCS 4649, pages 249–258. Springer, 2007.
8. M. Lohrey and G. Sénizergues. Theories of HNN-extensions and amalgamated products. In *Proc. ICALP 2006*, LNCS 4052, pages 681–692. Springer, 2006.
9. M. Lohrey and G. Sénizergues. Rational subsets in HNN-extensions and amalgamated products. *Internat. J. Algebra Comput.*, 18(1):111–163, 2008.
10. R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory.* Springer, 1977.
11. A. Myasnikov, V. Shpilrain, and A. Ushakov. *Group-based Cryptography.* Birkhäuser, 2008.
12. W. Plandowski. Testing equivalence of morphisms on context-free languages. In *Proc.ESA'94*, LNCS 855, pages 460–470. Springer, 1994.
13. W. Plandowski and W. Rytter. Complexity of language recognition problems for compressed words. In *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 262–272. Springer, 1999.
14. S. Schleimer. Polynomial-time word problems. *Comment. Math. Helv.*, 83(4):741–765, 2008.
15. J. R. Stallings. *Group Theory and Three-Dimensional Manifolds.* Number 4 in Yale Mathematical Monographs. Yale University Press, 1971.