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 [3], 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 2.1 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 [19] and independently Boone [2] proved the existence of a finitely presented group with undecidable word problem. 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 [18].

In order to prove upper bounds on the complexity of the word problem for a certain group G, a "compressed" variant of the word problem for G was introduced in [11, 12, 23]. In the *compressed word problem* for G, the input word over the generators is not given explicitly but succinctly via a so called *straight-line program* (SLP for short). This is a context free grammar that generates exactly one word, see Section 2.2. 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, see e.g. [1, 5, 10, 11, 17, 20, 21]. In [12, 23] 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 [23], it was shown that the

[⋆] The work of the second author is supported by the DFG research project ALKODA.

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 [23], which solved an open problem from [9]. Generalizations of this result can be found in [12, 16].

In this paper, we prove a transfer theorem for the compressed word problem of *HNN-extensions* [7]. For a base group H with two isomorphic subgroups A and B 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

$$G = \langle H_1 * H_2 \mid a = \varphi(a) \, (a \in A_1) \rangle.$$

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* (or *identified*) subgroups.

HNN-extensions were introduced by Higman, Neumann, and Neumann in 1949 [7]. 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 [15]. 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. identified) subgroups, which is an important subcase. Stallings proved [24] that a group has more than one end if and only if it is either an HNN-extension with finite associated subgroups or an amalgamated free product with finite identified 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 identified subgroups and HNN-extensions with finite associated subgroups [4].

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 (Section 3.1), 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 $t_i^{-1} w t_i$ (resp. $t_i w t_i^{-1}$), where the string w represents a group element from the domain (resp. range) of φ_i. In a second step (Section 3.2) 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 δ many stable letters (and associated partial isomorphisms from A to B) into two simpler problems: (i) the same problem but with only $\delta - 1$ 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 [13], 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.

2 Preliminaries

Let Σ be a finite alphabet. The empty word is denoted by ε. With $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ we denote the set of non-empty words over Σ. For a word $w = a_1 \cdots a_n$ let $|w| = n$, $\mathrm{alph}(w) = \{a_1, \ldots, a_n\}$, and $w[i:j] = a_i \cdots a_j$ for $1 \le i \le j \le n$. Moreover, let $w[i:] = w[i:n]$ and $w[:i] = w[1:i]$.

2.1 Groups and the word problem

For background in combinatorial group theory see [15]. 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, $\widetilde{\Sigma} = \Sigma \cup \Sigma^{-1}$ is a finite *monoid generating set* for G and there exists a canonical monoid homomorphism $h : \widetilde{\Sigma}^* \to G$, which maps a word $w \in \widetilde{\Sigma}^*$ to the group element represented by w. For $u, v \in \widetilde{\Sigma}^*$ we will also say that $u = v$ in G in case $h(u) = h(v)$. The *word problem for G with respect to Σ* is the following decision problem:

INPUT: A word $w \in \widetilde{\Sigma}^*$.

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 with respect to Σ is logspace many-one reducible to the word problem for G with respect to 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

$$F(\Sigma) = \widetilde{\Sigma}^*/\{aa^{-1} = \varepsilon \mid a \in \widetilde{\Sigma}\}.$$

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 $u, v \in \widetilde{\Sigma}^*$. 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* $R \subseteq \widetilde{\Sigma}^+$, 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 $G_1 * G_2 \simeq \langle \Sigma_1 \cup \Sigma_2 \mid R_1 \cup R_2 \rangle$.

The following transformations on group presentations (in either direction) are known as *Tietze transformations*:

$$\begin{aligned} (\Sigma, R) &\leftrightarrow (\Sigma, R \cup \{u = v\}) && \text{if } uv^{-1} \in N(R) \\ (\Sigma, R) &\leftrightarrow (\Sigma \cup \{a\}, R \cup \{a = w\}) && \text{if } a \notin \widetilde{\Sigma}, w \in \widetilde{\Sigma}^* \end{aligned}$$

If (Σ', R') can be obtained by a sequence of Tietze transformations from (Σ, R), then $\langle \Sigma \mid R \rangle \simeq \langle \Sigma' \mid R' \rangle$ [15].

2.2 Straight-line programs

We are using straight-line programs as a compressed representation of strings with reoccuring subpatterns [22]. 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 $\{(X, Y) \in V \times V \mid \exists \alpha : (X, \alpha) \in P, Y \in \mathrm{alph}(\alpha)\}$. 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 $a \in \Gamma$. The following tasks can be solved in polynomial time. Except for the last one, proofs are straightforward.

– Given an SLP $\mathbb{A}$, calculate $|\mathrm{val}(\mathbb{A})|$.
– Given an SLP $\mathbb{A}$ and a natural number $i \leq |\mathrm{val}(\mathbb{A})|$, calculate $\mathrm{val}(\mathbb{A})[i]$.
– Given SLPs $\mathbb{A}$ and $\mathbb{B}$ decide whether $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ [20].

A *deterministic rational transducer* is a 5-tuple $T = (\Sigma, \Gamma, Q, \delta, q_0, F)$, where Σ is the input alphabet, Γ is the output alphabet, Q is the set of states, $\delta : Q \times \Sigma \to Q \times \Gamma^*$ is the partial transition function, $q_0 \in Q$ is the initial state, and $F \subseteq Q$ is the set of final states. Let $\widehat{\delta} : Q \times \Sigma^* \to Q \times \Gamma^*$ be the canonical extension of δ. The partial mapping defined by T is $[\![T]\!] = \{(u, v) \in \Sigma^* \times \Gamma^* \mid \widehat{\delta}(q_0, u) \in F \times \{v\}\}$. A proof of the following lemma can be found in [1].

Lemma 2.1. *From a given SLP $\mathbb{A}$ and a given deterministic rational transducer T we can compute in polynomial time an SLP for the string $[\![T]\!](\mathrm{val}(\mathbb{A}))$ (if it is defined).*

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

INPUT: An SLP $\mathbb{A}$ over the terminal alphabet $\widetilde{\Sigma}$.

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 $\mathrm{CWP}(G)$.

A *composition system* $\mathbb{A} = (V, \Gamma, S, P)$ is an SLP, which additionally allows productions of the form $A \to B[i : j]$ where $1 \leq i \leq j \leq |\mathrm{val}(B)|$ [5]. For such a production we define $\mathrm{val}(A) = \mathrm{val}(B)[i : j]$. In [6], Hagenah presented a polynomial time algorithm that transforms a given composition system into an SLP that generates the same word.

2.3 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).

2.4 HNN-extensions

Let us fix throughout this section a *base group* $H = \langle \Sigma \mid R \rangle$. Let us also fix isomorphic subgroups $A_i, B_i \leq H$ $(1 \leq i \leq n)$ and isomorphisms $\varphi_i : A_i \to B_i$.

Let $h : \widetilde{\Sigma}^* \to H$ be the canonical morphism, which maps a word $w \in \widetilde{\Sigma}^*$ to the element of H it represents. We consider the *HNN-extension*

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

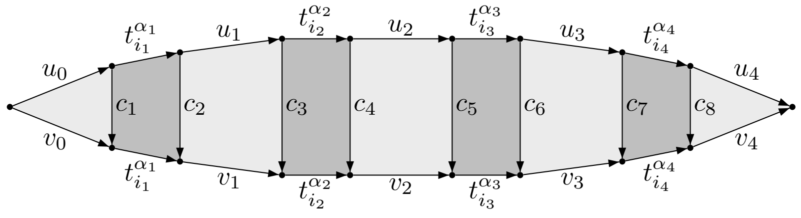
This means that $G = \langle \Sigma \cup \{t_1, \ldots, t_n\} \mid R \cup \{a^{t_i} = \varphi_i(a) \mid 1 \leq i \leq n, a \in A_i\} \rangle$. It is known that the base group H naturally embeds into G [7]. In this paper, we will be only concerned with the case that all groups $A_1, \ldots, A_n$ are finite 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 $\alpha \in \{+1, -1\}$.

