The Conjugacy Problem in Amalgamated Products I: Regular Elements and Black Holes

Borovik, Alexandre and Myasnikov,
Alexei G. and Remeslennikov, Vladimir V.

2006

MIMS EPrint: **2006.14**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

The Conjugacy Problem in Amalgamated Products I: Regular Elements and Black Holes

Alexandre V. Borovik, Alexei G. Myasnikov, and Vladimir N. Remeslennikov

Abstract. We discuss the time complexity of the word and conjugacy problems for free products $G = A \star_C B$ of groups A and B with amalgamation over a subgroup C. We stratify the set of elements of G with respect to the complexity of the word and conjugacy problems and show that for the generic stratum the conjugacy search problem is decidable under some reasonable assumptions about groups A, B, C. Moreover, the decision algorithm is fast on the generic stratum.

Contents

Introduction

Motivation. This is the first paper in a series of four written on the Word and Conjugacy Problems in amalgamated free products and HNN extensions. When this improves the presentation of the main concepts we mention several results from the subsequent papers.

Free products with amalgamation and HNN extensions are among the most studied classical constructions in algorithmic and combinatorial group theory. Methods developed for the study of the Word and Conjugacy Problems in these groups became the classical models much imitated in other areas of group theory. We refer to Magnus, Karrass, and Solitar book [**24**] for amalgamated free product techniques and to Lyndon and Schupp book [**23**] for HNN extensions.

The third author was supported by EPSRC grant GR/R29451 and by RFFI grants 02-01-00192 and 05-01-00057.

In 1971 Miller proved that the class of free products $A *_C B$ of free groups A and B with amalgamation over a finitely generated subgroup C contains specimens with algorithmically undecidable conjugacy problem [**25**]. This remarkable result shows that the conjugacy problem can be surprisingly difficult even in groups whose structure we seem to understand well. In few years more examples of HNN extensions with decidable word problem and undecidable conjugacy problem followed (see the book by Bokut and Kukin [**7**]). The striking undecidability results of this sort scared away any general research on the word and conjugacy problems in amalgamated free products and HNN extensions. The classical tools of amalgamated products have been abandoned and replaced by methods of hyperbolic groups [**6, 19, 27**], or automatic groups [**4, 14**], or relatively hyperbolic groups [**12, 29**].

In this and the subsequent paper [**9, 10, 11**] we make an attempt to rehabilitate the classical algorithmic techniques to deal with amalgams. Our approach treats both decidable and undecidable cases simultaneously. We show that, despite the common belief, the Word and Conjugacy Problems in amalgamated free products are generically easy and the classical algorithms are very fast on "most" or "typical" inputs. In fact, we analyze the computational complexity of even harder algorithmic problems which lately attracted much attention in cryptography (see [**3, 22, 31**], and surveys [**13, 33**]), the so-called *Normal Form Search Problem* and *Conjugacy Search Problem*. Our analysis is based on recent ideas of stratification and generic complexity [**8, 20**], which we briefly discuss below.

Stratification of the set of inputs. We start with a general formulation of our approach to algorithmic problems and then specify it to algorithmic problems in groups. We follow the book *Computational Complexity* of Papadimitriou [**30**] for our conventions on computational complexity.

Let M be a set with a fixed *size* function $size : M \to \mathbb{R}_{\geqslant 0}$ and $\mathcal{A}$ a partial algorithm with inputs from M. Denote by $\mathrm{Dom}\mathcal{A} \subseteq M$ the set of inputs on which $\mathcal{A}$ halts. For $w \in \mathrm{Dom}\mathcal{A}$ by $T_{\mathcal{A}}(w)$ we denote the number of steps required for the algorithm $\mathcal{A}$ to halt on the input w. If $f : \mathbb{R}_{\geqslant 0} \longrightarrow \mathbb{R}_{\geqslant 0}$ is a standard complexity time bound, say $f(x) = x^n$, or $f(x) = n^x$, $n \in \mathbb{N}$, then we say that $f(x)$ is a *worst case time upper bound* for $\mathcal{A}$ (with respect to the size function *size*) if there exists a constant $C \in \mathbb{R}$ such that for every $w \in M$

$$T_{\mathcal{A}}(w) \leqslant C f(size(w)) + C.$$

The set

$$M_f = \{w \in M \mid T_{\mathcal{A}}(w) \leqslant f(size(w))\}$$

is called the *f-stratum* of $\mathcal{A}$.

Assume now that the set M comes equipped with a (finitely additive) measure μ which takes values in $[0, 1]$. A subset $Q \subseteq M$ is called *generic* (*negligible*) if $\mu(Q) = 1$ ($\mu(Q) = 0$). A bound f is called a *generic upper bound* for $\mathcal{A}$ if the set M_f is generic with respect to μ. A generic upper bound f is *tight* if it is a minimal (with respect to the standard order $\preceq$ on the bounds) generic upper bound for $\mathcal{A}$ from a fixed list of upper bounds U. If not said otherwise, we always assume that U consists of the set of polynomial bounds $x^n, n \in \mathbb{N}$ and a simple exponential bound 2^x. It may happen that an algorithm $\mathcal{A}$ does not have a tight generic upper bound.

If f is a tight generic upper bound for $\mathcal{A}$ then the stratum M_f is called a *generic stratum*. Sometimes it is difficult to determine generic strata precisely, in

which case it is convenient to replace M_f by a "large enough" part of it. To this end we introduce the following notion. A subset $RP \subseteq \mathrm{Dom}(\mathcal{A})$ is called a *regular part* of M relative to $\mathcal{A}$ if RP is a generic subset of M such that $RP \subseteq M_f$ for some tight generic upper bound f for $\mathcal{A}$. One can view RP as the set of "algorithmically typical" inputs for $\mathcal{A}$ with respect to μ, so RP describes the most typical behavior of the algorithm on M. The compliment $BH = M \setminus RP$ is called a *black hole*. Clearly, the regular part PR and the black hole BH are defined up to a negligible set. In applications BH consists of elements w in M for which either the algorithm $\mathcal{A}$ does not work at all, or $T_{\mathcal{A}}(w)$ is not bounded by $f(size(w))$, or for some reason it is just not known whether w is in M_f or not. Finally, for a bound $h \in U$ we say that the regular part RP of $\mathcal{A}$ has at most h time complexity if M_h is generic. In particular, we say that RP is polynomial time if it has at most h time complexity for some polynomial h.

In what follows the measure μ appears either as the asymptotic density function on M with respect to the size function *size*, or the exponential distribution on M which comes from a corresponding random walk on M (we refer to [**8**] for details). To explain this we need a few definitions. Let

$$M = \bigcup_{i=0}^{\infty} M_i$$

be a partition of M with respect to the given size function $size : M \to \mathbb{R}$, thus

$$M_i = \{w \in M \mid size(w) = i\}.$$

In this case for a subset Q of M the fraction

$$\frac{\mu(Q \cap S_i)}{\mu(M_i)}$$

can be viewed as the probability of an element of M of size i to be in Q. The limit (if it exists)

$$\rho(Q) = \lim_{i \to \infty} \frac{\mu(Q \cap M_i)}{\mu(M_i)}$$

is called the *asymptotic density* of Q. The set Q is *generic* (*negligible*) with respect to ρ if $\rho(Q) = 1$ ($\rho(Q) = 0$), and Q is *strongly generic* if the convergence

$$\frac{\mu(Q \cap S_i)}{\mu(S_i)} \to 1$$

is exponentially fast when $i \to \infty$.

It is not hard to see that the collection $\mathcal{F}$ of all generic and negligible subsets of M is an algebra of subsets of M and the asymptotic density ρ is a measure on the space $(M, \mathcal{F})$.

Search problems in groups. The Word and Conjugacy Problems are two classical algorithmic problems introduced by M. Dehn in 1912. Since then much of the research in combinatorial group theory was related to these problems. We refer to surveys [**1, 25, 26, 32**] on algorithmic problems in groups.

Let G be a fixed group given by a finite presentation $G = \langle X; R \rangle$, and $M(X) = (X^{\pm 1})^*$ a free monoid over the alphabet $X^{\pm 1}$. Sometimes, slightly abusing notations, we identify words in $M(X)$ with their canonical images in the free group $F(X)$.

An algorithmic problem P over G can be described as a subset $D = D_P$ of a Cartesian power $M(X)^k$ of $M(X)$. The problem is *decidable* if there exists a *decision algorithm* $A = A_P$ which on a given input $w \in M(X)^k$ halts and outputs "Yes" if $w \in D_P$, otherwise it outputs "No". In applications it is often required to find a decision algorithm A_{Yes} for the "Yes" part which on an input $w \in D$ provides a "reasonable proof" that w is, indeed, in D. This leads to the *search* (or *witness*) variations of the algorithmic problems (see [**20, 28**] for a more detailed discussion of the Search Problems in groups).

Conjugacy Search Problem for G: *For a given pair $(u, v) \in M(X) \times M(X)$ verify if u and v are conjugate in G, and if they are find a conjugator.*

Search Word problem for finitely presented groups has several formulations which depend on the form of the witness. For example, if an input $w \in M(X)$ is equal to 1 in G then one may require to provide a presentation of w as a product of conjugates of relators from R as a witness (see [**28**]). However, in free products with amalgamation it is convenient to consider the following more general problem.

Let R be a set of reduced forms and $\mathcal{N}$ be a a fixed set of normal forms (viewed as words in $M(X)$) of elements from G.

Reduced Forms Search Problem for G and R: *For a given $w \in M(X)$ find its reduced form $\bar{w} \in R$.*

Let $\mathcal{N}$ be a fixed set of normal forms (viewed as words in $M(X)$) of elements from G, and $\bar{w}$ a representative of w.

Normal Forms Search Problem for G and $\mathcal{N}$: *For a given $w \in M(X)$ find its normal form $\bar{w} \in \mathcal{N}$.*

This new requirement for the search decision problems to provide a "proof", or a "witness", of the correct decision brings quite a few new algorithmic aspects, which were not studied in group theory (see discussion in [**28**]).

