ELECTRONIC RESEARCH ANNOUNCEMENTS
OF THE AMERICAN MATHEMATICAL SOCIETY
Volume 9, Pages 40–50 (June 24, 2003)
S 1079-6762(03)00110-0

THE CONJUGACY PROBLEM FOR GROUPS, AND HIGMAN EMBEDDINGS

A. YU. OL'SHANSKII AND M. V. SAPIR

(Communicated by Efim Zelmanov)

ABSTRACT. For every finitely generated recursively presented group $\mathcal{G}$ we construct a finitely presented group $\mathcal{H}$ containing $\mathcal{G}$ such that $\mathcal{G}$ is (Frattini) embedded into $\mathcal{H}$ and the group $\mathcal{H}$ has solvable conjugacy problem if and only if $\mathcal{G}$ has solvable conjugacy problem. Moreover, $\mathcal{G}$ and $\mathcal{H}$ have the same r.e. Turing degrees of the conjugacy problem. This solves a problem by D. Collins.

In 1961, G. Higman [Hi] published the celebrated theorem that a finitely generated group is recursively presented if and only if it is a subgroup of a finitely presented group. Along with the results of Novikov and Boone [Rot] this result showed that objects from logic (in that case, recursively enumerable sets) have group theoretic characterizations (see Manin [Ma] for the philosophy of Higman embeddings).

Clapham [Cla] (see also corrections in [Va]) was probably the first to investigate properties preserved under Higman embeddings. In particular, he slightly modified the original Higman construction and showed that his embedding preserves solvability (and even the r.e. Turing degree) of the word problem. Valiev [Va] gave a new construction of Higman embedding, and sketched a proof of a much stronger result: every finitely generated recursively presented group G embeds into a finitely presented group H such that the word problem in H reduces to the word problem in G in polynomial time. This implies that if the word problem of a finitely generated group is in P (i.e., can be solved in polynomial time by a deterministic Turing machine), then it can be embedded into a finitely presented group whose word problem is also in P. In [BORS], Birget, Rips and the authors of this paper obtained a group theoretic characterization of NP (nondeterministic polynomial time): a finitely generated group G has word problem in NP if and only if G is embedded into a finitely presented group with polynomial Dehn function.

The conjugacy problem turned out to be much harder to preserve under embeddings. Gorjaga and Kirkinskiĭ [GK] and Collins and Miller [CM] proved that even subgroups of index 2 of finitely presented groups do not inherit solvability or unsolvability of the conjugacy problem.

Received by the editors March 2, 2003.
2000 *Mathematics Subject Classification.* Primary 20F10; Secondary 03D40, 20M05.
Both authors were supported in part by the NSF grant DMS 0072307. In addition, the research of the first author was supported in part by the Russian Fund for Basic Research 02-01-00170 and by the INTAS grant 99-1224; the research of the second author was supported in part by the NSF grant DMS 9978802 and the US-Israeli BSF grant 1999298.

In 1976 D. Collins [KT] posed the following question (problem 5.22): *Does there exist a version of the Higman embedding theorem in which the degree of unsolvability of the conjugacy problem is preserved?*

It was quickly realized that the main problem would be in preserving the smallest Turing degree, that is, the solvability of the conjugacy problem. In 1980, Collins [Col] analyzed existing proofs of Higman's theorem and discovered that there are essential difficulties. If a finitely generated group C is embedded into a finitely presented group B using any existing at that time constructions, then the solvability of the conjugacy problem in B implies certain properties of C which are much stronger than solvability of the conjugacy problem. Collins even wrote the following pessimistic comments: "There seems at present to be no hope to establishing the analogue of Clapham's theorem. ... Furthermore these difficulties seem to be more or less inevitable given the structure of the proof and probably a wholly new strategy will be needed to avoid them. For the present the most one can be hoped for is the isolation of conditions on C that are necessary and sufficient for the preservation of the solvability of the conjugacy problem in the Higman embedding."