We say that a word $u \in (\widetilde{\Sigma} \cup \{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\})^*$ is *reduced* if u does not contain a factor of the form $t_i^{-\alpha} w t_i^\alpha$ for $\alpha \in \{1, -1\}$, $w \in \widetilde{\Sigma}^*$ 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 $u \in (\widetilde{\Sigma} \cup \{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\})^*$ 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 an HNN-extension [14]:

Lemma 2.2. *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 $u_0, \ldots, u_\ell, v_0, \ldots, v_m \in \widetilde{\Sigma}^*$, $\alpha_1, \ldots, \alpha_\ell, \beta_1, \ldots, \beta_m \in \{1, -1\}$, 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:*

- *$\pi_t(u) = \pi_t(v)$ (i.e., $\ell = m$, $i_k = j_k$, and $\alpha_k = \beta_k$ for $1 \leq k \leq \ell$)*
- *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 $0 \leq k \leq \ell$ (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 $1 \leq k \leq \ell$.*

The second condition of the lemma can be visualized by a diagram of the following form (also called a Van Kampen diagram, see [15] for more details), where $\ell = m = 4$. Light-shaded (resp. dark-shaded) faces represent relations in H (resp. relations of the form $c t_i^\alpha = t_i^\alpha \varphi_i^\alpha(c)$ with $c \in A_i(\alpha)$).



(†)

The elements $c_1, \ldots, c_{2\ell}$ in such a diagram are also called *connecting elements.*

When solving the compressed word problem for HNN-extensions we will make use of the following simple lemma, which allows us to transform an arbitrary string over the generating set of an HNN-extension into a reduced one.

Lemma 2.3. *Assume that $u = u_0 t_{i_1}^{\alpha_1} u_1 \cdots t_{i_n}^{\alpha_n} u_n$ and $v = v_0 t_{j_1}^{\beta_1} v_1 \cdots t_{j_m}^{\beta_m} v_m$ are reduced strings. Let $d(u, v)$ be the largest number $d \geq 0$ such that* *(a)* $A_{i_{n-d+1}}(\alpha_{n-d+1}) = A_{j_d}(-\beta_d)$ *(we set* $A_{i_{n+1}}(\alpha_{n+1}) = A_{j_0}(-\beta_0) = 1$*) and*

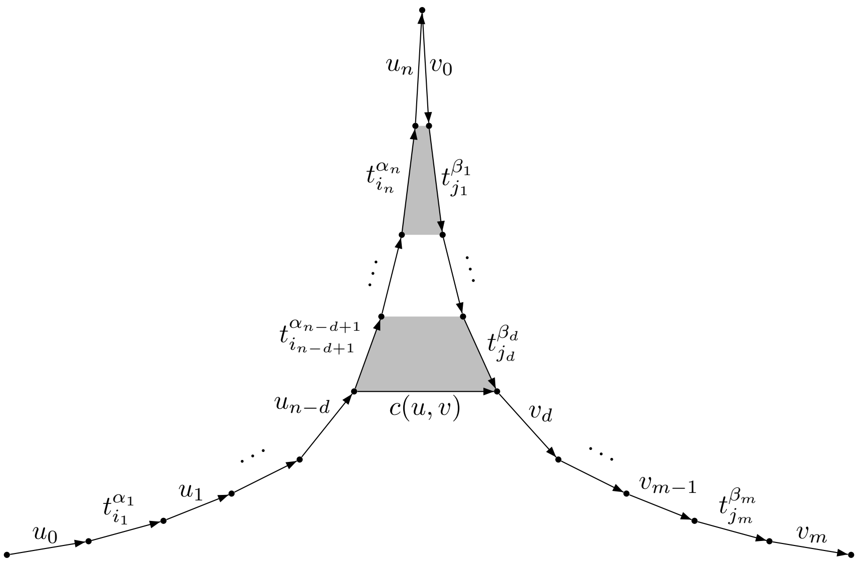
(b) $\exists c \in A_{j_d}(-\beta_d) : t_{i_{n-d+1}}^{\alpha_{n-d+1}} u_{n-d+1} \cdots t_{i_n}^{\alpha_n} u_n v_0 t_{j_1}^{\beta_1} \cdots v_{d-1} t_{j_d}^{\beta_d} = c$ *in the group* G *from (2) (note that this condition is satisfied for* $d = 0$*).*

Moreover, let $c(u,v) \in A_{j_d}(-\beta_d)$ *be the element* c *in (b) (for* $d = d(u,v)$*). Then*

$$u_0 t_{i_1}^{\alpha_1} u_1 \cdots t_{i_{n-d(u,v)}}^{\alpha_{n-d(u,v)}} (u_{n-d(u,v)}\, c(u,v)\, v_{d(u,v)}) t_{j_{d(u,v)+1}}^{\beta_{d(u,v)+1}} v_{d(u,v)+1} \cdots t_{j_m}^{\beta_m} v_m$$

is a reduced string equal to uv *in* G*.*

The above lemma can be visualized by the following diagram.



2.5 Some simple compressed word problems

We will use the following theorem on free products $G_1 * G_2$ that was shown in [12].

Theorem 2.4. $\mathrm{CWP}(G_1 * G_2) \leq_T^P \{\mathrm{CWP}(G_1), \mathrm{CWP}(G_2)\}$.

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 2.5. *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* $\langle H, t_1, \dots, t_n \mid h^{t_i} = \varphi_i(h)\ (1 \leq i \leq n, h \in H)\rangle$*?*

Proof. Let $s \in (H \cup \{t_1, t_1^{-1}, \dots t_n, t_n^{-1}\})^*$. From the defining equations of the group $G = \langle H, t_1, \dots, t_n \mid h^{t_i} = \varphi_i(h)\ (1 \leq i \leq n, h \in H)\rangle$ it follows that there exists a unique $h \in H$ with $s = \pi_t(s)h$ in G. Hence, $s = 1$ in G if and only if $\pi_t(s) = 1$ in the free group $F(t_1, \dots, t_n)$ and $h = 1$ in H.

Now, let $\mathbb{A}$ be an SLP over the alphabet $H \cup \{t_1, t_1^{-1}, \dots t_n, t_n^{-1}\}$. W.l.o.g. assume that $\mathbb{A}$ is in Chomsky normal form. It is straightforward to compute an SLP for the projection $\pi_t(\mathrm{val}(\mathbb{A}))$. Since by Theorem 2.4 the word problem for the free group $F(t_1, \dots, t_n)$ can be solved in polynomial time, it suffices to compute for every nonterminal A of $\mathbb{A}$ the unique $h_A \in H$ with $\mathrm{val}(A) = \pi_t(\mathrm{val}(A))h_A$ in G. We compute the elements h_A bottom up. The case that the right-hand side for A is a terminal symbol from $H \cup \{t_1, t_1^{-1}, \dots t_n, t_n^{-1}\}$ is clear. Hence, assume that $A \to BC$ is a production of $\mathbb{A}$ and assume that $h_B, h_C \in H$ are already computed. Hence, in G we have:

$$\mathrm{val}(A) = \mathrm{val}(B)\mathrm{val}(C) = \pi_t(\mathrm{val}(B))h_B\pi_t(\mathrm{val}(C))h_C.$$

Thus, it suffices to compute the unique $h \in H$ with $h_B\pi_t(\mathrm{val}(C)) = \pi_t(\mathrm{val}(C))h$ in G. Note that if $\pi_t(\mathrm{val}(C)) = t_{i_1}^{\alpha_1} t_{i_2}^{\alpha_2} \cdots t_{i_n}^{\alpha_n}$, then

$$h = \varphi_{i_n}^{\alpha_n}(\cdots \varphi_{i_2}^{\alpha_2}(\varphi_{i_1}^{\alpha_1}(h_B))\cdots) = (\varphi_{i_1}^{\alpha_1} \circ \cdots \circ \varphi_{i_n}^{\alpha_n})(h_B).$$

The automorphism $f = \varphi_{i_1}^{\alpha_1} \circ \cdots \circ \varphi_{i_n}^{\alpha_n}$ can be easily computed from an SLP $\mathbb{C}$ for the string $\pi_t(\mathrm{val}(C))$ by replacing in $\mathbb{C}$ the terminal symbol t_i (resp. t_i^{-1}) by φ_i (resp. φ_i^{-1}). This allows to compute f bottom-up and then to compute $f(h_B)$. ☐

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 will prove 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 will 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 finitely generated 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

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

we assume implicitly that every φ_i is in fact an isomorphism between subgroups $A_1 \leq A$ and $B_1 \leq B$. Hence, φ_i can be viewed as a *partial* isomorphism from our fixed subgroup A to our fixed subgroup B, and (3) is in fact an abbreviation for the group

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

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 CWP(G). Our most general problem is the following computational problem 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 $\widetilde{\Sigma} \cup \{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\}$.
QUESTION: $\mathrm{val}(\mathbb{A}) = 1$ in $\langle H, t_1, \ldots, t_n \mid a^{t_i} = \varphi_i(a) \ (1 \leq i \leq n, a \in A)\rangle$?