Results. We show below that in the free product $G = A *_C B$ of free groups A and B with amalgamation over a finitely generated subgroup C of infinite index in A and in B, the Canonical Form and the Conjugacy Search Problem is decidable for the set of *regular* elements in G. Moreover, we analyze the time complexity of these problems (modulo the corresponding algorithms in the factors A, B).

In Section 3.2 we study algorithmic properties of the standard rewriting Algorithm I for computing the normal forms of elements in amalgamated free products (as, basically, described in [**24**]). It turns out that Algorithm I has the following algorithmic properties (modulo the above mentioned algorithms in the factors):

- Algorithm 0 and I have exponential worst case time complexity in the length of the inputs w.

The results of the papers show that this upper bound for complexity is not actually reached on many groups, and, in some precise sense, the number of such groups is large.

The paper [**9**] contains a detailed quantitative analysis of the complexity of the Conjugacy Search Problem in amalgamated products of free groups.

We note also an important special case of free products with amalgamation $G = A *_C B$ where the previous result can be improved. Let A, B, C be free groups of finite rank and assume that C has infinite index in both A and B. We prove, that, in that case, the following statements hold.

- The regular part RP of Algorithm I has linear time complexity on a generic subset [**9**].

This result plays the key role in the estimates of the generic complexity of the Conjugacy Search Problem in G.

Notice that [**9**] also contains

- natural procedures for generating random reduced, normal and cyclically reduced normal forms of elements of G.
- We also give explicit formulae for the probabilities of generation of the given random form.

Next important ingredient of [**9**] is

- the extended subgroup graph Γ_C^* defined for a finitely generated subgroup C of a free group F.

This graph allows us to construct, by means of a finite state automaton, all Schreier systems of representatives for the subgroup C. Furthermore, for a fixed Schreier system S for the subgroup C in F we define representatives with special properties: stable and regular. A representative $s \in S$ is called *stable*, if, for every $c \in C$, the element sc belongs to S. A representative $s \in S$ is called *regular* if it is not contained in the generalized normalizer $N_C^*(F)$ of the subgroup C in F (the definition is given later in the paper). A normal form of an element $g \in G$ is called *stable* if contains at least one stable representative. A normal form of an element is called *regular* if it contains at least one regular representative.

- Let S be a Schreier system of representatives of F with respect to C. Then if C is a subgroup of infinite index in F then the sets of stable and regular representatives are strictly generic in S [**9**].
- If $G = A *_C B$, where A, B, C are free groups of finite rank and C has infinite index in both A and B, then, under some minor additional assumption about C (details are in [**9**]), we show that
  - the set of all stable reduced forms is strictly generic with respect to the set of all reduced forms;
  - the set of all regular normal forms is strictly generic with respect to all normal forms.

In Section 4 we study the time complexity of the Search Conjugacy Problem in G. More precisely, we study the time complexity of the standard decision algorithm $\mathcal{B}$ for the Conjugacy Problem in amalgamated free products (following the description in [**24**]). The main result of this paper shows that the algorithm $\mathcal{B}$ solves the Search Conjugacy Problem in G for all regular elements, and their conjugates. To describe regular elements in G we need the following definitions. The *generalized normalizer* $N_H^*(K)$ of a subgroup K of a group H is a set of all elements $g \in H$ such that $K \cap K^g \neq 1$ [**5**]. $N_H^*(K)$ "measures malnormality" of K in H. For an element $g \in H$ define

$$Z_g(K) = \{h \in K \mid h^g \in K\} = K^{g^{-1}} \cap K$$

and put

$$Z_H(K) = \bigcup_{g \in N_H^*(K) \setminus K} Z_g(K) = \bigcup_{g \in N_H^*(K)} Z_g(K).$$

Now in the group $G = A *_C B$ the set

$$BH = (N^*{}_G(C) \setminus C) \cup Z_G(C)$$

is called a *Black Hole*, and its complement $RP = G \setminus BH$ is called the *Regular Part* of $\mathcal{A}$.

It turns out that $\mathcal{B}$ has the following algorithmic properties (modulo the corresponding algorithms in the factors):

- $\mathcal{B}$ decides whether or not a given element from G is conjugated to a given regular element of G.
- the regular part RP is generic in G ([**9**]);
- $\mathcal{B}$ works fast on elements from RP.
- RP is a decidable subset of G.

In [**10**] these results (with the exception of the genericity of the regular part RP in G) are transferred to HNN extensions of free groups. In [**11**] we further specilise them for the class of Miller's groups.

Starting with a presentation for a finitely presented group H, Miller [**26**] constructed a generalized HNN-extension $G(H)$; he then showed that the Conjugacy Problem in $G(H)$ is decidable if and only if the Word Problem is decidable in H. Varying the group H, one can easily construct infinitely many groups $G(H)$ with decidable word problem and undecidable conjugacy problem.

In [**10**], working under some mild assumptions about the groups involved in a given HNN-extension G , we stratify G into two parts with respect to the "hardness" of the conjugacy problem:

- a *Regular Part* RP, consisting of so-called *regular elements* for which the conjugacy problem is decidable by standard algorithms. We show that the regular part RP has very good algorithmic properties:
  - the standard algorithms are very fast on regular elements;
  - if an element is a conjugate of a given regular element then the algorithms quickly provide a conjugator, so the Search Conjugacy problem is also decidable for regular elements;
  - the set RP is *generic* in G, that is, it is very "big" in some particular sense explained in the previous paper [**9**];
- the *Black Hole* BH (the complement of the set of regular elements) which consists of elements in G for which either the standard algorithms do not work at all, or they require a considerable modification, or it is not clear yet whether these algorithms work or not.

This general technique for solving the conjugacy problem in HNN-extensions does not work in Miller's groups, for in this case the Black Hole (BH) of the conjugacy problem algorithm coincides with the whole group. We therefore have to weaken the definition of regular elements to so called weakly regular elements and change the notion of the Black Hole to the notion of a Strongly Black Hole. It is proven that the Conjugacy Search Problem for elements that do not lie in the Strongly Black Hole (SBH) is decidable in cubic time. We give an explicit description of the size of SBH for Miller's groups and prove that SBH is a strongly sparse set.

This is the first example of a non-trivial solution of the Stratified Conjugacy Problem in finitely presented groups.

1. Preliminaries

1.1. Amalgamated products. In this section we briefly discuss definitions and some known facts about free products with amalgamation. We refer to [**24**] for details.

Let A, B, C be groups and $\phi : C \longrightarrow A$ and $\psi : C \longrightarrow B$ be monomorphisms. Then one can define a group

$$G = A *_C B,$$

called the *amalgamated product of A and B over C* (the monomorphisms ϕ, ψ are suppressed from notation). If A and B are given by presentations $A = \langle X \mid R_A = 1\rangle$, $B = \langle Y \mid R_B = 1\rangle$, and a generating set Z is given for the group C, then the group G has presentation

$$G = \langle X \cup Y \mid R_A = 1, R_B = 1, z^\phi = z^\psi (z \in Z)\rangle. \tag{1}$$

Notice that if the groups A and B are finitely presented and C is finitely generated then the group G is finitely presented. If we denote

$$z^\phi = u_z(x), \quad z^\psi = v_z(y)$$

then G has presentation

$$G = \langle X \cup Y \mid R_A = 1, R_B = 1, u_z(x) = v_z(y),\ (z \in Z)\rangle.$$

The groups A and B a called *factors* of the amalgamated product $G = A *_C B$, they are isomorphic to the subgroups in G generated respectively by X and Y. We identify A and B with these subgroups via the identical maps $x \to x$, $y \to y$ $(x \in X,\ y \in Y)$. Notice also that

$$C = A \cap B = \langle u_z\rangle_{z\in Z} = \langle v_z\rangle_{z\in Z} \leqslant G.$$

1.2. Canonical forms of elements. Let $G = A *_C B$ be an amalgamated product of groups as in (1). Denote by S and T fixed systems of right coset representatives of C in A and B. Throughout this paper we assume that the representative of C is the identity element 1.

The following notation will be in use throughout the paper. For an element $g \in (A \cup B) \setminus C$ we define $F(g) = A$ if $g \in A$ and $F(g) = B$ if $g \in B$.

THEOREM 1.1. [**24**, Theorem 4.1] *An arbitrary element g in $G = A *_C B$ can be uniquely written in the* canonical *normal form with respect to S and T*

$$g = cg_1g_2 \cdots g_n, \tag{2}$$

where $c \in C$, $g_i \in T \cup S \setminus \{1\}$, and $F(g_i) \neq F(g_{i+1})$, $i = 1, \ldots, n$, $n \geqslant 0$.

COROLLARY 1.2. *Every element $g \in A *_C B$ can be written in a* reduced form

$$g = cg_1g_2 \cdots g_n \tag{3}$$

where $c \in C, g_i \in (A \cup B) \setminus C$, and $F(g_i) \neq F(g_{i+1})$, $i = 1, \ldots, n$, $n \geqslant 0$. This form may not be unique, but the number n is uniquely determined by g. Moreover, $g = 1$ if and only if $n = 0$ and $c = 1$.

Let $g \in A *_C B$ and $g = cg_1g_2 \cdots g_n$ be a reduced form of g. Then the number n is called the *length* of g and it is denoted by $l(g)$. Observe, that $l(g) = 0 \Longleftrightarrow g \in C$.

DEFINITION 1.3. Let $g \in A *_C B$. A reduced form $g = cg_1g_2 \cdots g_n$ is called *cyclically reduced* if one of the following conditions is satisfied:

(a) $n = 0$;
(b) $n = 1$ and g is not a conjugate of an element in C;

(c) $n \geqslant 2$ and $F(g_1) \neq F(g_n)$.

Notice that our definition of cyclically reduced forms is slightly different from the standard one (see, for example, [**24**]). Usually, the condition (b) is not required, but the difference is purely technical, and it is convenient to have (b) when dealing with conjugacy problems. Observe also, that if one of the reduced forms of g is cyclically reduced then all of them are cyclically reduced. In this event, g is called *cyclically reduced* element.