Recall that a subgroup $\mathcal{G}$ of a group $\mathcal{H}$ is called *Frattini embedded* if any two elements of $\mathcal{G}$ that are conjugate in $\mathcal{H}$ are also conjugate in $\mathcal{G}$. Clearly, if $\mathcal{G}$ is Frattini embedded into $\mathcal{H}$ and $\mathcal{H}$ has solvable conjugacy problem, then $\mathcal{G}$ has solvable conjugacy problem.

The following theorem solves the problem of Collins:

Theorem 1. *A finitely generated group $\mathcal{G}$ has conjugacy problem of recursively enumerable Turing degree α if and only if $\mathcal{G}$ can be Frattini embedded into a finitely presented group $\mathcal{H}$ with conjugacy problem of the same Turing degree. In particular, a finitely generated group has solvable conjugacy problem if and only if it is Frattini embedded into a finitely presented group with solvable conjugacy problem.*

Consider the case when $\mathcal{G} = \langle A \mid \mathcal{E}\rangle$ has decidable conjugacy problem and we want to embed it into a finitely presented group with decidable conjugacy problem (the case when the conjugacy problem in $\mathcal{G}$ has an arbitrary r.e. Turing degree is similar). The standard idea is to start with a machine that recognizes the set $\mathcal{E}$ and then interpret this machine in a group. Of course we would like to have a machine which is easier to interpret. The most suitable machines for our purposes are the so called S-machines invented by the second author in [SBR] and successfully used in [SBR], [BORS], [OlSa01]. An S-machine works with words which can have complicated structure. Different parts of these words can be elements of different groups (so we do not distinguish between words whose respective parts are equal in the corresponding groups). As in Turing machines, every rule of an S-machine replaces subwords containing state letters by other subwords. The advantage of S-machines is that they are very easily simulated by groups.

Let us start with an S-machine which essentially goes back to C. Miller [Mil] (although Miller did not use S-machines). Assume that A is a symmetric set, that is, it contains the formal inverses of its elements, and let us embed $\mathcal{G} = \langle A \mid \mathcal{E}\rangle$ into *some* finitely presented group $\bar{\mathcal{G}} = \langle A \cup Y \mid \bar{\mathcal{E}}\rangle$ using, say, the standard Higman embedding. Now consider the S-machine $\mathcal{M}$ with one state letter q and the tape alphabet $A \cup Y$ regarded as a generating set of the free group, and the following commands (each command says which subwords of a word should be replaced and gives the replacement word):

- $[q \to a^{-1}qa]$, for every $a \in A \cup Y$,

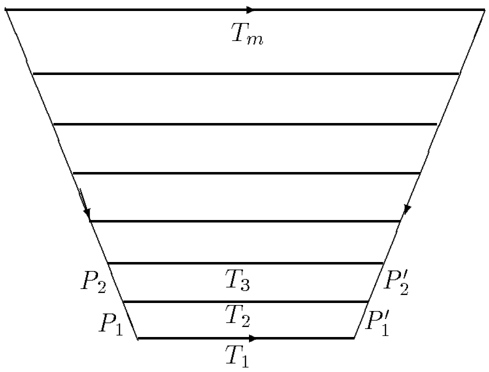

FIGURE 1.

- $[q \to rq]$, for every $r \in \bar{\mathcal{E}}$.

The first set of rules allows the state letter to move freely along a word of the form uqv, where u, v are words from the free group on $A \cup Y$. Such words are called *admissible words* for $\mathcal{M}$.

The second set of rules allows us to insert any relation from $\bar{\mathcal{E}}$ into the word (we are also allowed to insert and remove subwords of the form aa^{-1} because $A \cup Y$ is a generating set of a free group).

It is easy to see that a word qu, where u is a word over A, is recognized by this machine (that is, the machine takes it to the word q) if and only if $u = 1$ in the group $\bar{\mathcal{G}}$, that is, if and only $u = 1$ in $\mathcal{G}$ (because $\mathcal{G}$ is embedded into $\bar{\mathcal{G}}$).