The restriction of this problem UCWP(H, A, B) to reduced input strings is denoted by 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 $\widetilde{\Sigma} \cup \{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\}$ such that $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(H, \varphi_1, \ldots, \varphi_n)$.
QUESTION: $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ in $\langle H, t_1, \ldots, t_n \mid a^{t_i} = \varphi_i(a) \ (1 \leq i \leq n, a \in A)\rangle$?

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

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

Our main result is:

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

The rest of Section 3 is concerned with the proof of Theorem 3.1.

3.1 Reducing to reduced sequences

First we show that we may restrict ourselves to SLPs that evaluate to reduced strings:

Lemma 3.2. UCWP(H, A, B) $\leq_P^T$ RUCWP(H, A, B). *More precisely, there is a polynomial time Turing-reduction from* UCWP(H, A, B) *to* RUCWP(H, A, B) *that on input* ($\varphi_1, \ldots, \varphi_n, \mathbb{A}$) *only asks* 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).*

Proof. Consider partial isomorphisms $\varphi_i : A \to B$ $(1 \le i \le n)$ and let

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

Moreover, let $\mathbb{A}$ be an SLP in Chomsky normal form over the alphabet $\widetilde{\Sigma} \cup \{t_1, t_1^{-1}, \ldots, t_n, t_n^{-1}\}$. Using oracle access to RUCWP(H, A, B), we will construct bottom-up a *composition system* $\mathbb{A}'$ with $\mathrm{val}(\mathbb{A}') = \mathrm{val}(\mathbb{A})$ in G and $\mathrm{val}(\mathbb{A}')$ reduced, on which finally the RUCWP(H, A, B)-oracle can be asked whether $\mathrm{val}(\mathbb{A}') = 1$ in G. The system $\mathbb{A}'$ has the same variables as $\mathbb{A}$ but for every variable X, $\mathrm{val}(\mathbb{A}', X)$ is reduced and $\mathrm{val}(\mathbb{A}', X) = \mathrm{val}(\mathbb{A}, X)$ in G.

Assume that $X \to YZ$ is a production of $\mathbb{A}$, where Y and Z were already processed during our bottom-up reduction process. Hence, $\mathrm{val}(Y)$ and $\mathrm{val}(Z)$ are reduced. Let

$$\mathrm{val}(Y) = u_0 t_{i_1}^{\alpha_1} u_1 \cdots t_{i_\ell}^{\alpha_\ell} u_\ell \quad \text{and} \quad \mathrm{val}(Z) = v_0 t_{j_1}^{\beta_1} v_1 \cdots t_{j_m}^{\beta_m} v_m.$$

with $u_0, \ldots, u_\ell, v_0, \ldots, v_m \in \widetilde{\Sigma}^*$. For $1 \le k \le \ell$ (resp. $1 \le k \le m$) let $p(k)$ (resp. $q(k)$) be the unique position within $\mathrm{val}(Y)$ (resp. $\mathrm{val}(Z)$) such that $\mathrm{val}(Y)[: p(k)] = u_0 t_{i_1}^{\alpha_1} u_1 \cdots t_{i_k}^{\alpha_k}$ (resp. $\mathrm{val}(Z)[: q(k)] = v_0 t_{j_1}^{\beta_1} v_1 \cdots t_{j_k}^{\beta_k}$). These positions can be computed in polynomial time from k using simple arithmetic.

According to Lemma 2.3 it suffices to find $d = d(\mathrm{val}(Y), \mathrm{val}(Z)) \in \mathbb{N}$ and $c = c(\mathrm{val}(Y), \mathrm{val}(Z)) \in A \cup B$ in polynomial time. This can be done, using binary search: First, compute $\min\{l, m\}$. For a given number $k \le \min\{\ell, m\}$ we want to check whether

$$t_{i_{\ell-k+1}}^{\alpha_{\ell-k+1}} u_{\ell-k+1} \cdots t_{i_\ell}^{\alpha_\ell}\, u_\ell\, v_0\, t_{j_1}^{\beta_1} \cdots v_{k-1} t_{j_k}^{\beta_k} \in A_{i_{\ell-k+1}}(\alpha_{\ell-k+1}) = A_{j_k}(-\beta_k) \quad (4)$$

in the group G. Note that (4) is equivalent to $t_{i_{\ell-k+1}}^{\alpha_{\ell-k+1}} = t_{j_k}^{-\beta_k}$ and

$$\bigvee_{c \in A_{j_k}(-\beta_k)} \mathrm{val}(Y)[p(\ell-k+1) :]^{-1} c = \mathrm{val}(Z)[: q(k)]. \quad (5)$$

The two sides of this equation are reduced strings and the number of possible values $c \in A_{j_k}(-\beta_k)$ is bounded by a constant. Hence, (5) is equivalent to a constant number of RUCWP(H, A, B)-instances that can be computed in polynomial time.

In order to find with binary search the value d (i.e. the largest $k \ge 0$ such that (4) holds), one has to observe that (4) implies that (4) also holds for every smaller value k (this follows from Lemma 2.2). From d, we can compute in polynomial time positions $p(\ell - d + 1)$ and $q(d)$. Then, according to Lemma 2.3, the string

$$\mathrm{val}(Y)[: p(\ell-d+1) - 1]\, c\, \mathrm{val}(Z)[q(d) + 1 :]$$

is reduced and equal to $\mathrm{val}(Y)\mathrm{val}(Z)$ in G. Hence, we can replace the production $X \to YZ$ by $X \to Y[: p(\ell-d+1) - 1]\, c\, Z[q(d) + 1 :]$. □

The above proof can be also used in order to derive:

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

3.2 Reduction to a constant number of stable letters

In this section, 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 $\delta = 2 \cdot |A|! \cdot 2^{|A|}$ for the rest of the paper. 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 $\widetilde{\Sigma} \cup \{t_1, t_1^{-1}, \ldots t_n, t_n^{-1}\}$ 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 $\widetilde{\Sigma} \cup \{t_1, t_1^{-1}, \ldots t_k, t_k^{-1}\}$ 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 $\langle H, t_1, \ldots, t_k \mid a^{t_i} = \psi_i(a) \ (1 \leq i \leq k, a \in A)\rangle$.*

Proof. Fix an input $(\varphi_1, \ldots, \varphi_n, \mathbb{A}, \mathbb{B})$ for the problem $\mathrm{RUCWP}(H, A, B)$. In particular, $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(H, \varphi_1, \ldots, \varphi_n)$. Define the function $\tau : \{1, \ldots, n\} \to \{1, \ldots, n\}$ by

$$\tau(i) = \min\{k \mid \varphi_k = \varphi_i\}.$$

This mapping can be easily computed in polynomial time from the sequence $\varphi_1, \ldots, \varphi_n$. Assume w.l.o.g. that $\mathrm{ran}(\tau) = \{1, \ldots, \gamma\}$ for some $\gamma \leq n$. Note that $\gamma \leq |A|! \cdot 2^{|A|} = \frac{\delta}{2}$. For every t_i $(1 \leq i \leq \gamma)$ we take two stable letters $t_{i,0}$ and $t_{i,1}$. Hence, the total number of stable letters is at most δ. Moreover, we define a sequential transducer T which, reading as input the word $u_0 t_{i_1}^{\alpha_1} u_1 \cdots t_{i_m}^{\alpha_m} u_m$ (with $u_0, \ldots, u_m \in \widetilde{\Sigma}^+$ and $1 \leq i_1, \ldots, i_m \leq n$) returns

$$[\![T]\!](w) = u_0 \, t_{\tau(i_1),1}^{\alpha_1} \, u_1 \, t_{\tau(i_2),0}^{\alpha_2} \, u_2 \, t_{\tau(i_3),1}^{\alpha_3} \, u_3 \cdots t_{\tau(i_m),m \bmod 2}^{\alpha_m} \, u_m.$$

Finally, we define the HNN-extension

$$G' = \langle H, t_{1,0}, t_{1,1}, \ldots, t_{\gamma,0}, t_{\gamma,1} \mid a^{t_{i,k}} = \varphi_i(a) \ (1 \leq i \leq \gamma, k \in \{0,1\}, a \in A)\rangle.$$

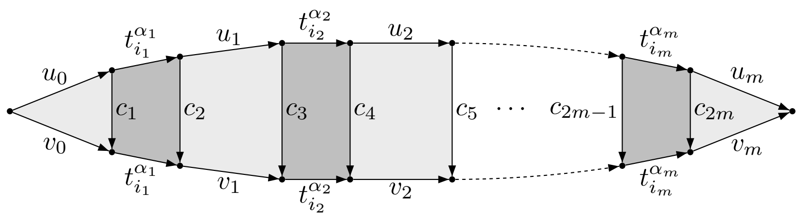
This HNN-extension has $2\gamma \leq \delta$ many stable letters; it is the HNN-extension $\langle H, t_1, \ldots, t_k \mid a^{t_i} = \psi_i(a) \ (1 \leq i \leq k, a \in A)\rangle$ from the lemma.