LEMMA 1.4 ([**24**]). *Let $g \in A *_C B$. Then g is a conjugate of some element g_0 in a cyclically reduced canonical form. This element g_0 is not uniquely defined, but its length $l(g_0)$ is uniquely determined by g.*

The canonical form of g_0 is called *a cyclically reduced canonical form* of g. The uniquely determined number $l(g_0)$ is called the *cyclic length* of g and it is denoted by $l_0(g)$. Observe that

$$\begin{aligned} l_0(g) = 0 &\iff \text{some conjugate of } g \text{ is in } C, \\ l_0(g) = 1 &\iff \text{some conjugate of } g \text{ is in } (A \cup B) \setminus C. \end{aligned}$$

1.3. Conjugacy criterion.

THEOREM 1.5. [**24**, Theorem 4.6] *Let $G = A *_C B$ be an amalgamated product, and let g be a cyclically reduced element in G.*

(i) *If $l_0(g) = 0$, i.e., $g \in C$, and g is conjugate to an element $c \in C$ then there exists a sequence of elements $c = c_0, c_1, \dots, c_t = g$, where $c_i \in C$ and adjacent elements c_i and c_{i+1}, $i = 0, \dots, t-1$, are conjugate in A or in B.*

(ii) *If $l_0(g) = 1$, i.e., $g \in A \cup B \setminus C$, and g' is a cyclically reduced element which is a conjugate of g in G then $l(g') = 1$, $F(g) = F(g')$ and g and g' are conjugate in $F(g)$.*

(iii) *Let $l_0(g) = r \geqslant 2$ and $g = g_1 \cdots g_r$ be a cyclically reduced form of g. Assume that g is conjugate to a cyclically reduced element $h = h_1 \cdots h_s$ in G. Then $r = s$ and h can be obtained from g by a cyclic permutation of the elements $g_1, \dots, g_r$ followed by a conjugation by an element from C.*

1.4. Malnormal subgroups. Recall, that a subgroup H of a group G is called *malnormal* in G if $H \cap H^g = 1$ for all $g \in G \setminus H$.

It follows immediately from the conjugacy criterion (Theorem 1.5) that free factors A and B are malnormal in the free product $A * B$. It is known that maximal abelian subgroups (= proper centralizers) are malnormal in torsion-free hyperbolic groups, in particular in free groups. We refer to [**17**] for results on malnormality of maximal abelian groups in free products with amalgamation and HNN extensions.

DEFINITION 1.6. Let G be a group and H be a subgroup of G. The *generalized normalizer* $N^*_G(H)$ is a set of all elements $g \in G$ such that $H \cap H^g \neq 1$.

Notice that, $N_G(H) \subseteq N^*_G(H)$, and, in general, $N^*_G(H)$ is not a subgroup. It is obvious that if $g \in N^*_G(H)$ then $N^*_G(H)$ contains the whole double coset HgH. A set of representatives $\{g_i \mid i \in I\}$ of double cosets of H is called a *double transversal*

of H in $N_G^*(H)$, in this event

$$N_G^*(H) = \bigcup_{i \in I} Hg_iH$$

If H is a finitely generated subgroup of a free group G then H has a finite double transversal in $N_G^*(H)$, moreover such a transversal can be found algorithmically [**5**]. A more convenient algorithm (in terms of subgroup graphs) can be found in [**19**].

For an element $g \in G$ define

$$Z_g(H) = \{h \in H \mid h^g \in H\} = H^{g^{-1}} \cap H$$

and put

$$Z_G(H) = \bigcup_{g \in N_G^*(H) \setminus H} Z_g(H) = \bigcup_{g \in N_G^*(H)} Z_g(H).$$

Even though $Z_g(H)$ is a subgroup of G for every $g \in G$, the set $Z_G(H)$ may not be a subgroup. Observe, that for any $u, v \in H$

$$Z_{ugv} = {Z_g}^{u^{-1}}.$$

Hence if T is a double transversal of H in $N_G^*(H)$ then $Z_G(H)$ is union of conjugacy classes:

$$Z_G(H) = \bigcup_{h \in H, t \in T} Z_t(H)^h$$

In particular, if the transversal T is finite then $Z_G(H)$ is union of finitely many conjugacy classes of subgroups $Z_t(H)$.

Definition 1.7. Let G be a group equipped with a map $L : G \to \mathbb{N}$ and H be a subgroup of G. For an element $g \in G$ define $L(g_H)$ as the minimal value of L on the double coset HgH. Then the *malnormality degree* $md(H)$ of H in G with respect to L is the smallest cardinal r such that $H \cap H^g = 1$ for all $g \in G$ with $L_H(g) \geqslant r$.

For example, the malnormality degree of subgroups can be defined in free groups, free products with amalgamation, and HNN extensions of groups with respect to the canonical length functions. In the sequel we always assume that for $H \leqslant A *_C B$ the degree $md(H)$ is viewed with respect to the canonical length function $l : A *_C B \to \mathbb{N}$.

Obviously, if a subgroup H has a finite double transversal in $N_G^*(H)$ then $md(H)$ is finite.

Lemma 1.8. *Let $G = A *_C B$ and $D \leqslant C$. Then*

(i) *If C is malnormal in A and B then $md(D) = 1$.*
(ii) *If C is malnormal in one of the groups A and B then $md(D) \leqslant 2$.*

Proof. Let $g = g_1 \cdots g_n$ be a reduced form of an element $g \in G$. Suppose $l(g) \geqslant 1$, in particular, $g_n \notin C$. Suppose also that $c, c' \in C$. If

$$g_1 \cdots g_n c g_n^{-1} \cdots g_1^{-1} = c'$$

then $g_n c g_n^{-1} \in C$. Assume that C is malnormal in both A and B. This implies that $g_n \in C$ -contradiction. Then $n = 0$ and therefore $md(D) = 1$.

If C is malnormal either in A or in B then similar argument shows that $md(D) \leqslant 2$. □

QUESTION 1.9. Let $G = A *_C B$ and H be a finitely generated subgroup of G. Assume that the malnormality degree $md_G(C)$ of C in G is finite, and H contains no elements of length $\leqslant md_G(C)$.

(a) Is it true that $md(H)$ is finite?
(b) Is it true that $N^*_G(H)$ is union of finitely many double cosets of H?

2. Algorithmic problems in groups

In this section we list some requirements on A, B, C which enable one to solve various problems in $A *_C B$ algorithmically.

Fix a group H given by a presentation given group $H = \langle X \mid R \rangle$. We discuss below several algorithmic problems for H. Many algorithmic problems for H come in three variations: specific, uniform, and search. For example, the *specific* membership problem in H is decidable for a given fixed finitely generated subgroup D of H if there exists an algorithm which for every word $w \in F(X)$ decides whether the element represented by w in H belongs to D or not. In this case an input to the algorithm is a word w in $F(X)$ and outputs are answers "yes" or "no". Decidability of the *uniform* membership problem for H requires an algorithm which would solve the specific membership problem for every finitely generated subgroup D of H. In this case, inputs come in pairs: a subgroup D and a word w, and outputs are "yes" or "no". Meanwhile, the *search* membership problem requires an algorithm which, for a given fixed subgroup D and a given element w, decides whether or not w belongs to D, and if it does, the algorithm finds a presentation of w as a product of the given generators of D. In this case answers are either "no", or "yes" with a word in the given generators of D. Search problems could also be uniform or specific. It is convenient to treat uniform and specific forms as particular cases of problems which are *uniform relative to a given class of objects* Φ. For example, the membership problem for a class of subgroups Φ of H solves the specific membership problem for every subgroup D from Φ. This relative approach is very natural, since there are groups in which the uniform version of a particular algorithmic problem is undecidable, but still there are interesting subclasses of objects Φ for which this problem is uniformly decidable. Moreover, even if the uniform version of the problem is decidable the class of all objects in the question can be partition into different subclasses with respect to different complexities of the decision algorithms.

Below we list some algorithmic problems for H in their uniform relative to a subclass search variation. These algorithmic problems involve different subsets of H (subgroups, cosets, double cosets, regular sets, recursive sets, etc.) given by some natural effective (constructive) descriptions. For example, finitely generated subgroups D are given by finite generating sets (which are given as words from $F(X)$), cosets wD are given as pairs (D, w), regular sets are given either by finite automata or by regular expressions, etc. Usually, we do not specify any particular descriptions of these subsets, unless it is required by complexity issues or by a particular algorithm.

Word Search Problem for a given subset of elements Φ ($\mathbf{WSP_\Phi}$): *Let Φ be a given subset of elements from H (given as words from $F(X)$). For a given $w \in \Phi$ decide whether $w = 1$ in H or not? If $w = 1$ then find a presentation of w as a product of conjugates of relators from R.*

We will see in Section 3.2 that there is a "very large" subset of elements from $G = A *_C B$ for which the standard algorithm for computing normal forms is "very fast".

Conjugacy Search Problem for a given set of pairs of elements Φ ($\mathbf{CSP_\Phi}$): *Let Φ be a given set of pairs of elements from H. For a given pair $(u, v) \in \Phi$ determine whether u is a conjugate of v in H or not, and if it is then find a conjugator.*

We will see in Section 4 that there is a large set of *regular* elements in $G = A*_C B$ for which the conjugacy problem is decidable, even though the standard conjugacy problem for G may be undecidable.

Conjugacy Membership Search Problem for a set of subgroups Φ ($\mathbf{CMSP_\Phi}$): *Let Φ be a set of finitely generated subgroups of H. For a given $D \in \Phi$ and a given $w \in F(X)$ determine whether w is a conjugate of an element from D, and if so, find such an element in D and a conjugator.*

Coset Representative Search Problem for a set of subgroups Φ ($\mathbf{CRSP_\Phi}$): *Let Φ be a set of finitely generated subgroups of H. For a given $D \in \Phi$ find a recursive set S of representatives of D in H and an algorithm A_S which for a given word $w \in F(X)$ finds a representative for Dw in S*

Observe that to solve $\mathbf{CRSP_\Phi}$ for a given $D \in \Phi$ it suffices to find the algorithm A_S, since $w \in S$ if and only if w is the output of A_S on the input w.