Now we present the ideas of a Higman embedding based on $\mathcal{M}$. Let $T_1, T_2, \ldots, T_m = W_0$ be an accepting *computation* of $\mathcal{M}$, that is, a sequence of admissible words such that T_{i+1} is obtained from T_i by applying an S-rule of $\mathcal{M}$, the last word in this sequence being equal to q in the free group.

Suppose that some words P_i and P'_i over a certain alphabet (it is included in the finite generating set of the group we are constructing) correspond to this S-rule, and modulo some set of group relations Z we have, for every $i = 1, \ldots, m-1$,

$$P_i T_i = T_{i+1} P'_i\,,$$

that is, we have a van Kampen diagram with the boundary label $P_i T_i (P'_i)^{-1} T_{i+1}^{-1}$. Then $PT_1 = W_0 P'$, where $P = P_{m-1} \ldots P_2 P_1$, $P'_i = P'_{m-1} \ldots P'_2 P'_1$. The corresponding van Kampen diagram Δ has the form of a *trapezoid* (Figure 1).

For example, if we use the S-machine $\mathcal{M}$ described above, then the presentation Z is easy to describe: for every rule $\tau = [q \to uqv]$ we have the following relations in Z: $\theta_\tau^{-1} q \theta_\tau = uqv$, $\theta_\tau a = a\theta_\tau$ for every $a \in A \cup Y$ (the letters θ_τ corresponding to the rules of $\mathcal{M}$ are added to the generating set). In this case the words P_i, P'_i are of length 1, and are equal to θ_τ.

The words P and P' contain the *history* of our computation, because P_i and P'_i correspond to S-rules applied during the computation. The words T_1 and T_m are labels of the bottom and the top of the trapezoid.

If the set of relations Z is chosen carefully (see [SBR], [BORS], [OlSa01]), then the converse is also true. Thus the word $PT_1(P')^{-1}W_0^{-1}$ is written on the boundary of a trapezoid if and only if the word T_1 is accepted.

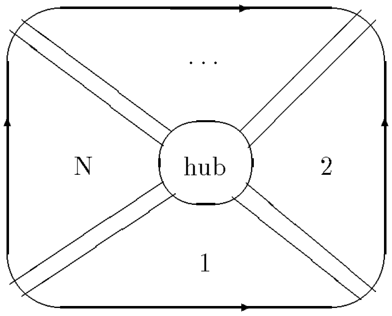

FIGURE 2.

The next step is to consider $N \geq 1$ copies of our trapezoid with labels taken from different alphabets. For technical reasons (similar to the hyperbolic graphs argument in [SBR], [Ol2], [BORS]) we need N to be even and large enough (say, $N \geq 8$). Let us choose disjoint alphabets in different copies of the trapezoid. Then we can glue two copies of the trapezoid by using a special letter k which conjugates letters on the right side of the first trapezoid with letters on the left side of the mirror image of the other copy. Notice that if P and P' are copies of each other then we can also glue the right side of one trapezoid to the left side of the other without taking mirror images. Of course the conjugacy relations involving letter k should be added into the set of group relations Z.

Suppose that T_m is equal to q, i.e., the word T_1 is *accepted* by the machine. If we connect the first copy of Δ with the second copy, then with the third copy, and finally connect the Nth copy with the first copy, using different letters $k_2, \ldots, k_N, k_1$, we get an *annular* diagram, a *ring*. The outer boundary of this ring has label $\mathcal{K}(T_1)$ of the form $k_1 T_1' k_2^{-1} (T_1'')^{-1} \ldots$ (here we use the fact that N is even). The words $T_1', \ldots, T_1^{(N)}$ are different copies of T_1. The inner boundary has label $k_1 q' k_2^{-1} (q'')^{-1} \ldots$ (this word is called the *hub*) (Figure 2 shows the diagram in the case of $N = 4$).