Claim: Let $u, v \in \mathrm{Red}(H, \varphi_1, \ldots, \varphi_n)$ be reduced. Then also $[\![T]\!](u)$ and $[\![T]\!](v)$ are reduced. Moreover, the following are equivalent:

(a) $u = v$ in $\langle H, t_1, \ldots, t_n \mid a^{t_i} = \varphi_i(a) \ (1 \leq i \leq n, a \in A)\rangle$
(b) $[\![T]\!](u) = [\![T]\!](v)$ in the HNN-extension G' and $\pi_t(u) = \pi_t(v)$.

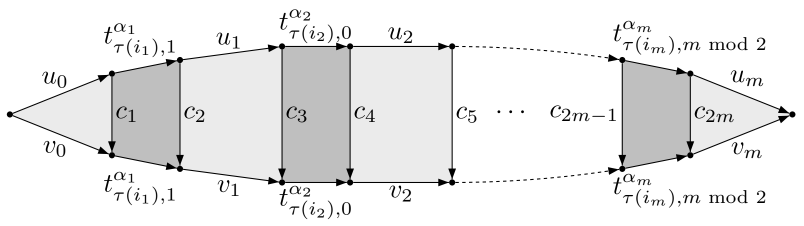
Proof of the claim. Let $u = u_0 t_{i_1}^{\alpha_1} u_1 \cdots t_{i_\ell}^{\alpha_\ell} u_\ell$ and $v = v_0 t_{j_1}^{\beta_1} v_1 \cdots t_{j_m}^{\beta_m} v_m$. The first statement is obvious due to the fact that $[\![T]\!](u)$ does not contain a subword of the form $t_{i,k}^{\alpha} w t_{j,k}^{\beta}$ for $k \in \{0,1\}$, and similarly for $[\![T]\!](v)$.

For $(a) \Rightarrow (b)$ note that by Lemma 2.2, $u = v$ in $\langle H, t_1, \ldots, t_n \mid a^{t_i} = \varphi_i(a)\ (1 \le i \le n, a \in A)\rangle$ implies $\pi_t(u) = \pi_t(v)$ (i.e. $\ell = m$, $\alpha_1 = \beta_1, \ldots, \alpha_m = \beta_m$, $i_1 = j_1, \ldots, i_m = j_m$), and that there exists a Van Kampen diagram of the following form:



(†)

The defining equations of G' imply that the following is a valid Van Kampen diagram in G':


$t_{\tau(i_1),1}^{\alpha_1}$
u_1
$t_{\tau(i_2),0}^{\alpha_2}$
u_2
$t_{\tau(i_m),m \bmod 2}^{\alpha_m}$
u_0
u_m
c_1
c_2
c_3
c_4
c_5
$\cdots$
c_{2m-1}
c_{2m}
v_0
v_m
$t_{\tau(i_1),1}^{\alpha_1}$
v_1
$t_{\tau(i_2),0}^{\alpha_2}$
v_2
$t_{\tau(i_m),m \bmod 2}^{\alpha_m}$
(‡)

Hence, $[\![T]\!](u) = [\![T]\!](v)$ in G'.

For $(b) \Rightarrow (a)$ note that we have already seen that $[\![T]\!](u)$ and $[\![T]\!](v)$ are reduced. Hence, $[\![T]\!](u) = [\![T]\!](v)$ in G' together with $\pi_t(u) = \pi_t(v)$ implies that there exists a Van Kampen diagram of the form (‡). Again, we can replace the dark-shaded t-faces by the corresponding t-faces of G in order to obtain a diagram of the form (†). This proofs the claim.

By the previous claim, $[\![T]\!](\mathrm{val}(\mathbb{A}))$ and $[\![T]\!](\mathrm{val}(\mathbb{B}))$ are reduced. Moreover, SLPs $\mathbb{A}'$ and $\mathbb{B}'$ for these strings can be computed in polynomial time by Lemma 2.1. In case $\pi_t(\mathrm{val}(\mathbb{A})) \neq \pi_t(\mathrm{val}(\mathbb{B}))$ we choose these SLPs such that e.g. $\mathrm{val}(\mathbb{A}') = t_1$ and $\mathrm{val}(\mathbb{B}') = t_1^{-1}$. Hence, $\mathrm{val}(\mathbb{A}') = \mathrm{val}(\mathbb{B}')$ in G' if and only if $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ in $\langle H, t_1, \ldots, t_n \mid a^{t_i} = \varphi_i(a)(1 \le i \le n, a \in A)\rangle$. This proves the lemma. $\square$

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 $k \le \delta$. Let

$$G_0 = \langle H, t_1, \ldots, t_k \mid a^{t_i} = \varphi_i(a)\ (1 \le i \le k, a \in A)\rangle \tag{6}$$

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 $\mathrm{dom}(\varphi_1) = A_1 \le A$ and $\mathrm{ran}(\varphi_1) = B_1 \le B$. 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 (6) as

$$G_0 = \langle K, t \mid a^t = \varphi_1(a) \ (a \in A_1)\rangle, \tag{7}$$

where

$$K = \langle H, t_2, \dots, t_k \mid a^{t_i} = \varphi_i(a) \ (2 \le i \le k, a \in A)\rangle. \tag{8}$$

The latter group K is generated by Γ. The goal of the next three Sections 3.3–3.5 is to prove:

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

3.3 Abstracting from the base group K

Our aim in this subsection will be to reduce the compressed word problem for G_0 to the compressed word problem for another group, where we have abstracted from most of the concrete structure of the base group K in (8).

Let us consider an input $(\mathbb{A}, \mathbb{B})$ for $\mathrm{RCWP}(H, A, B, \varphi_1, \dots, \varphi_k)$ with $k \le \delta$. W.l.o.g. assume that $k = \delta$. Thus, $\mathbb{A}$ and $\mathbb{B}$ are SLPs over the alphabet $\widetilde{\Sigma} \cup \{t_1, t_1^{-1}, \dots, t_\delta, t_\delta^{-1}\} = \widetilde{\Gamma} \cup \{t, t^{-1}\}$ with $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(H, \varphi_1, \dots, \varphi_\delta)$. 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 [20], and if it is not satisfied then we have $\mathrm{val}(\mathbb{A}) \ne \mathrm{val}(\mathbb{B})$ in G_0. Hence, there are $m \ge 0$, $\alpha_1, \dots, \alpha_m \in \{1, -1\}$, and strings $u_0, v_0 \dots, u_m, v_m \in \widetilde{\Gamma}^*$ such that

$$\mathrm{val}(\mathbb{A}) = u_0 t^{\alpha_1} u_1 \cdots t^{\alpha_m} u_m \text{ and} \tag{9}$$

$$\mathrm{val}(\mathbb{B}) = v_0 t^{\alpha_1} v_1 \cdots t^{\alpha_m} v_m. \tag{10}$$

One might think that the number of different words u_i (resp. v_i) may grow exponentially in the size of $\mathbb{A}$ (resp. $\mathbb{B}$). But we will see that this is actually not the case.

Let us replace every occurrence of t^α ($\alpha \in \{1, -1\}$) in $\mathbb{A}$ and $\mathbb{B}$ by $aa^{-1}t^\alpha aa^{-1}$, where $a \in \Gamma$ is arbitrary. This is to ensure that any two occurrences of symbols from $\{t, t^{-1}\}$ are separated by a non-empty word over $\widetilde{\Gamma}$, i.e., we can assume that $u_0, v_0, \dots, u_m, v_m \in \widetilde{\Gamma}^+$ in (9) and (10).

Our first goal is to transform $\mathbb{A}$ (and similarly $\mathbb{B}$) into an equivalent SLP that generates in a first phase a string of the form $X_0 t^{\alpha_1} X_1 \cdots t^{\alpha_m} X_m$, where X_i is a further variable that generates in a second phase the string $u_i \in \widetilde{\Gamma}^+$. Assume that $\mathbb{A} = (U, \{t, t^{-1}\} \cup \widetilde{\Gamma}, S, P)$ is in Chomsky normal form.