To formulate the next algorithmic problem we need the following definition. Let M be a subset of a group H. If $u, v \in H$ then the set uMv is called a shift of M. For a set $\mathcal{M}$ of subgroups of H denote by $\Phi(\mathcal{M}, H)$ the least set of subsets of H which contains $\mathcal{M}$ and is closed under shifts and intersections.

Cardinality Search Problem for $\Phi(C, H)$ ($\mathbf{CardSP_\Phi}$): *Let $\mathcal{M}$ be a set of finitely generated subgroups of H. Given a set $D \in \Phi(\mathcal{M}, H)$ decide whether D is empty, finite, or infinite and, if D is finite non-empty, list all elements of D.*

Malnormality Search Problem for a given set of subgroups Φ ($\mathbf{MalSP_\Phi}$): *Let Φ be a set of finitely generated subgroups of H. For a given $D \in \Phi$ find $N_H^*(D)$ and $Z_H(D)$.*

This problem, it seems, is a new type of algorithmic problems, which was not studied before. Therefore, we discuss it in more details. Usually (at least, in what follows), for a given $D \in \Phi$ the algorithm finds a *finite* double transversal S of D in $N_H^*(D)$, i.e., a finite set $S \subset H$ such that

$$N_H^*(D) = \bigcup_{s \in S} DsD.$$

So the subgroups from Φ have to have finite transversals. In this event, if the Membership Problem is decidable for double cosets DsD, when $s \in S$, then $N_H^*(D)$ is recursive. Moreover,

$$Z_H(D) = \bigcup_{d \in D, s \in S} Z_s(D)^d$$

Therefore, a given $w \in H$ belongs to $Z_H(D)$ if and only if $w \in D$ and some conjugate of w in D belongs to a subgroup from the finite set $Z_s(D)$, $s \in S$. Hence if the Conjugacy Membership Problem is decidable in D for subgroups $Z_s(D)$, $s \in S$ then the set $Z_H(D)$ is recursive and the problem $\mathbf{MalSP_\Phi}$ is decidable.

THEOREM 2.1. *Let H be a free group. Then all the problems above are decidable. Moreover, all these problems have decision algorithms of polynomial-time complexity.*

PROOF. Malnormality Search Problem for the class of all finitely generated subgroups of a free group was solved in [**5**]. A different proof (in terms of automata) could be extracted from [**19**]. □

For decidability of the other problems we refer to [**19**], even though these results have been proven much earlier, but it is easier to discuss complexity of algorithms from [**19**].

3. Computing canonical forms

3.1. Algorithm 0 for computation of reduced forms. In this Section we discuss the following algorithmic problems

Reduced Forms Problem. Let $G = A *_C B$ and suppose that Membership Search Problem (MSP) for C in A and in B is decidable. Give an algorithm which for a given $g \in F(X \cup Y)$ finds a reduced form of g.

We describe a decision algorithm for the problem above. Given a word $g \in F(X \cup Y)$ one can effectively present it as a product

$$g = g_1 \cdots g_k, \tag{4}$$

where $g_1, \ldots, g_k$ are reduced words in X or in Y and if g_i is a word in X then g_{i+1} is a word in Y and vice versa.

ALGORITHM 0: COMPUTING REDUCED FORMS.

INPUT: a word $g = g_1 \cdots g_k$ in the form (4).

STEP 1.:

Check if $g_i \in C$, $i = 1, \ldots, k$ or not. If none of the g_i's lies in C then (4) is reduced.

STEP 2. Let $g_i \in C$ and suppose that i is the minimal index with this property. If $i = 1$, rewrite g in the form $g = g_2' g_3 \cdots g_k$, where $g_2' = g_1 g_2 = c g_2$, $c \in C$, and go back to Step 1.

If $i > 1$ then rewrite g as follows $g = g_1 \cdots g_{i-2} g_{i+1}' g_{i+2} \cdots g_k$, where $g_{i+1}' = g_{i-1} c g_{i+1}$ in A or in B, and go to Step 1.

END OF ALGORITHM I

THEOREM 3.1. *Let $G = A *_C B$ and MSP is decidable for C in A and in B then Algorithm 0 finds the reduced form of g in at most quadratic time (is of at most quadratic worst case time complexity) modulo MSP.*

3.2. The standard Algorithm I to compute normal forms. We fix a free product with amalgamation $G = A *_C B$ given by the following presentation

$$G = \langle X \cup Y \mid R_A = 1, R_B = 1, u_z(x) = v_z(y), (z \in Z) \rangle.$$

In this section we discuss the following algorithmic problem.

Canonical Forms Search Problem: *Let $G = A *_C B$ and let S, T be recursive sets of representatives of A and B modulo C. Give an algorithm which for a given $g \in F(X \cup Y)$ finds the canonical form of g in G with respect to the sets S and T.*

Now we describe the standard known decision algorithm for the problem above (see, for example, [**25**]) provided we are given decision algorithms for the Coset Representatives Search Problem (**CRSP**) for the subgroup C in A and in B (relative to the sets S and T).

Given a word $g \in F(X \cup Y)$ one can effectively present it as a product

$$g = g_1 \cdots g_k, \tag{5}$$

where $g_1, \ldots, g_k$ are reduced words in X or in Y, and if g_i is a word in X, then g_{i+1} is a word in Y, and vice versa.

Modulo the algorithm for **CRSP** the process of computing the canonical form is the following.

Algorithm I: Computing Canonical Forms.

Input: a word $g = g_1 \cdots g_k$ in the form (5).

Step 0.:

- If g_k is a word in X, write it as $g_k = c_k u_k$ with $u_k \in S$ (using the Coset Representative Search Algorithm).
- If g_k is a word in Y, write it as $g_k = c_k u_k$ with $u_k \in T$ (using the Coset Representative Search Algorithm).

Comment. Notice that the end segment $c_k u_k$ of the word

$$g_1 \cdots g_k = g_1 \cdots g_{k-1} c_k u_k$$

now is written in the canonical form. So we may continue by induction.

Induction Step.: If g is represented in the form

$$g = g_1 \cdots g_i c_{i+1} u_{i+1} \cdots u_m$$

where $c_{i+1} u_{i+1} \cdots u_m$ is in the canonical form, do:

(a) Observe, that c_{i+1} is always given either as a word $c_{i+1}(v_1, \ldots, v_n)$ in generators v_j, or as a word $c_{i+1}(u_1, \ldots, u_n)$ in generators u_j (as the output of the **CRSP** algorithm). Now we rewrite c_{i+1} as follows:
   - If $g_i \in F(X)$ and c_{j+1} is given as a word $c_{i+1}(v_1, \ldots, v_n)$ in generators v_j then replace each v_j in $c_{i+1}(v_1, \ldots, v_n)$ by u_j; otherwise leave c_{i+1} unchanged (since it is already written as a word in X).
   - If $g_i \in F(Y)$ then rewrite c_{i+1} as a word in Y, replacing u_j by v_j if necessary (as above).

(b) Depending on whether g_i is a word in X or in Y, rewrite

$$g_i c_{i+1} = c_i u_i, \ \ c_i \in C, \ u_i \in S \text{ or } u_i \in T, \text{ correspondingly.}$$

(c) If both u_i and u_{i+1} are in S or both of them are in T, rewrite

$$u_i u_{i+1} = c' u_i' \text{ with } c' \in C \text{ and } u_i' \in S \text{ or } T, \text{ correspondingly,}$$

and change notation

$$c_i := c_i c', \ u_i := u_i', \ u_{i+1} := u_{i+2}, \ldots, u_{m-1} := u_m.$$

Output: The word

$$g = c_1 u_1 \cdots u_m$$

which is the canonical form of g relative to the set of representatives S and T.

End of Algorithm I

We summarize the discussion above as the following theorem

THEOREM 3.2. *Let $G = A *_C B$ and the* **CRSP** *is decidable in A and in B for the subgroup C. Then Algorithm I finds the canonical form of g for every given element $g \in G$.*

3.3. Complexity of Algorithm I. Now we discuss briefly time-complexity of Algorithm I. Recall that the *time function* $T_{\mathcal{A}}$ of an algorithm $\mathcal{A}$ is defined on an input g of $\mathcal{A}$ as the number of steps required by the algorithm $\mathcal{A}$ to halt on the input g.

The complexity of the time function T_I of the Algorithm I depends, firstly, on complexity of the time functions of decision algorithms for **CRSP** for A and B relative to C. Also, it depends on how the length of the words c_i grows during the rewriting process in the item (a) of the description of Algorithm I.

Complexity of the Coset Representative Search Algorithm depends on particular groups A, B, and C. For example, if A and B are free groups, then **CRSP** has linear time complexity for a fixed subgroup C (see, for example, [**19**]).

Estimating the complexity of the rewriting process (a) is more demanding, even in the case of amalgamated products of free groups. Recall, that in the rewriting process (a) we rewrite a word $c_{j+1}(u_1, \dots, u_n)$ into a word $c_{j+1}(v_1, \dots, v_n)$. Set

$$\lambda(u, v) = \frac{\max\{|u_1|, \dots, |u_n|\}}{\min\{|v_1|, \dots, |v_n|\}}$$

Then we have an upper bound estimate on the increase of the length

$$|c_{j+1}(v_1, \dots, v_n)| \leqslant \lambda(u, v) \cdot |c_{j+1}(u_1, \dots, u_n)|.$$

Similarly, in the case when we rewrite a word c_{j+1} given in the generators v_i into a word in generators u_i we have an estimate with the factor $\lambda(v, u)$. Therefore, if we denote

$$\lambda = \max\{\lambda(u, v), \lambda(v, u)\}$$

then at any rewriting step one has increase in length of at most by the factor λ.

Now suppose, for simplicity, that the length of c_j increases in the rewriting processes (b) and (c) at most by $M + |g_j|$ where M is a fixed constant (we make this assumption to focus on the process (a)). Under these assumptions

$$|c_j| \leqslant \lambda \cdot |c_{j+1}| + M + |g_j| \tag{6}$$

In particular, if the length of c_j does not increase at all in the rewriting processes b) and c) then in k steps we will have an exponential estimate

$$|c_1| \leqslant \lambda^{k-1} \cdot |c_k|$$

where $k = l(g)$. So if $\lambda > 1$ then we might have exponential growth of the length of the words c_i. The example below shows that this happens in the worst case scenario.