We add the hub to the list of defining relations Z. Now with every word T of the form uqv, where u, v are words in $A \cup Y$, we have associated the word $\mathcal{K}(T)$, and we see that if T is accepted, then $\mathcal{K}(T)$ is equal to 1 modulo Z. Again, if Z is chosen carefully, then the converse is also true: if $\mathcal{K}(T)$ is equal to 1 modulo Z then T is accepted by $\mathcal{S}$. Thus we have an interpretation of $\mathcal{S}$ in the group generated by the tape alphabet and the state alphabet of $\mathcal{S}$, the set Θ, and the set $\{k_1, \ldots, k_N\}$, subject to the relations from Z.

The diagram on Figure 2 is called a *disk*. The trapezoids forming this disk are numbered in the natural order (from 1 to N).

We assume that the copy of the alphabet A used in the first subtrapezoid of a disk coincides with A itself (the generating set of $\mathcal{G}$).

Denote the group given by the presentation Z we have got so far by $G(\mathcal{M})$.

There are several ways to use the group $G(\mathcal{M})$ to embed $\mathcal{G}$ into a finitely presented group. We use a method described in [OlSa01].

Consider $N - 1$ new copies of the trapezoid Δ. Number these trapezoids by $2', \ldots, N'$. Identify the copy of A (but not Y) and the state letter q in the trapezoid number i' with the copy of A and the copy of q in the "old" trapezoid number i. Now glue the trapezoid number $2'$ with the trapezoid number $3'$ using the letter k_3 (as before), then glue in the trapezoid number $4'$ and so on, but glue the trapezoid number N' with the trapezoid number $2'$ using $k_1qk_2^{-1}$ (that will be possible because, as it turns out, in that case the words P, P' will be copies of each other). Thus, conjugation of the letters on the side of the trapezoid number N' by $k_1qk_2^{-1}$ gives the letters of the side of the mirror image of the trapezoid number $2'$. Let us add the relations used in the new trapezoids and the conjugation relations (involving k_i) to Z.

The resulting picture is an annular diagram. The inner boundary of it is labelled by the hub. If we add the hub to the diagram, we get a van Kampen diagram which also looks like a disk but has $N-1$ sectors. The outer boundary of this new disk is labelled by the word $k_1qk_2^{-1}V$, where V is the suffix of $\mathcal{K}(qu)$ staying after k_2^{-1}. Thus the word $k_1qk_2^{-1}V$ and the word $k_1quk_2^{-1}V$ are both equal to 1 modulo Z, so Z implies $u=1$. We denote by $\mathcal{H}$ the group given by the set of relations Z that we have constructed.

We have proved that the identity map on A can be extended to a homomorphism from $\mathcal{G}$ to the subgroup $\langle A\rangle$ of the group $\mathcal{H}$ given by the set of relations Z. It is possible to show that this homomorphism is injective, so $\mathcal{G}$ is embedded into $\mathcal{H}$.

Unfortunately, even if $\mathcal{G}$ has solvable conjugacy problem, $\mathcal{H}$ may have unsolvable conjugacy problem. In fact the difficulties are similar to those described in [Col]. Let us give just one example.