In a first step, we remove every variable $X \in U$ from $\mathbb{A}$ such that $X \to t$ or $X \to t^{-1}$ is a production of $\mathbb{A}$ by replacing X in all right-hand sides of $\mathbb{A}$ by t or t^{-1}, respectively. Now, all productions of $\mathbb{A}$ are of the form $X \to YZ$, $X \to t^\alpha Z$, $X \to Y t^\alpha$, or $X \to x \in \widetilde{\Gamma}$, where $Y, Z \in U$.

Next we split the set U of variables of $\mathbb{A}$ into two parts:

$$U_K^0 = \{X \in U \mid \mathrm{val}(X) \in \widetilde{\Gamma}^+\} \qquad \text{and} \qquad U_t^0 = U \setminus U_K^0.$$

Let P^0_K (resp. P^0_t) be the set of all productions from P with a left-hand side in U^0_K (resp. U^0_t). The subscript K refers to the fact that every nonterminal from U^0_K defines an element from the new base group $K \leq G_0$, whereas the subscript t refers to the fact that every nonterminal from U^0_t generates a string where K-generators as well as t or t^{-1} occurs.

Now we manipulate all productions from P^0_t in a bottom-up process, which adds further variables and productions to U^0_K and P^0_K, respectively. The set U^0_t will not change in the process. After stage i, we have production sets P^i_t and P^i_K, and the set of left-hand sides of P^i_t (resp. P^i_K) is U^0_t (resp. U^i_K). The system $\mathbb{A}^i_t := (U^0_t, \{t, t^{-1}\} \cup U^i_K, S, P^i_t)$ is a *composition system* that generates a string from $(U^i_K)^+ t^{\alpha_1} (U^i_K)^+ \cdots t^{\alpha_m} (U^i_K)^+$.

In stage $i+1$ we do the following: Consider a production $(X \to u) \in P^i_t$ such that every variable in u is already processed, but X is not yet processed. If u is of the form $t^\alpha Z$ or $Y t^\alpha$, then there is nothing to do. Now assume that $u = YZ$ such that Y and Z are already processed. Consider the last symbol $\omega \in \{t, t^{-1}\} \cup U^i_K$ of $\mathrm{val}(\mathbb{A}^i_t, Y)$ and the first symbol $\alpha \in \{t, t^{-1}\} \cup U^i_K$ of $\mathrm{val}(\mathbb{A}^i_t, Z)$ (these symbols can be computed in polynomial time after stage i). If either $\omega \in \{t, t^{-1}\}$ or $\alpha \in \{t, t^{-1}\}$, then again nothing is to do. Otherwise, $\omega, \alpha \in U^i_K$. We now set $U^{i+1}_K = U^i_K \cup \{X'\}$, where X' is a fresh variable, and $P^{i+1}_K = P^i_K \cup \{X' \to \omega\alpha\}$. Finally, we obtain P^{i+1}_t from P^i_t by replacing the production $X \to YZ$ by $X \to Y[:\ell-1]X'Z[2:]$. Here $\ell = |\mathrm{val}(\mathbb{A}^i_t, Y)|$.

After the last stage, we transform the final composition system $\mathbb{A}^k_t$ (where k is the number of stages) into an equivalent SLP, let us denote this SLP by $\mathbb{A}_t$. Moreover, write U_K and P_K for U^k_K and P^k_K. The construction implies that

$$\mathrm{val}(\mathbb{A}_t) = X_0 t^{\alpha_1} X_1 \cdots t^{\alpha_m} X_m \tag{11}$$

with $X_0, \ldots, X_m \in U_K$ and $\mathrm{val}(U_K, \widetilde{\Gamma}, X_i, P_K) = u_i$. Note that the number of different X_i is polynomially bounded, simply because the set U_K was computed in polynomial time. Hence, also the number of different u_i in (9) is polynomially bounded.

For the SLP $\mathbb{B}$ the same procedure yields the following data:

- An SLP $\mathbb{B}_t$ such that
$$\mathrm{val}(\mathbb{B}_t) = Y_0 t^{\alpha_1} Y_1 \cdots t^{\alpha_m} Y_m.$$
- A set of productions Q_K with left-hand sides V_K, where $\{Y_1, \ldots, Y_m\} \subseteq V_K$ and $\mathrm{val}(V_K, \widetilde{\Gamma}, Y_i, Q_K) = v_i$.

W.l.o.g. assume that $U_K \cap V_K = \emptyset$. Let $W_K = U_K \cup V_K$ and $R_K = P_K \cup Q_K$. In the following, for $Z \in W_K$ we write $\mathrm{val}(Z)$ for $\mathrm{val}(W_K, \widetilde{\Gamma}, Z, R_K) \in \widetilde{\Gamma}^+$.

Let us next consider the free product $F(W_K) * A_1 * B_1$. Recall that A_1 (resp. B_1) is the domain (resp. range) of the partial isomorphism φ_1. Clearly, in this free product, A_1 and B_1 have trivial intersection (even if $A_1 \cap B_1 > 1$ in H). We now define a set of defining relations $\mathcal{E}$ by

$$\begin{aligned}\mathcal{E} = \{Z_1 c_1 = c_2 Z_2 \mid{} & Z_1, Z_2 \in W_K, c_1, c_2 \in A_1 \cup B_1, \\ & \mathrm{val}(Z_1)\, c_1 = c_2\, \mathrm{val}(Z_2) \text{ in the group } K\}.\end{aligned} \tag{12}$$

We can compute the set $\mathcal{E}$ in polynomial time using oracle access to $\mathrm{CWP}(K)$ or alternatively, by Lemma 3.3, using oracle access to $\mathrm{RCWP}(H, A, B, \varphi_2, \dots, \varphi_k)$. This is the only time, where we need oracle access to $\mathrm{RCWP}(H, A, B, \varphi_2, \dots, \varphi_k)$ in Lemma 3.5.

Consider the group

$$\begin{aligned} G_1 &= \langle (F(W_K) * A_1 * B_1)/N(\mathcal{E}), t \mid a^t = \varphi_1(a) \ (a \in A_1) \rangle \\ &= \langle F(W_K) * A_1 * B_1, t \mid \mathcal{E},\, t^{-1}at = \varphi_1(a) \, (a \in A_1) \rangle. \end{aligned}$$

Recall that $N(\mathcal{E}) \leq F(W_K) * A_1 * B_1$ is the smallest normal subgroup of $F(W_K) * A_1 * B_1$ that contains all elements xy^{-1} with $(x = y) \in \mathcal{E}$. We can define a morphism

$$\psi : F(W_K) * A_1 * B_1 \to K$$

by $\psi(Z) = \mathrm{val}(Z)$ for $Z \in W_K$, $\psi(a) = a$ for $a \in A_1$, and $\psi(b) = b$ for $b \in B_1$. Of course, the restrictions of ψ to A_1 as well as B_1 are injective. Moreover, each of the defining relations in $\mathcal{E}$ is preserved under ψ: for $(Z_1 c_1 = c_2 Z_2) \in \mathcal{E}$ we have $\psi(Z_1 c_1) = \mathrm{val}(Z_1)\, c_1 = c_2 \,\mathrm{val}(Z_2) = \psi(c_2 Z_2)$ in K. Thus, ψ defines a morphism

$$\widehat{\psi} : (F(W_K) * A_1 * B_1)/N(\mathcal{E}) \to K.$$

Moreover, $A_1 \cap N(\mathcal{E}) = 1$: if $a \in N(\mathcal{E}) \cap A_1$ then $\psi(a) \in \psi(N(\mathcal{E})) = 1$; thus $a = 1$, since ψ is injective on A_1. Similarly, $B_1 \cap N(\mathcal{E}) = 1$. This means that A_1 and B_1 can be naturally embedded in $(F(W_K) * A_1 * B_1)/N(\mathcal{E})$ and $\varphi_1 : A_1 \to B_1$ can be considered as an isomorphism between the images of this embedding in $(F(W_K) * A_1 * B_1)/N(\mathcal{E})$. Therefore, the group G_1 is an HNN-extension with base group $(F(W_K) * A_1 * B_1)/N(\mathcal{E}) \leq G_1$. Moreover, $\widehat{\psi} : (F(W_K) * A_1 * B_1)/N(\mathcal{E}) \to K$ can be lifted to a morphism

$$\widehat{\psi} : G_1 \to G_0 = \langle K, t \mid a^t = \varphi_1(a) \ (a \in A_1) \rangle.$$

The idea for the construction of G_1 is to abstract as far as possible from the concrete structure of the original base group K. 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.