EXAMPLE 3.3. Let $A = F(a, b, d), B = F(\tilde{a}, \tilde{b}, \tilde{d})$ be two free groups of ranks 3. Consider two subgroups of rank 2:

$$C = \langle a^p, b\rangle \leqslant A, \tilde{C} = \langle \tilde{a}, \tilde{b}^p\rangle \leqslant B,$$

where $p \geqslant 2$ is an integer. Then the map ϕ defined by $\phi(a^k) = \tilde{a}, \phi(b) = \tilde{b}^k$ gives rise to an isomorphism $\phi : C \to \tilde{C}$. Put

$$G = A *_{C=\tilde{C}} B = \langle a, b, d, \tilde{a}, \tilde{b}, \tilde{d} \mid a^p = \tilde{a}, b = \tilde{b}^p\rangle.$$

Let S be a recursive set of representatives of A modulo C such that the representative in S of the coset Cda^{pm} is $b^{-pm}da^{pm}$ for all integers m. In particular,

$$da^{pm} = b^{pm}(b^{-pm}da^{pm}) \quad (m \in \mathbb{Z})$$

It is not difficult to construct such S since the set of elements of the type da^{pm} is recursive, as well as cosets of C. Similarly, let T be a recursive set of representatives of B modulo $\tilde{C}$ such that the representative in T of the coset $\tilde{C}\tilde{d}\tilde{b}^{pm}$ is $\tilde{a}^{-pm}\tilde{d}\tilde{b}^{pm}$ for all integers m, which implies that

$$\tilde{d}\tilde{b}^{pm} = \tilde{a}^{pm}(\tilde{a}^{-pm}\tilde{d}\tilde{b}^{pm}).$$

Now consider the following element in G:

$$g = \tilde{d}d\tilde{d}d\cdots\tilde{d}d\tilde{a} = g_1\cdots g_k$$

Then, in the notations of Algorithm I, the rewriting process (a) goes as follows:

$$c_k = \tilde{a} = a^p$$

$$g_{k-1} = d, g_{k-1}c_k = da^p = b^p(b^{-p}da^p) = b^p u_{k-1}$$

Now the next step will be

$$c_{k-1} = b^p = \tilde{b}^{p^2}, g_{k-2} = \tilde{d}$$

Hence

$$g_{k-2}c_{k-1} = \tilde{d}\tilde{b}^{p^2} = \tilde{a}^{p^2}(\tilde{a}^{-p^2}\tilde{d}\tilde{b}^{p^2}) = \tilde{a}^{p^2}\cdot u_{k-2} = c_{k-2}\cdot u_{k-2}.$$

In this case $\lambda = p$, lengths of the words c_i do not change in rewriting processes (b) and (c), so the word c_i grows every step by a factor of p, so

$$|c_1| = p^k$$

where $k = l(g) - 1$.

Example 3.4. Let $A = F(a,b)$, $B = F(a',b')$ be two free groups of rank 2. Consider two subgroups of rank 2, $C = \langle a, a^b\rangle$, $C' = \left\langle a'^p, a'^{b'}\right\rangle$, where $p \geqslant 2$ is an integer. Then the map ϕ defined by $\phi(a) = a'^{b'}$ and $\phi(a^b) = a'^p$ gives rise to an isomorphism $\phi C \to C'$. Put

$$G = A *_{C=C'} B = \left\langle a, b, a', b' | a = a'^{b'}, a^b = a'^p\right\rangle$$

Let S be a recursive set of representatives of A modulo B such that every element from $\langle b\rangle$ is in S. Analogously, T is the set of representatives B modulo C' such that every element from $\langle b'\rangle$ is in T. Now consider the following element in G:

$$g = (bb')^{-n}a(bb')^n = a^{p^n}$$

Rewriting of this element into the canonical form involves the exponential growth of the lengths of intermediate words c_i.

Now we turn to the complexity of rewriting processes (b) and (c). In general, this complexity depends on the particular algorithms for solving **CRSP**. In the case of free groups A and B the decision algorithm in [**19**] for solving **CRSP** have some important features. If we denote by $\bar{w}$ the representative of the coset Cw produced by the algorithm on the input word w, then the following conditions hold:

- For a given $w \in A$ the representative $\bar{w}$ of the coset Cw has the minimal possible length in Cw.

- There exists a constant M such that for a given $w \in A$ if $w = c\bar{w}$ for a (unique) $c \in C$ then $|c| \leqslant |w| + M$.
- the time spent by the algorithm on an input w is bounded from above by $L|w|$ for some fixed constant L.

This allows one to estimate the complexity of Algorithm I in the case of free groups. From now on we assume that Algorithm I has subalgorithms for solving **CRSP** which satisfy the conditions above.

LEMMA 3.5. *Let $A *_C B$ be a free product of free groups with finitely generated amalgamated subgroup C. Then the lengths of the words c_i that occur in computations with Algorithm I on an input w is bounded from above by*

$$\lambda^k \frac{|w| + M}{\lambda - 1}, \tag{7}$$

where $k = l(w)$.

PROOF. Let $w = g_1 \dots g_k$ be an input for Algorithm I in the form (5), where $k = l(w)$. It requires k steps for Algorithm I to produce the input. According to (6) on each step the length of the word c_j is bounded by

$$|c_j| \leqslant \lambda \cdot |c_{j+1}| + M + |g_j| \leqslant \lambda \cdot |c_{j+1}| + M + |w|.$$

Hence in k steps we will have the following estimate on the lengths of the words c_j, $j = 1, \dots k$.

$$\begin{aligned} \lambda(\cdots(\lambda(|w| + M) + |w| + M))\cdots) + |w| + M &= (\lambda^{k-1} + \cdots + 1)(|w| + M) \\ &\leqslant \lambda^k \frac{|w| + M}{\lambda - 1}, \end{aligned}$$

as required. □

COROLLARY 3.6. *Let $A *_C B$ be a free product of free groups with finitely generated amalgamated subgroup C. Then the time spent by Algorithm I on an input w is bounded above by*

$$k \cdot L_1 \cdot |w| \cdot \lambda^k \cdot (|w| + M)$$

where L_1 is a fixed constant and $k = l(w)$.

PROOF. Indeed, Algorithm I works k steps on an input w with $l(w) = k$. On each step it rewrites a current word c_j of the length bounded from above in (7). The rewriting involves the subalgorithms for solving **CRSP**. These algorithms spend at most linear time with respect to the length of the input. Putting all the estimates together we have the resulting estimate above. □

Combining the corollaries above with the example we have the following result.

THEOREM 3.7.

(1) *Let $A *_C B$ be a free product of free groups with finitely generated amalgamated subgroup C. Then Algorithm I has at most exponential (in the length of the input words) time complexity function bounded by:*

$$kL_1|w|\lambda^k(|w| + M)$$

where k, L_1, λ, M, and w are as above;

(2) *There are finitely generated free groups A and B and a finitely generated subgroup C in A and B such that in the free product with amalgamation $A *_C B$ the Algorithm I has precisely the exponential time complexity as above.*

However, we will show in the subsequent paper [**9**] that the situation in the example above is very rare, and in every free product with amalgamation $G = A *_C B$ of free groups with a finitely generated group C the Algorithm I is very fast on generic inputs.

3.4. Computation of cyclically reduced forms: Algorithm II. We fix the free product with amalgamation $G = A *_C B$ given by a presentation

$$G = \langle X \cup Y \mid R_A = 1, R_B = 1, u_z(x) = v_z(y), (z \in Z)\rangle.$$

In this section, we shall discuss the standard algorithm to find a cyclically reduced canonical form of an element $g \in G$ given in the form (5):

$$g = g_1 \cdots g_n,$$

where $g_1, \ldots, g_n$ are reduced words in X or in Y, and if g_i is a word in X, then g_{i+1} is a word in Y, and vice versa.

We work under assumption that the Coset Representative Search Problem (**CRSP**) and the Conjugacy Membership Search Problem (**CMSP**) are decidable in A and B for the subgroup C, and we have the decision algorithms in our possession. Notice that we need **CMSP** only because we have a slightly stronger notion of reduced forms than the usual one (see Section 1.2).

Observe, that the uniform version of **CMSP** is decidable in free groups and the decision algorithm has linear time complexity (in the length of the input word w) for a given finitely generated subgroup C [**19**].

Algorithm II: Computing Cyclically Reduced Forms.

Input: a word g in the form (5).

Step 0: Find the canonical form of g using the Algorithm I:

$$g = c_g g_1 \cdots g_k.$$

Observe that $l(g) = k$ and for every g_i we know its factor $F(g_i)$.

Induction Step:

- If $l(g) = 0$ then g is already in cyclically reduced from.
- If $l(g) = 1$, for example, if $g \in A$, then check whether g is a conjugate of an element $c \in C$ or not, using the algorithm for **CMSP**. In the former case, c is a cyclically reduced form of g and the algorithm for **CMSP** gives one of such elements c. In the latter case, g is already in cyclically reduced form.
- Let $l(g) \geqslant 2$.
  - If $F(g_1) \neq F(g_k)$, then g is already in a cyclically reduced canonical form.
  - If $F(g_1) = F(g_k)$. Then g is conjugate to

$$(g_k c_q g_1) g_2 \cdots g_{k-1}.$$

Now apply the decision algorithm for **CRSP** to the word $(g_k c_q g_1)$ to find the canonical form $c' g_1'$ of it. If $g_1' \neq 1$ then

$$c' g_1' g_2 \cdots g_{k-1}$$

is a cyclically reduced canonical form of g. Otherwise,

$$F(c' g_2) = F(g_{k-1})$$

and we apply the procedure above to $c' g_2 \cdots g_{k-1}$.

End of Algorithm II

Remark 3.8. If $l_0(g) \geqslant 2$ then Algorithms II needs only a decision algorithm for **CRSP** to find a cyclically reduced form of g.