Consider two pairs of words u_1, u_2 and v_1, v_2 over the alphabet A (we can view these words as elements of $\mathcal{G}$). Let q be the first state letter in $\mathcal{K}(qu)$. Consider the words $W_1 = q^{-1}u_1qu_2q^{-1}$ and $W_2 = q^{-1}v_1qv_2q^{-1}$. Suppose that there exists a word $W(A)$ in the alphabet A such that $W(A)u_1W(A)^{-1} = v_1, W(A)u_2W(A)^{-1} = v_2$. Let $W(A) = a_1a_2\dots a_k$, $a_i \in A$. For every $a \in A$, let $\theta(a)$ correspond to the S-rule of the form $[q \to a^{-1}qa]$. Consider the word $W(\Theta) = \theta(a_1)\theta(a_2)\dots\theta(a_k)$. Then it is easy to see that $W(\theta)W_1W(\theta^{-1}) = W_2$. Thus if the pair (u_1, u_2) is conjugate to the pair (v_1, v_2) in $\mathcal{G}$, then the words W_1 and W_2 are conjugate in $\mathcal{G}$. One can also prove that the converse statement is true. In this example, pairs of words can be obviously replaced by any t-tuples of words, $t \geq 2$. Therefore, if the conjugacy problem is solvable in $\mathcal{G}$, then the conjugacy of t-tuples of elements in $\mathcal{G}$ is solvable. It is known [Col] that there exists a finitely generated group $\mathcal{G}$ with solvable conjugacy problem and unsolvable problem of conjugacy for sequences of elements. For such a group $\mathcal{G}$ the group $\mathcal{H}$ has undecidable conjugacy problem.

Fortunately, since S-machines are easier to simulate in groups than other types of Turing machines, we can overcome this difficulty (and many others) by modifying our machine $\mathcal{M}$ (see below).

It is important to mention that we cannot allow all parts of the admissible words to be arbitrary group words. To avoid long computations without inserting new relation from $\bar{\mathcal{E}}$, we had to require that certain subwords of admissible words stay positive during the computation. This is a major difference between the S-machine used in the proof of Theorem 1 and the S-machine used in our previous papers.

In order to keep these subwords positive we use a trick that goes back to Novikov, Boone and Higman (see Rotman [Rot]). They used a special letter x and Baumslag-Solitar relations of the form $a^{-1}xa = x^2$ for every a in the tape alphabet of the machine. The idea is that if we have relations $a^{-1}xa = x^2$ for all $a \in A$ and a word $u^{-1}xu$, where u is a reduced word in A, is equal modulo these relations to a power of x, then the first letter of u is positive. Notice that one problem with using the x-letters and Baumslag-Solitar relations is that, since $N-1$ is odd, we cannot use x-letters in the second disk described above (indeed, otherwise it would be impossible to glue the $N-1$ sectors together since the words P and P' would not be copies of each other). Thus we need to consider two S-machines (one responsible for the first disk, the other for the second disk) which are very similar but have different "hardware": the admissible words of the first machine must have positive parts described above, and the admissible words of the second machine do not have to satisfy this restriction.

Now let us briefly describe the strategy of the proof that the conjugacy problem in $\mathcal{H}$ is solvable. As in our previous papers, the main tool in this paper is bands (other people call them "strips", "corridors", etc.).

In terms of annular (Schupp) diagrams, our task is the following: given two words u and v in generators of $\mathcal{H}$, find out if there exists an annular diagram over the presentation of $\mathcal{H}$ with boundary labels u and v. It turns out that any annular diagram over $\mathcal{H}$ (after removing some parts of recursively bounded size) becomes a diagram of one of three main types: a *ring* (where the boundary labels contain k-letters but do not contain θ-letters, where k-letters and θ-letters belong to the alphabets $\mathcal{K}$ and $\Theta\cup\bar\Theta$, respectively, defined below, and all maximal θ-bands are annuli surrounding the hole of the diagram), a *roll* (where the boundary label does not contain k-letters but contains θ-letters; all maximal k-bands are annuli surrounding the hole), and a *spiral* (where the boundary labels contain both k-letters and θ-letters; both the k-bands and the θ-bands start and end on different boundary components, and each θ-band crosses each k-band many times. Different methods are used to treat different cases. Roughly speaking, the study of rings amounts to studying the lengths of computations of our S-machines, the study of rolls amounts to the study of the space complexity (how much space is needed by the machines during a computation), and the study of spirals amounts to the study of computations with periodic history. The x-letters and the Baumslag-Solitar relations allow us to treat the case of rings, but they cause the main technical difficulties in the cases of rolls and spirals.