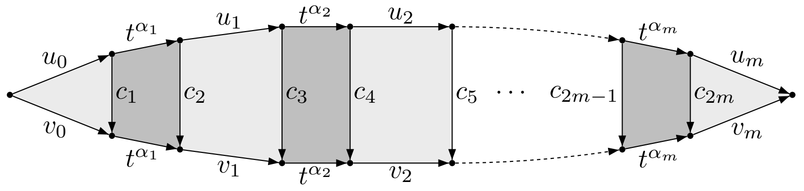
Note that since $\mathrm{val}(\mathbb{A}), \mathrm{val}(\mathbb{B}) \in \mathrm{Red}(K, \varphi_1)$, we have $\mathrm{val}(\mathbb{A}_t), \mathrm{val}(\mathbb{B}_t) \in \mathrm{Red}((F(W_K) * A_1 * B_1)/N(\mathcal{E}), \varphi_1)$: Consider for instance a factor $t^{-1} X_i t$ of $\mathrm{val}(\mathbb{A}_t)$ from (11). If $X_i = a$ in $(F(W_K) * A_1 * B_1)/N(\mathcal{E})$ for some $a \in A_1$, then after applying $\widehat{\psi}$ we have $\mathrm{val}(X_i) = u_i = a$ in K. Hence, $\mathrm{val}(\mathbb{A})$ from (9) would not be reduced.

Lemma 3.6. *The following are equivalent:*

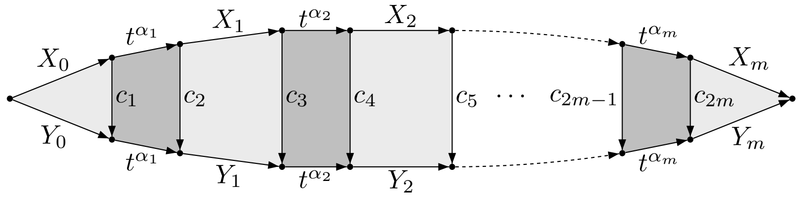
(a) $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ *in* G_0 *from (7).*
(b) $\mathrm{val}(\mathbb{A}_t) = \mathrm{val}(\mathbb{B}_t)$ *in* G_1

Proof. For $(b) \Rightarrow (a)$ assume that $\mathrm{val}(\mathbb{A}_t) = \mathrm{val}(\mathbb{B}_t)$ in G_1. We obtain in G_0: $\mathrm{val}(\mathbb{A}) = \widehat{\psi}(\mathrm{val}(\mathbb{A}_t)) = \widehat{\psi}(\mathrm{val}(\mathbb{B}_t)) = \mathrm{val}(\mathbb{B})$.

For $(a) \Rightarrow (b)$ assume that $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ in the group G_0. Since $\mathrm{val}(\mathbb{A})$ and $\mathrm{val}(\mathbb{B})$ are reduced and $\pi_t(\mathrm{val}(\mathbb{A})) = \pi_t(\mathrm{val}(\mathbb{B}))$, we obtain a Van Kampen diagram of the form:



In this diagram, we can replace every light-shaded face, representing the K-relation $u_i c_{2i+1} = c_{2i} v_i$, by a face representing the valid $\mathcal{E}$-relation $X_i c_{2i+1} = c_{2i} Y_i$, see (12). We obtain the following Van Kampen diagram, which shows that $\mathrm{val}(\mathbb{A}_t) = \mathrm{val}(\mathbb{B}_t)$ in G_1:



(★)

□

By Lemma 3.6, it remains to check, whether $\mathrm{val}(\mathbb{A}_t) = \mathrm{val}(\mathbb{B}_t)$ in the HNN-extension G_1, where $\mathrm{val}(\mathbb{A}_t)$ and $\mathrm{val}(\mathbb{B}_t)$ are both reduced.

3.4 Eliminating B_1 and t

By using the identities $b = t^{-1}\varphi_1^{-1}(b)t$ $(b \in B_1 \setminus \{1\})$ as Tietze transformations we can eliminate in the group G_1 the generators from $B_1 \setminus \{1\}$. After this transformation, we may have apart from relations of the form

$$Z_1 a_1 = a_2 Z_2 \text{ with } a_1, a_2 \in A_1 \tag{13}$$

also defining relations of the forms

$$\begin{aligned} Z_1 t^{-1} a_1 t &= a_2 Z_2 \\ Z_1 a_1 &= t^{-1} a_2 t Z_2 \\ Z_1 t^{-1} a_1 t &= t^{-1} a_2 t Z_2, \end{aligned}$$

where $a_1, a_2 \in A_1$. We can replace these relations by relations of the following types

$$Z_1 t^{-1} a_1 = a_2 Z_2 t^{-1} \tag{14}$$
$$t Z_1 a_1 = a_2 t Z_2 \tag{15}$$
$$t Z_1 t^{-1} a_1 = a_2 t Z_2 t^{-1} \tag{16}$$

and end up with the isomorphic group

$$G_2 = \langle F(W_K) * A, t \mid (13) - (16) \rangle.$$

Let us now introduce for every $Z \in W_K$ the new generators

$$[Zt^{-1}], [tZ], [tZt^{-1}]$$

together with the defining relations

$$[Zt^{-1}] = Zt^{-1},\ [tZ] = tZ,\ [tZt^{-1}] = tZt^{-1}. \tag{17}$$

This allows to replace the defining relations (14)–(16) by

$$[Z_1t^{-1}]a_1 = a_2[Z_2t^{-1}] \tag{18}$$

$$[tZ_1]a_1 = a_2[tZ_2] \tag{19}$$

$$[tZ_1t^{-1}]a_1 = a_2[tZ_2t^{-1}] \tag{20}$$

leading to the group

$$G_3 = \langle F(\{Z, [Zt^{-1}], [tZ], [tZt^{-1}] | Z \in W_K\}) * A_1, t \mid (13), (17) - (20)\rangle. \tag{21}$$

Finally, we can eliminate t and t^{-1} by replacing (17) by

$$[tZ] = [Zt^{-1}]^{-1}Z^2,\ [tZt^{-1}] = [tZ]Z^{-1}[Zt^{-1}]. \tag{22}$$

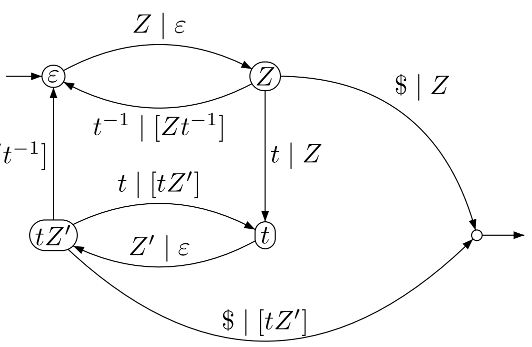
Doing this replacement we end up with the group

$$G_4 = \langle F(\{Z, [Zt^{-1}], [tZ], [tZt^{-1}] \mid Z \in W_K\}) * A_1 \mid (13),\ (18)\text{-}(20),\ (22)\rangle. \tag{23}$$

Since each transformation from G_1 to G_4 is a Tietze transformation, G_1 is isomorphic to G_4. We now want to rewrite the SLPs $\mathbb{A}_t$ and $\mathbb{B}_t$ into new SLPs over the generators of G_4. For this, we can define a deterministic rational transducer T that reads a word $X_0t^{\alpha_1}X_1t^{\alpha_2}X_2 \cdots t^{\alpha_m}X_m$ from the input tape and

– replaces every occurrence of a factor tX_i with $\alpha_{i+1} \neq -1$ by the symbol $[tX_i]$,
– replaces every occurrence of a factor X_it^{-1} with $\alpha_i \neq 1$ by the symbol $[X_it^{-1}]$, and finally
– replaces every occurrence of a factor tX_it^{-1} by the symbol $[tX_it^{-1}]$.

The state set of the transducer T is $\{\varepsilon, t\} \cup \{Z, tZ \mid Z \in W_K\}$ and the transitions are the following (for all $Z, Z' \in W_k$), where $\$$ is an end marker:


$$\begin{tikzcd}
{} \arrow[r] & \varepsilon \arrow[r, bend left, "Z \mid \varepsilon"] & Z \arrow[l, bend left, "t^{-1} \mid {[Zt^{-1}]}"] \arrow[d, "t \mid Z"] \arrow[rd, bend left, "\$ \mid Z"] & & \\
& tZ' \arrow[u, "t^{-1} \mid {[tZ't^{-1}]}"] \arrow[r, bend left, "t \mid {[tZ']}"] \arrow[rr, bend right, "\$ \mid {[tZ']}"'] & t \arrow[l, bend left, "Z' \mid \varepsilon"] & \circ \arrow[r] & {}
\end{tikzcd}$$

By Lemma 2.1 we can construct in polynomial time SLPs that generate the strings $[\![T]\!](\mathrm{val}(\mathbb{A}_t)\$)$ and $[\![T]\!](\mathrm{val}(\mathbb{B}_t)\$)$.