Theorem 3.9. *Let $G = A *_C B$ and the problems* **CRSP** *and* **CMSP** *are decidable in A and B for the subgroup C. Then for a given element $g \in G$ Algorithm II finds a cyclically reduced canonical form of g in time $T_{II}(g)$ which can be bounded from above as follows:*

$$T_{II}(g) \leqslant T_I(g) + K \cdot \max\{T_{CMSP}(c_g g_1), T_{CRSP}(g_k c_g g_1) \cdot l(g)\},$$

where T_I, T_{CMSP}, T_{CRSP} are the time functions, correspondingly, of Algorithm I, and the decision algorithms for **CMSP**, **CRSP**, *K is a constant. In particular, if A and B are free groups then*

$$T_{II}(g) \leqslant T_I(g) + K_1 \cdot |g| \cdot l(g),$$

where K_1 is a constant (depending on C) and $|g|$ is the length of the input g given as a word in $F(X \cup Y)$.

4. Conjugacy Search Problem for regular elements

4.1. Regular Elements. In this section we introduce and study *regular* elements. Roughly speaking, regular elements are those elements of $G = A *_C B$ for which the condition 1) in the Conjugacy Criterion - does not apply.

Definition 4.1. We say that $(c, g) \in C \times G$ is a bad pair if $c \neq 1, g \notin C$, and $gcg^{-1} \in C$.

Notice that if (c, g) is a bad pair then $g \in N_G^*(C) \setminus C$ and $c \in Z_g(C)$. The following lemma gives a more detailed description of bad pairs.

Lemma 4.2. *Let $c \in C \setminus \{1\}$ and $g \in G \setminus C$. If $g = c_g p_1 \cdots p_k$ is the canonical normal form of g then (c, g) is a bad pair if and only if the following system $B_{c,g}$ has a solution $c_1, \ldots, c_k$ with $c_i \in C$:*

$$\begin{aligned} p_k c p_k^{-1} &= c_1 \\ p_{k-1} c_1 p_{k-1}^{-1} &= c_2 \\ &\vdots \\ p_1 c_{k-1} p_1^{-1} &= c_k \end{aligned}$$

Moreover, in this case $p_i \in N^*_{F(p_i)}(C)$ and $c \in Z_A(C) \cup Z_B(C)$.

Proof. This lemma is a particular case of Lemma 4.5. □

Observe, that consistency of the system $B_{c,g}$ does not depend on a particular choice of representatives of A and B modulo C.

Now we specify, in our particular context, the general concepts of a "black hole" and "regular part" as discussed in the Introduction.

DEFINITION 4.3. The set

$$BH = (N^*{}_G(C) \setminus C) \cup Z_G(C)$$

is called a *black hole*. Elements from BH are called *singular*, and elements from $G \setminus BH$ *regular*.

Notice that if the subgroup C has a finite malnormality degree in G then every element g with $l_0(g) > md_G(C)$ is regular. In particular, it follows from Lemma 1.8 that if C is malnormal in A or in B then every element $g \in G$ with $l(g) \geqslant 2$ is regular. Notice also, that if $g \in G \setminus C$ is regular then all elements in CgC are regular.

Observe, that the condition 1) in the Conjugacy Criterion, indeed, does not apply for regular elements.

The following description of singular elements follows from Lemma 4.2.

COROLLARY 4.4. *Let $G = A *_C B$. Then:*

1) *an element $g \in G \setminus C$ is singular if and only if the system $B_{g,c}$ has a solution $c, c_1, \ldots, c_k$, where c, c_i are non-trivial elements from C;*
2) $Z_G(C) = Z_A(C) \cup Z_B(C)$

As we have seen already, an element $g \in G$ is singular if and only if the system $gc = c_1 g$ has a nontrivial solution c, c_1 in C. Now we will study slightly more general equations of the type $gc = c'g'$ and their solutions c, c' in C.

LEMMA 4.5. *Let $G = A *_C B$, $g, g' \in G$ be elements given by their canonical forms:*

$$g = c_g p_1 \cdots p_k, \quad g' = c_{g'} p'_1 \cdots p'_k \quad (k \geqslant 1). \tag{8}$$

Then the equation $gc = c'g'$ has a solution $c, c' \in C$ if and only if the following system $S_{g,g'}$ in variables $c, c', c_1, \ldots, c_k$ has a solution in C:

$$\begin{aligned} p_k c &= c_1 p'_k \\ p_{k-1} c_1 &= c_2 p'_{k-1} \\ &\vdots \\ p_1 c_{k-1} &= c_k p'_1 \\ c_g c_k &= c' c_{g'} \end{aligned}$$

Proof. Let $c, c' \in C$ be a solution to the equation $gc = c'g'$. We then rewrite the equality $gc = c'g'$ as

$$c_g p_1 \cdots p_k c = c' c_{g'} p'_1 \cdots p'_k.$$

Notice that the right hand side of this equality is in the canonical form. Following Algorithm I we shall rewrite the left hand side of this equality into the canonical form. After rewriting the both sides must coincide as the canonical normal forms of the same element. This gives rise to the system of equations for some elements $c, c', c_1, \ldots, c_k \in C$, as above. Conversely, if the system $S_{g,g'}$ has a solution then the elements c, c' give a solution of the equation $gc = c'g'$. □

The first k equations of the system $S_{g,g'}$ form what we call the *principal system* of equations, we denote it by $PS_{g,g'}$. In what follows we consider $PS_{g,g'}$ as a system in variables $c, c_1, \dots, c_k$ which take values in C, the elements $p_1, p'_1, \dots, p_k, p'_k$ are constants.

To study solution sets of the system $PS_{g,g'}$ we need the following definition.

DEFINITION 4.6. Let M be a subset of a group G. If $u, v \in G$ then the set uMv is called a G-shift of M. For a set $\mathcal{M}$ of subgroups of G denote by $\Phi(\mathcal{M}, G)$ the least set of subsets of G which contains $\mathcal{M}$ and is closed under G-shifts and intersections.

LEMMA 4.7. *Let G be a group and C be a subgroup of G. If $D \in \Phi(\{C\}, G)$, $D \neq \emptyset$ then there exist elements $g_1, \dots, g_n, h \in G$ such that*

$$D = (C^{g_1} \cap \dots \cap C^{g_n})h$$

In particular, non-empty sets in $\Phi(\{C\}, G)$ are particular cosets from G.

PROOF. Induction on the number of operations required to construct D from C. For a tuple $\bar{g} = (g_1, \dots, g_n)$ of elements from G put

$$C_{\bar{g}} = (C^{g_1} \cap \dots \cap C^{g_n}).$$

Let $D = C_{\bar{g}}h$ for some $\bar{g} \in G^n, h \in G$. Then for any $a, b \in G$:

$$aDb = D^{a^{-1}}ab = C_{\bar{g}a^{-1}}ab,$$

where $\bar{g}a^{-1} = (g_1a^{-1}, \dots, g_na^{-1})$, i.e., aDb is in the required form.

Observe, that for arbitrary subgroups $K, L \leqslant G$ and elements $a, b \in G$ if $h \in Ka \cap Lb$ then

$$Ka \cap Lb = (K \cap L)h. \tag{9}$$

Therefore, if $h_3 \in C_{\bar{g}_1}h_1 \cap C_{\bar{g}_2}h_2$, then

$$C_{\bar{g}_1}h_1 \cap C_{\bar{g}_2}h_2 = (C_{\bar{g}_1} \cap C_{\bar{g}_2})h_3 = C_{\bar{g}_3}h_3,$$

where $\bar{g}_3$ is concatenation of $\bar{g}_1$ and $\bar{g}_2$. □

LEMMA 4.8. *Let $G = A *_C B$. Then for given two elements $g, g' \in G$ in their canonical forms*

$$g = c_g p_1 \cdots p_k, \ g' = c_{g'} p'_1 \cdots p'_k \quad (k \geqslant 1)$$

the set $E_{g,g'}$, of all elements c in C for which the system $PS(g, g')$ has a solution $c, c_1, \dots, c_k \in C$, is equal to

$$E_{g,g'} = C \ \cap p_k^{-1}Cp'_k \ \cap \ \cdots \ \cap p_k^{-1} \cdots p_1^{-1}Cp'_1 \cdots p'_k.$$

In particular, if $E_{g,g'} \neq \emptyset$ then $E_{g,g'} = C_{g,g'}c_{g,g'}$ for some subgroup $C_{g,g'} \leqslant C$ and some element $c_{g,g'} \in C$.

PROOF. Let

$$g = c_g p_1 \cdots p_k, \quad g' = c_{g'} p'_1 \cdots p'_k.$$

Denote by V_i the set of all solutions $(c, c_1, \dots, c_i) \in C^{i+1}$ of the system formed by the first i equations of $PS(g, g')$. Let $D_{m,i}$ be the projection of V_i onto its m-s component.

The first equation of the system $PS(g, g')$ gives:

$$p_k c_0 (p'_k)^{-1} = c_1$$

where for uniformity we denote c by c_0. Therefore,

$$D_{1,1} = p_k C (p'_k)^{-1} \cap C, \quad D_{0,1} = p_k^{-1} D_{1,1} p'_k$$

and $(c_0, c_1) \in V_1$ if and only if

$$c_1 \in D_{1,1}, \quad c_0 = p_k^{-1} c_1 p'_k.$$

Clearly, the sets $D_{0,1}$ and $D_{1,1}$ are in Φ_C.