Remark 1. Notice that even though the conjugacy problem in $\mathcal{H}$ is solvable provided it is solvable in $\mathcal{G}$, the computational complexity of the conjugacy problem in $\mathcal{H}$ can be higher than in $\mathcal{G}$. In particular, the algorithm solving the conjugacy problem in $\mathcal{H}$ uses Makanin's algorithm [Mak] for solving systems of equations in the free group. Thus we do not know whether it is possible to embed any finitely generated group with conjugacy problem, say, in NP into a finitely presented group with conjugacy problem in NP.

Now let us give a precise description of the S-machines $\mathcal{S}$ and $\bar{\mathcal{S}}$ and the group $\mathcal{H}$ simulating these machines.

Fix an even number $N\ge 8$. Let $A=\{a_1,\dots,a_m\}$, $A\cup Y=\{a_1,\dots,a_{\bar m}\}$. The set $\mathcal{K}$ of state letters of $\mathcal{S}$ consists of letters $z_j(r,i)$, where $z\in\{K,L,P,R\}$, $j=1,\dots,N$, $r\in\bar{\mathcal{E}}$, $i\in\{1,2,3,4,5\}$.

We also define the set of *basic* letters $\tilde{\mathcal{K}}$ which consists of letters z_j, where $z\in\{K,L,P,R\}$, $j=1,\dots,N$, and their inverses. If z is a basic letter, $r\in\bar{\mathcal{E}}$, $i\in\{1,2,3,4,5\}$, then $z(r,i)\in\mathcal{K}$. If U is a word in $\tilde{\mathcal{K}}$ and other letters, $r\in\bar{\mathcal{E}}$, $i\in\{1,2,3,4,5\}$, then $U(r,i)$ is a word obtained from U by replacing every letter

$z \in \tilde{\mathcal{K}}$ by $z(r,i)$. The parameters r and i in the letter $z(r,i)$ or in the word $U(r,i)$ will be called the $\bar{\mathcal{E}}$-coordinate and the Ω-coordinate of the word.

The set $\mathcal{A}^{\pm 1}$ of *tape letters* of the machine $\mathcal{S}$ consists of letters $a_i(z)^{\pm 1}$, where $i = 1, \dots, \bar{m}$, $z \in \tilde{\mathcal{K}}$. For every $z \in \tilde{\mathcal{K}}$ we define $\mathcal{A}(z)$ as the set of all $a_i(z)$, $i = 1, \dots, \bar{m}$.

Let $\tilde{\Sigma}$ be the following word (considered as a cyclic word):

$$K_1 L_1 P_1 R_1 K_2^{-1} R_2^{-1} P_2^{-1} L_2^{-1} K_3 L_3 P_3 R_3 K_4^{-1} \dots K_N^{-1} R_N^{-1} P_N^{-1} L_N^{-1}.$$

Notice that for every basic letter z precisely one of z and z^{-1} occurs in the word $\tilde{\Sigma}$. The word $\Sigma = \tilde{\Sigma}(\emptyset, 1)$ will be called the *hub*.

For every $z \in \tilde{\mathcal{K}} \cup \tilde{\mathcal{K}}^{-1}$ we denote by z_- the letter immediately preceding z in the cyclic word $\tilde{\Sigma}$ or in $\tilde{\Sigma}^{-1}$. If $z' = z_-$ then we set $z = z'_+$. Similarly we define z_- and z_+ for $z \in \mathcal{K}$.