Let G_5 be the group that is obtained by removing the relations (22) from the presentation of G_4 in (23), i.e.,

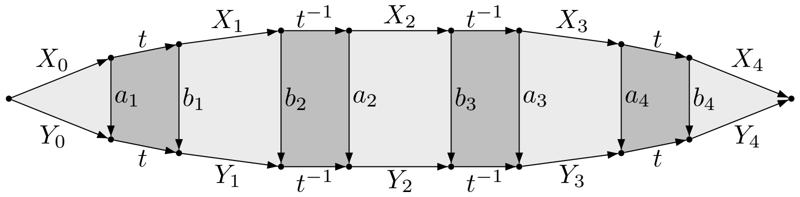
$$G_5 = \langle F(\{Z, [Zt^{-1}], [tZ], [tZt^{-1}] \mid Z \in W_K\}) * A_1 \mid (13), (18)\text{–}(20)\rangle. \qquad (24)$$

Lemma 3.7. *The following are equivalent:*

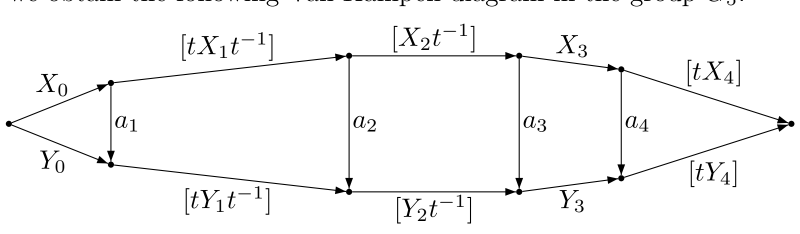
(a) $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ *in* G_0
(b) $\mathrm{val}(\mathbb{A}_t) = \mathrm{val}(\mathbb{B}_t)$ *in* G_1
(c) $[\![T]\!](\mathrm{val}(\mathbb{A}_t)\$) = [\![T]\!](\mathrm{val}(\mathbb{B}_t)\$)$ *in* G_4
(d) $[\![T]\!](\mathrm{val}(\mathbb{A}_t)\$) = [\![T]\!](\mathrm{val}(\mathbb{B}_t)\$)$ *in* G_5

Proof. The equivalence of (a) and (b) was stated in Lemma 3.6. The equivalence of (b) and (c) is clear since G_1 and G_4 are isomorphic and the transducer T rewrites a string over the generators G_1 into a string over the generators of G_4. Moreover, (d) implies (c) because we omit one type of relations, namely (22), when going from G_5 to G_4. It remains to prove that (a) implies (d). If $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ in G_0, then, as argued in the proof of Lemma 3.6, we obtain a Van Kampen diagram of the form (★) in the group G_1. The boundary of every light-shaded face is labeled with a relation from $\mathcal{E}$. We obtain a Van Kampen diagram for $[\![T]\!](\mathrm{val}(\mathbb{A}_t)\$) = [\![T]\!](\mathrm{val}(\mathbb{B}_t)\$)$ in G_5, basically by removing all vertical edges that connect (i) target nodes of t-labeled edges or (ii) source nodes of t^{-1}-labeled edges (there are B_1-labeled edges in (★)), see the following example. □

Example 3.8. Let us give an example of the transformation from a diagram of the form (★) into a Van Kampen diagram for the group G_5. Assume that the diagram in G_1 is:



Then we obtain the following Van Kampen diagram in the group G_5:



Only the relations (13) and (18)–(20) are used in this diagram.

For the further considerations, we denote the SLPs for the strings $[\![T]\!](\mathrm{val}(\mathbb{A}_t)\$)$ and $[\![T]\!](\mathrm{val}(\mathbb{B}_t)\$)$ again with $\mathbb{A}$ and $\mathbb{B}$, respectively. It remains to check whether $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$ in G_5. Let

$$\mathcal{Z} = \{Z, [Zt^{-1}], [tZ], [tZt^{-1}] \mid Z \in W_K\}$$

and let us redefine the set of defining relations $\mathcal{E}$ as the set of all defining relations of the form (13), (18)–(20). Thus,

$$G_5 = \langle F(\mathcal{Z}) * A_1 \mid \mathcal{E}\rangle,$$

where every defining relation in $\mathcal{E}$ is of the form $Z_1 a_1 = a_2 Z_2$ for $Z_1, Z_2 \in \mathcal{Z}$ and $a_1, a_2 \in A_1$.

3.5 Transforming $\langle F(\mathcal{Z}) * A_1 \mid \mathcal{E}\rangle$ into an HNN-extension

By further Tietze transformations we will show that G_5 is actually an HNN-extension with base group A_1 and associated subgroups A_1 and A_1. This will prove Lemma 3.5. To this end, let us take a relation $Z_1 a_1 = a_2 Z_2$ with $Z_1 \neq Z_2$. We can eliminate Z_2 by replacing it with $a_2^{-1} Z_1 a_1$. Subwords of the form aa' with $a, a' \in A_1$ that arise after this Tietze transformation can of course be multiplied out in the finite group A_1. We carry out the same replacement $Z_2 \mapsto a_2^{-1} Z_1 a_1$ also in the SLPs $\mathbb{A}$ and $\mathbb{B}$ which increases the size only by an additive constant and repeat these steps. After polynomially many Tietze transformations we arrive at a presentation, where all defining relations are of the form $Z = a_1 Z a_2$, i.e. $a_2 = Z^{-1} a_1^{-1} Z$. Let us write the resulting presentation as

$$G_6 = \langle A_1, Z_1, \ldots, Z_m \mid Z_i^{-1} a Z_i = \psi_i(a) \ (1 \leq i \leq m, a \in \mathrm{dom}(\psi_i))\rangle.$$

Note that every mapping ψ_i is a partial automorphism on A_1 since it results from the conjugation by some element in our initial group. Hence, we obtained an HNN-extension over A_1.

We can now finish the proof of Lemma 3.5, which states that the problem $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$ is polynomial time Turing-reducible to the problems $\mathrm{RCWP}(H, A, B, \varphi_2, \ldots, \varphi_k)$ and $\mathrm{RUCWP}(A_1, A_1, A_1)$. Using oracle access to $\mathrm{RCWP}(H, A, B, \varphi_2, \ldots, \varphi_k)$ (which was necessary for computing the set of defining relations $\mathcal{E}$ from (12)), we have computed in polynomial time from a given $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$-instance an $\mathrm{UCWP}(A_1, A_1, A_1)$-instance, which is a positive instance if and only if the original $\mathrm{RCWP}(H, A, B, \varphi_1, \ldots, \varphi_k)$-instance is positive. A final application of Lemma 3.2 allows to reduce $\mathrm{UCWP}(A_1, A_1, A_1)$ to $\mathrm{RUCWP}(A_1, A_1, A_1)$. This finishes the proof of Lemma 3.5.

3.6 Finishing the proof of Theorem 3.1

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 $\mathrm{RCWP}(A_1, A_1, A_1, \psi_1, \dots, \psi_k)$, where $\psi_1, \dots, \psi_k : A_1 \to A_1$ and $k \le \delta$ (we even have $k \le 2|A_1|! \cdot 2^{|A_1|} \le 2|A|! \cdot 2^{|A|} = \delta$).