Now we rewrite the i-s equation $p_{k-i+1} c_{i-1} = c_i p'_{k-i+1}$ of the system $PS(g, g')$ in the form

$$p_{k-i+1} c_{i-1} (p'_{k-i+1})^{-1} = c_i$$

It follows that

$$D_{i,i} = p_{k-i+1} D_{i-1,i-1} (p'_{k-i+1})^{-1} \cap C, \tag{10}$$

where $i = 1, \ldots, k$ and $D_{0,0} = C$. In particular

$$D_{k,k} = p_1 D_{k-1,k-1} (p'_1)^{-1} \cap C$$

Clearly, $(c, c_1, \ldots, c_k)$ is a solution of the system $PS(g, g')$ if and only if $c_k \in D_{k,k}$ and $c_{i-1} = p_{k-i+1}^{-1} c_i p'_{k-i+1}$. More precisely, since

$$D_{i-1,k} = p_{k-i+1}^{-1} D_{i,k} p'_{k-i+1}$$

it follows now that,

$$D_{k-i,k} = D_{k-i,k-i} \cap p_i^{-1} C p'_i \cap \cdots \cap p_i^{-1} \cdots p_1^{-1} C p'_1 \cdots p'_i.$$

In particular,

$$E_{g,g'} = D_{0,k} = C \cap p_k^{-1} C p'_k \cap \cdots \cap p_k^{-1} \cdots p_1^{-1} C p'_1 \cdots p'_k.$$

So $E_{g,g'} \in \Phi(C, G)$. By Lemma 4.7

$$p_k^{-1} C p'_k \cap \cdots \cap p_k^{-1} \cdots p_1^{-1} C p'_1 \cdots p'_k = Hu$$

for some subgroup $H \leqslant G$ and $u \in G$. Now we can see from (9) that

$$E_{g,g'} = C \cap Hu = C_{g,g'} c_{g,g'}$$

for some subgroup $C_{g,g'} \leqslant C$ and $c_{g,g'} \in C$, as required. □

Denote buy $Sub(C)$ the set of all subgroups of C. By Lemma 4.7 non-empty sets from $\Phi(Sub(C), A)$ (respectively, from $\Phi(Sub(C), B)$) are some cosets of subgroups from A (respectively, from B).

Corollary 4.9. *Let $G = A *_C B$. If the Cardinality Search Problem is decidable for $\Phi(Sub(C), A)$ in A and for $\Phi(Sub(C), B)$ in B then given g, g' as above, one can effectively find the set $E_{g,g'}$. In particular, one can effectively check whether or not $E_{g,g'}$ is empty, singleton, or infinite.*

Proof. In notations of Lemma 4.8

$$E_{g,g'} = p_k^{-1} \cdots p_1^{-1} D_{k,k} p'_1 \cdots p'_k.$$

Therefore it suffices to solve the cardinality problem for the set $D_{k,k}$. The quality 10

$$D_{i,i} = p_{k-i+1} D_{i-1,i-1} (p'_{k-i+1})^{-1} \cap C,$$

and Lemma 4.7 show that each $D_{i-1,i-1}$ is a coset of the type $C_i c_i$ where $C_i \leqslant C$ and $c_i \in C$. Moreover, since the Cardinality Search Problem is decidable for

$\Phi(Sub(C), A)$ in A, and for $\Phi(Sub(C), B)$ in B, the equality (9) shows how one can effectively find the element c_i and the direct expression for the subgroup C_i (in terms of shifts and intersections). Therefore, in k steps one can find $D_{k,k}$, and hence the set $E_{g,g'}$. Moreover, on each step one can find the cardinality of the set $D_{i,i}$. This proves the corollary. □

LEMMA 4.10. *Let $G = A *_C B$ and $g, g' \in G$. If $l(g) = l(g') \geqslant 1$ and the system $PS(g, g')$ has more then one solution in C then the elements g, g' are singular.*

PROOF. Let $c, c_1, \ldots, c_k$ and $b, b_1, \ldots, b_k$ be two distinct solutions of the principal system $PS(g, g')$. Denote for uniformity $c_0 = c, b_0 = b$. Hence we have the following systems of equations:

$$p_k c_0 = c_1 p'_k, \quad p_k b_0 = b_1 p'_k$$

$$p_{k-1} c_1 = c_2 p'_{k-1}, \quad p_{k-1} b_1 = b_2 p'_{k-1}$$

$$\vdots$$

$$p_1 c_{k-1} = c_k p'_1, \quad p_1 b_{k-1} = b_k p'_1$$

Expressing p'_k from the first two equations in the system above, and then p'_{k-1} from the next two equations, and so on, we get the following equalities:

$$c_1{}^{-1} p_k c_0 = b_1{}^{-1} p_k b_0$$
$$c_2{}^{-1} p_{k-1} c_1 = b_2{}^{-1} p_{k-1} b_1$$
$$\vdots$$
$$c_k{}^{-1} p_1 c_{k-1} = b_k{}^{-1} p_1 b_{k-1}$$

Rewriting these equalities we obtain:

$$p_k^{-1} b_1 c_1^{-1} p_k = b_0 c_0^{-1},$$
$$p_{k-1}^{-1} b_2 c_2^{-1} p_{k-1} = b_1 c_1^{-1},$$
$$\vdots$$
$$p_1^{-1} b_k c_k^{-1} p_1 = b_{k-1} c_{k-1}^{-1}.$$

Observe that all the elements $b_i c_i^{-1}$ are non-trivial. By Lemma 4.2 the element g is singular. Similar argument shows that g' is also singular. □

The next result shows that one can effectively determine whether a given element $g \in G$ is regular or not.

LEMMA 4.11. *Let $G = A *_C B$ be a free product of finitely presented groups A and B amalgamated over a finitely generated subgroup C. Assume also that A and B allow algorithms for solving the following problems:*

- *Coset Representative Search Problem for the subgroup C.*
- *Cardinality Search Problem for $\Phi(Sub(C), A)$ in A and for $\Phi(Sub(C), B)$ in B.*
- *Malnormality problem for C in A and in B.*

Then there exists an algorithm to determine whether a given element in G is regular or not.

PROOF. For a given $g \in G$ we can find the canonical normal form of g using Algorithm II. Now there are two cases to consider.

1) If $l(g) > 1$ then by Lemma 4.2 g is a singular element if and only if the system $B_{c,g}$ has a nontrivial solution $c, c_1, \ldots, c_k \in C$. Observe, that if the system $B_{c,g}$ has two distinct solutions then one of them is non-trivial (i.e., $c, c_1 \ldots, c_k \neq 1$).

Now if $B_{c,g}$ has no solutions in C (and we can check it effectively) then g is regular. If $B_{c,g}$ has precisely one solution then we can find it and check whether it is trivial or not, hence we can find out whether g is regular or not. If $B_{c,g}$ has more then one solution (and we can verify this effectively) then g is not regular.

2) If $l(g) = 0$ then g is regular if and only if $g \notin Z_G(C)$. By Corollary 4.4 $Z_G(C) = Z_A(C) \cup Z_B(C)$. Since the Malnormality Problem is decidable for C in A and B the sets $Z_A(C)$ and $Z_B(C)$ are recursive, as well as their union. □

COROLLARY 4.12. *Let $G = A *_C B$ be a free product with amalgamation of free groups A, B. Then the set of regular elements in G is recursive.*

REMARK 4.13. The decision algorithm for checking whether a given element is regular or not is fast "modulo" Algorithm I and the algorithm $\mathcal{B}$ for finding cardinality of sets of the type $E_{g,g'}$. In general, both Algorithm I and $\mathcal{B}$ can be exponential in the worst case. However, we will show later that generically both the algorithms are fast.

Denote by CR the set of all elements in G which have at least one regular cyclically reduced canonical form, i.e., CR is the set of elements in G which are conjugates of cyclically reduced regular elements. Observe that if g is cyclically reduced regular element in G with $l(g) \geqslant 1$ then g^c is regular for every $c \in C$ (since if $g \in N_G^*(C)$ then $CgC \subset N_G^*(C)$). Therefore, if one of the cyclically reduced canonical forms of g is regular then all of these forms are regular. The set CR plays an important part in our analysis of the conjugacy search problems in G.

LEMMA 4.14. *Let $G = A *_C B$. Assume also that A and B allow algorithms for solving the following problems:*

- *Coset Representative Search Problem for the subgroup C.*
- *Cardinality Search Problem for $\Phi(Sub(C), A)$ in A and for $\Phi(Sub(C), B)$ in B.*
- *Malnormality problem for C in A and in B.*

Then there exists an algorithm $\mathcal{A}$ to determine whether a given element in G is in CR or not.

PROOF. Follows from Lemma 4.11 and Theorem 3.9. □

4.2. Conjugacy Search problems and regular elements. The aim of this section is to study Conjugacy Search Problem for regular elements in free products with amalgamation $G = A *_C B$. We show that the conjugacy search problem for regular elements is solvable under some very natural restrictions on the factors.

We start with the following particular case of the Conjugacy Search Problem:

The Conjugacy Search Problem for a fixed element g: *It is the Conjugacy Search Problem for the set of pairs*

$$\Phi_g = \{(g, u) \mid u \in G\}.$$

THEOREM 4.15. *Let $G = A *_C B$ be a free product of finitely presented groups A and B amalgamated over a finitely generated subgroup C. Assume also that A and B allow algorithms for solving the following problems:*

- *Coset Representative Search Problem for the subgroup C.*
- *Cardinality Search Problem for $\Phi(Sub(C), A)$ in A and for $\Phi(Sub(C), B)$ in B.*

Then the Conjugacy Search Problem in G is decidable for cyclically reduced regular elements g of length $l(g) > 1$.

PROOF. Let g be a fixed regular cyclically reduced element, and g' be an arbitrary element from G. Applying Algorithm I we can find the canonical forms of g and g'. In view of this we can assume from the beginning that g and g' are given already in their cyclically reduced canonical forms:

$$g = cp_1 \dots p_k, \quad g' = c' p'_1 \dots p_{k'}{}'.$$

According to the Conjugacy Criterion, the elements g and g' are conjugate in G if and only if $k = k'$ and for some cyclic permutation $\pi(g')$ of g' the equation $c^{-1}gc = \pi(g')$ has a solution c in C. So the Conjugacy Search Problem is decidable in G for regular elements g if and only if the Diophantine problem is decidable for equations of the type $c^{-1}gc = \pi(g')$ (i.e., one can determine algorithmically whether a given equation of this type has a solution in C or not). By Lemma 4.5 the equation $c^{-1}gc = \pi(g')$ has a solution in C if and only if the system $S_{g,\pi(g')}$ has a solution in C. Since g is regular the system $PS_{g,\pi(g')}$ has at most one solution in C. Decidability of the Cardinality Search Problem problems for $\Phi(Sub(C), A)$ in A and for $\Phi(Sub(C), B)$ in B allows one to check whether $PS_{g,\pi(g')}$ has a solution in C or not, and if it does, one can find the solution. Now one can verify whether this solution satisfies the last equation of the system $S_{g,\pi(g')}$ or not. If not, the system $S_{g,\pi(g')}$ has no solutions in C, as well as the equation $c^{-1}gc = \pi(g')$. Otherwise, the system $S_{g,\pi(g')}$ and the equation $c^{-1}gc = \pi(g')$ have solutions in C and we have found one of these solutions. This proves the lemma. □

Now we study conjugacy search problem for regular elements of length $\leqslant 1$.

LEMMA 4.16. *Let $G = A *_C B$ and g be a cyclically reduced regular element of G with $l(g) \leqslant 1$. If the Coset Representatives Search problem for C in A and B and the Conjugacy Search Problem in A and in B are decidable then the Conjugacy Search Problem for g is decidable in G.*

PROOF. It follows from the conjugacy criterion. □

REMARK 4.17. The decision algorithms from Theorem 4.15 and Lemma 4.16 have polynomial time complexity "modulo" the algorithms for finding canonical forms of elements and the decision algorithms for the problems listed in the statements.

We are ready to formulate a general conjugacy search problem for regular elements.

Recall that by CR we denote the set of all conjugates in G of cyclically reduced regular elements.

The Conjugacy Search Problem for CR: *is the Conjugacy Search Problem for the set of pairs*

$$\Phi_g = \{(g, u) \mid g \in CR, u \in G\}.$$

THEOREM 4.18. *Let $G = A *_C B$ be a free product of finitely presented groups A and B amalgamated over a finitely generated subgroup C. Assume also that A and B allow algorithms for solving the following problems:*

- *Coset Representative Search Problem for the subgroup C.*
- *Cardinality Search Problem for $\Phi(Sub(C), A)$ in A and for $\Phi(Sub(C), B)$ in B.*
- *Conjugacy Search Problem in A and in B.*
- *Conjugacy Membership Search Problem for C in A and B.*

Then the Conjugacy Search Problem in G is decidable for elements from CR.

COROLLARY 4.19. *Let $G = A*_C B$ be a free product of free groups A and B with amalgamated finitely generated subgroup C. Then the Conjugacy Search Problem in G is decidable for elements from CR.*

COROLLARY 4.20. *Let $G = A *_C B$ and C is malnormal in A. If*

- *Coset Representative Search Problem for the subgroup C.*
- *The Conjugacy Search Problem decidable in A and in B.*
- *Cardinality Search Problem for $\Phi(Sub(C), A)$ in A and for $\Phi(Sub(C), B)$ in B.*

then:

(1) *There exists an algorithm for solving the conjugacy problem in G;*
(2) *Two elements from C are conjugate in G if and only if they are conjugate in B;*

PROOF. Since C is malnormal in A every element $g \in G$ with $l(g) \geqslant 2$ is regular (see Section 1.4). Hence by Theorem 4.15 conjugacy problem for every g with $l(g) \geqslant 2$ is decidable.

Assume now that $c_1, c_2 \in C$ are conjugate in G. By the Conjugacy Criterion, there exists a sequence of elements $c_1 = d_1, d_2, \ldots, d_k = c_2$ from C such that the neighboring elements are conjugate either in A or in B. By malnormality of C in A this implies that c_1 and c_2 are conjugate in B, which is algorithmically decidable since the Conjugacy Search Problem is decidable in B. This proves the corollary. □

References

[1] S. Adian and V. Durnev, *Algorithmic problems for groups and semigroups,* Uspekhi Mat. Nauk, **55**, no. 2, 3–94, 2000; translation in Russian Math. Surveys, **55**, no. 2, 207–296, 2000.

[2] J. Alonso, T. Brady, D. Cooper, V. Ferlini, M. Lustig, M. Mihalik, M. Shapiro and H. Short, *Notes on hyperbolic groups,* In: *Group theory from a geometrical viewpoint*, Proceedings of the workshop held in Trieste, É. Ghys, A. Haefliger and A. Verjovsky (editors). World Scientific Publishing Co., 1991.

[3] I. Anshel, M. Anshel and D. Goldfeld, *An algebraic method for public-key cryptography.* Math. Res. Lett. **6** (1999), 287–291.

[4] G. Baumslag, S. M. Gersten, M. Shapiro and H. Short, *Automatic groups and amalgams.* J. Pure Appl. Algebra **76** (1991), 229–316.

[5] G. Baumslag, A. G. Myasnikov and V. N. Remeslennikov, *Malnormality is decidable in free groups*, preprint, City Colelge of CUNY, New York, 1997.

[6] M. Bestvina and M. Feighn, *A combination theorem for negatively curved groups,* J. Differential Geom. **35** (1992), no. 1, 85–101.

[7] L. A. Bokut and G. P. Kukin, *Algorithmic and combinatorial algebra*, Math. and its Applications, **255**, Kluwer Academic Publishers Group, Dordreht, 1994.
[8] A. V. Borovik, A. G. Myasnikov and V. N. Remeslennikov, *Multiplicative measures on free groups*, Internat. J. Algebra Comp., **13** no. 6 (2003), 705–731.
[9] A. V. Borovik, A. G. Myasnikov and V. N. Remeslennikov, *The conjugacy problem in amalgamated products II: random normal forms and generic complexity of algorithmic problems*, submitted.
[10] A. V. Borovik, A. G. Myasnikov and V. N. Remeslennikov, *Conjugacy problem in HNN-extensions: regular elements, black holes, and generic complexity*, submitted.
[11] A. V. Borovik, A. G. Myasnikov and V. N. Remeslennikov, Algorithmic stratification of the conjugacy problem in Miller's groups, submitted.
[12] I. Bumagina, *The conjugacy problem for relatively hyperbolic groups*, Algebraic and Geometric Topology, to appear.
[13] P. Dehornoy, *Braid-based cryptography*, Contemporary Mathematics, **360** (2004), 5–33.
[14] D. Epstein, J. Cannon, D. Holt, S. Levy, M. Paterson and W. Thurston, *Word Processing in Groups,* Jones and Bartlett, Boston, 1992.
[15] B. Farb, *Automatic groups: a guided tour,* Enseign. Math. (2) **38** (1992), 291–313.
[16] B. Farb, *Relatively hyperbolic groups*, Geometric and functional analysis, **8** (1998), 810–840.
[17] D. Gildenhuys, O. Kharlampovich and A. Myasnikov, *CSA groups and separated free constructions*, Bull. Austr. Math. Soc. **52** (1995), 63–84.
[18] M. Gromov, *Hyperbolic groups*, Essays in group theory, Springer, New York, 1987, pp. 75–263.
[19] I. Kapovich and A. G. Myasnikov, *Stallings foldings and subgroups of free groups*, J. Algebra **248** (2002), 608–668.
[20] I. Kapovich, A. Myasnikov, P. Schupp and V. Shpilrain *Generic-case complexity and decision problems in group theory*, J. Algebra, **264** (2003), 665–694.
[21] O. Kharlampovich and A. Myasnikov. *Hyperbolic groups and free constructions.* Transactions of Math., **350** no. 2 (1998), 571–613.
[22] K. H. Ko, S. J. Lee, J. H. Cheon, J. W. Han, J. Kang and C. Park, *New public-key cryptosystem using braid groups*, Advances in cryptology—CRYPTO 2000 (Santa Barbara, CA), 166–183, Lect. Notes Comp. Sci. **1880**, Springer, Berlin, 2000.
[23] R. Lyndon, P. Schupp, *Combinatorial Group Theory*, Springer, 1977.
[24] W. Magnus, A. Karras and D. Solitar, *Combinatorial Group Theory*, Interschience Pulishers, New York a. o., 1966.
[25] C. F. Miller III, *On group-theoretic decision problems and their classification,* Ann. of Math. Studies, **68** (1971). Princeton University Press, Princeton.
[26] C. Miller III, *Decision problems for groups - Survey and reflections*, in "Algorithms and Classification in Combinatorial Group Theory" (G. Bamuslag and C.F. Miller III, eds), Springer, 1992, pp. 1–60.
[27] K. V. Mikhajlovski and A. Yu. Olshanskii, *Some constructions relating to hyperbolic groups*, 1994, Proc. Int. Conf. on Cohomological and Geometric Methods in Group Theory.
[28] A. Myasnikov, A. Ushakov, *Random van Kampen Diagrams and algorithmic problems in groups*, to appear.
[29] D. Osin, *Relatively hyperbolic groups: Intrinsic geometry, algebraic properties, and algorithmic problems*, Memoirs Amer. Math. Soc., to appear.
[30] C. Papadimitriou, *Computation Complexity*, (1994), Addison-Wesley, Reading.
[31] G. Petrides, *Cryptanalysis of the public key cryptosystem based on the word problem on the Grigorchuk groups*, in: *Cryptography and Coding. 9th IMA Internat. Conf., Cirencister, UK, Dec 2003*, Lect. Notes Comp. Sci. **2898**, Springer-Verlag, 2003, 234–244.
[32] V. N. Remeslennikov and V. A. Romankov, *Algorithmic and model theoretic problems in groups*, Itogi Nauki, Algebra, Topology and Geometry, **21** (1983), 3–89.
[33] V. Shpilrain, *Assessing security of some group based cryptosystems*, Contemp. Math., Amer. Math. Soc. **360** (2004), 167–177.

Alexandre V. Borovik, School of Mathematics, PO Box 88, Sackville Street, University of Manchester, Manchester M60 1QD, United Kingdom
E-mail address: alexandre.borovik@manchester.ac.uk

Alexei G. Myasnikov, Department of Mathematics, The City College of New York, New York, NY 10031, USA
E-mail address: alexeim@att.net

Vladimir N. Remeslennikov, Omsk Branch of the Mathematical Institute SB RAS, 13 Pevtsova Street, 644099 Omsk, Russia
E-mail address: remesl@iitam.omsk.net.ru