We now separate the (constantly many) stable letters $t_1, \dots, t_k$ that occur in the $\mathrm{RCWP}(A_1, A_1, A_1, \psi_1, \dots, \psi_k)$-instance into two sets: $\{t_1, \dots, t_k\} = S_1 \cup S_2$ where $S_1 = \{t_i \mid \mathrm{dom}(\psi_i) = A_1\}$ and $S_2 = \{t_1, \dots, t_k\} \setminus S_1$. W.l.o.g. assume that $S_2 = \{t_1, \dots, t_\ell\}$. Then we can write our HNN-extension G_6 as

$$G_6 = \langle H', t_1, \dots, t_\ell \mid a^{t_i} = \psi_i(a) \ (1 \le i \le \ell, a \in \mathrm{dom}(\psi_i)) \rangle, \tag{25}$$

where

$$H' = \langle A_1, t_{\ell+1}, \dots, t_k \mid a^{t_i} = \psi_i(a) \ (\ell + 1 \le i \le k, a \in A_1) \rangle.$$

Note that $|\mathrm{dom}(\psi_i)| < |A_1|$ for every $1 \le i \le \ell$ and that $A_1 = \mathrm{dom}(\psi_i)$ for every $\ell+1 \le i \le k$. By Lemma 2.5, $\mathrm{CWP}(H')$ can be solved in polynomial time; H' is in fact the semidirect product $A_1 \rtimes_\varphi F(t_{\ell+1}, \dots, t_k)$, where $\varphi : F(t_{\ell+1}, \dots, t_k) \to \mathrm{Aut}(A_1)$ is defined by $\varphi(t_i) = \psi_i$. Recall also that at the end of Section 3.2, A_1 was chosen to be of maximal cardinality among the domains of all partial isomorphisms $\varphi_1, \dots, \varphi_k$. The following proposition summarizes what we have shown so far:

Proposition 3.9. *Let $\varphi_1, \dots, \varphi_k : A \to B$ be partial isomorphisms, where $k \le \delta$, $A_1 = \mathrm{dom}(\varphi_1)$, and w.l.o.g $|A_1| \ge |\mathrm{dom}(\varphi_i)|$ for $1 \le i \le k$. From an instance $(\mathbb{A}, \mathbb{B})$ of the problem $\mathrm{RCWP}(H, A, B, \varphi_1, \dots, \varphi_k)$ we can compute in polynomial time with oracle access to the problem $\mathrm{RCWP}(H, A, B, \varphi_2, \dots, \varphi_k)$*

(1) a semidirect product $A_1 \rtimes_\varphi F$, where F is a free group of rank at most δ,
(2) partial automorphisms $\psi_1, \dots, \psi_\ell : A_1 \to A_1$ with $\ell \le \delta$ and $|\mathrm{dom}(\psi_i)| < |A_1|$ for all $1 \le i \le \ell$, and
(3) an $\mathrm{RCWP}(A_1 \rtimes_\varphi F, A_1, A_1, \psi_1, \dots, \psi_\ell)$-instance, which is positive if and only if the initial $\mathrm{RCWP}(H, A, B, \varphi_1, \dots, \varphi_k)$-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 2.5.

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 $\mathrm{RCWP}(H, A, B, \varphi_1, \dots, \varphi_k)$ (with $k \le \delta$) in polynomial time. For this we apply Proposition 3.9 repeatedly. We obtain a computation tree, where the root is labeled with an $\mathrm{RCWP}(H, A, B, \varphi_1, \dots, \varphi_k)$-instance and every other node is labeled with an instance of a problem $\mathrm{RCWP}(C \rtimes_\varphi F, C, C, \theta_1, \dots, \theta_p)$, where F is a free group of rank at most δ, C is a subgroup of our finite group A, and $p \le \delta$. 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 polynomially. □

4 Amalgamated Products

In this section we prove a transfer theorem for the compressed word problem for an amalgamated free product, where the amalgamated subgroups are finite. We will deduce this result from our transfer theorem for HNN-extensions.

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 [15, 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

$$\Phi(x) \ = \ \begin{cases} t^{-1}xt & \text{if } x \in H_1 \\ x & \text{if } x \in H_2. \end{cases}$$

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

$$\begin{aligned} \mathrm{val}(\mathbb{A}) = 1 \text{ in } G &\iff \Phi(\mathrm{val}(\mathbb{A})) = 1 \text{ in } \Phi(G) \\ &\iff \mathrm{val}(\mathbb{B}) = 1 \text{ in } G'. \end{aligned}$$

By Theorem 3.1 and Theorem 2.4, $\mathrm{CWP}(G')$ 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.

One might also consider the compressed word problem for HNN-extensions of semigroups [8].

References

1. A. Bertoni, C. Choffrut, and R. Radicioni. Literal shuffle of compressed words. In *Proceeding of the 5th IFIP International Conference on Theoretical Computer Science (IFIP TCS 2008), Milano (Italy)*, pages 87–100. Springer, 2008.
2. W. W. Boone. The word problem. *Annals of Mathematics (2)*, 70:207–265, 1959.
3. M. Dehn. Über die Toplogie des dreidimensionalen Raumes. *Mathematische Annalen*, 69:137–168, 1910. In German.
4. W. Dicks and M. J. Dunwoody. *Groups Acting on Graphs.* Cambridge University Press, 1989.
5. L. Gasieniec, M. Karpinski, W. Plandowski, and W. Rytter. Efficient algorithms for Lempel-Ziv encoding (extended abstract). In R. G. Karlsson and A. Lingas, editors, *Proceedings of the 5th Scandinavian Workshop on Algorithm Theory (SWAT 1996), Reykjavík (Iceland)*, number 1097 in Lecture Notes in Computer Science, pages 392–403. Springer, 1996.
6. C. Hagenah. *Gleichungen mit regulären Randbedingungen über freien Gruppen.* PhD thesis, University of Stuttgart, Institut für Informatik, 2000.
7. G. Higman, B. H. Neumann, and H. Neumann. Embedding theorems for groups. *Journal of the London Mathematical Society. Second Series*, 24:247–254, 1949.
8. J. M. Howie. Embedding theorems for semigroups. *The Quarterly Journal of Mathematics. Oxford. Second Series*, 14:254–258, 1963.
9. I. Kapovich, A. Myasnikov, P. Schupp, and V. Shpilrain. Generic-case complexity, decision problems in group theory, and random walks. *Journal of Algebra*, 264(2):665–694, 2003.
10. Y. Lifshits. Processing compressed texts: A tractability border. In B. Ma and K. Zhang, editors, *Proceedings of the 18th Annual Symposium on Combinatorial Pattern Matching (CPM 2007), London (Canada)*, number 4580 in Lecture Notes in Computer Science. Springer, 2007.
11. M. Lohrey. Word problems and membership problems on compressed words. *SIAM Journal on Computing*, 35(5):1210 – 1240, 2006.
12. M. Lohrey and S. Schleimer. Efficient computation in groups via compression. In *Proceedings of Computer Science in Russia (CSR 2007), Ekatarinburg (Russia)*, number 4649 in Lecture Notes in Computer Science, pages 249–258. Springer, 2007.
13. M. Lohrey and G. Sénizergues. Theories of HNN-extensions and amalgamated products. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Proceedings of the 33st International Colloquium on Automata, Languages and Programming (ICALP 2006), Venice (Italy)*, number 4052 in Lecture Notes in Computer Science, pages 681–692. Springer, 2006.
14. M. Lohrey and G. Sénizergues. Rational subsets in HNN-extensions and amalgamated products. *International Journal of Algebra and Computation*, 18(1):111–163, 2008.
15. R. C. Lyndon and P. E. Schupp. *Combinatorial Group Theory.* Springer, 1977.
16. J. Macdonald. Compressed words and automorphisms in fully residually free groups. *International Journal of Algebra and Computation*, 20(3):343–355, 2010.
17. M. Miyazaki, A. Shinohara, and M. Takeda. An improved pattern matching algorithm for strings in terms of straight-line programs. In A. Apostolico and J. Hein, editors, *Proceedings of the 8th Annual Symposium on Combinatorial Pattern Matching (CPM 97), Aarhus (Denmark)*, Lecture Notes in Computer Science, pages 1–11. Springer, 1997.

18. A. Myasnikov, V. Shpilrain, and A. Ushakov. *Group-based Cryptography.* Birkhäuser, 2008.
19. P. S. Novikov. On the algorithmic unsolvability of the word problem in group theory. *American Mathematical Society, Translations, II. Series*, 9:1–122, 1958.
20. W. Plandowski. Testing equivalence of morphisms on context-free languages. In J. van Leeuwen, editor, *Second Annual European Symposium on Algorithms (ESA'94), Utrecht (The Netherlands)*, number 855 in Lecture Notes in Computer Science, pages 460–470. Springer, 1994.
21. W. Plandowski and W. Rytter. Application of Lempel-Ziv encodings to the solution of word equations. In *Proceedings of the 25th International Colloquium on Automata, Languages and Programming (ICALP 1998)*, number 1443 in Lecture Notes in Computer Science, pages 731–742. Springer, 1998.
22. W. Plandowski and W. Rytter. Complexity of language recognition problems for compressed words. In J. Karhumäki, H. A. Maurer, G. Paun, and G. Rozenberg, editors, *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 262–272. Springer, 1999.
23. S. Schleimer. Polynomial-time word problems. *Commentarii Mathematici Helvetici*, 83(4):741–765, 2008.
24. J. R. Stallings. *Group Theory and Three-Dimensional Manifolds.* Number 4 in Yale Mathematical Monographs. Yale University Press, 1971.