The language of *admissible words* of the machine $\mathcal{S}$ consists of all reduced words of the form $W \equiv y_1 u_1 y_2 u_2 \dots y_t u_t y_{t+1}$, where $y_1, \dots, y_{t+1} \in \mathcal{K}^{\pm 1}$, u_i are words in $\mathcal{A}(y_i)$, $i = 1, 2, \dots, t$, and for every $i = 1, 2, \dots, t$, either $y_{i+1} \equiv (y_i)_+$ or $y_{i+1} \equiv y_i^{-1}$. (Here $\equiv$ is the letter-for-letter, or graphical, equality of words.) The projection of $y_1 \dots y_{t+1}$ onto $\tilde{\mathcal{K}}$ is called the *base* of the admissible word W. The subword $y_i u_i y_{i+1}$ is called the $y_i y_{i+1}$-*sector* of the admissible word W, $i = 1, 2, \dots$. The word u_i is called the *inner part* of the $y_i y_{i+1}$-sector. We assume that the inner part of any zL_j-, zP_j- or $R_j z$-sector of W and W^{-1} is a positive word ($z \in \tilde{\mathcal{K}}$), that is, it does not contain a^{-1} for any $a \in \mathcal{A}$.

All rules of $\mathcal{S}$ have the form $[k_1 \to v_1 k'_1 u_1, \dots, k_n \to v_n k'_n u_n]$, where the projections k_i and k'_i on $\tilde{\mathcal{K}}$ will always be the same, and the $\bar{\mathcal{E}}$-coordinates and the Ω-coordinates of all state letters $k_1, \dots, k_n$ (resp. $k'_1, \dots, k'_n$) will be the same. In addition, each u_i (resp. v_i) will contain only letters from $\mathcal{A}(k_i)$ (resp. $\mathcal{A}((k_i)_-)$), $i = 1, \dots, n$.

For every word W in the alphabet $\{a_1, \dots, a_{\bar{m}}\}$ and every $k \in \tilde{\mathcal{K}}$, $W(k)$ will denote the word obtained from W by substitution $a_i \to a_i(k)$, $i = 1, \dots, \bar{m}$.

In order to simplify notation, we shall write a rule τ of the S-machines $\mathcal{S}$ in the form

$$[k_1 \to v_1 k_1 u_1, \dots, k_n \to v_n k_n u_n; r \to r', \omega \to \omega'],$$

where $k_i \in \tilde{\mathcal{K}}$, v_i are words in $\mathcal{A}((k_i)_-)$, and u_i are words in $\mathcal{A}(k_i)$. We shall also omit k in $u(k)$ if the value of k is obvious from the context.

Some of the arrows in a rule τ can have the form $k_i \xrightarrow{\ell}$. This means that the rule τ can be applied to an admissible word W only if the inner part of $k_i k'_i$-sectors and $k_{i''}(k_i)_+$-sectors in W are empty words and $k'_i = (k_i)_+$, $k_{i''} = k_i$ (i.e., $k'_i \neq k_i^{-1}$ and $k_{i''} \neq (k_i)_+^{-1}$). In that case $u(k)$ and $v(k)$ must be empty.

Let us expand the definition of $u_\tau(k), v_\tau(k_-)$ to the base letters not occurring in the rule by assuming that in this case $u_\tau(k)$ and $v_\tau(k_-)$ are empty.

By definition, to *apply* the rule $\tau = [z_1 \to v_1 z_1 u_1, \dots, z_s \to v_s z_s u_s; r \to r', \omega \to \omega']$ to an admissible word $W = y_1 w_1 y_2 \dots w_k y_{k+1}$ means to replace each $y_i(r, \omega)$ with $v_i y_i(r', \omega') u_i$, reduce the resulting word, and trim the prefix $v_\tau((y_1)_-)$ from the beginning and the suffix $u_\tau(y_{k+1})$ from the end of the resulting word. The rule τ is called *applicable* to W if:

(1) for every $z \in \tilde{\mathcal{K}}$ such that τ locks zz_+-sectors, the inner parts of zz_+-sectors in W are empty words, and W does not have zz^{-1}-sectors or $z_+^{-1} z_+$-sectors;

(2) the resulting word W' is again admissible (that is, the inner parts of all sectors which are supposed to be negative or positive are such).

The rules from $\mathcal{S}^+ \subset \mathcal{S}$, described below, will be called *positive*, the inverses of these rules will be called *negative*, and each rule of $\mathcal{S}$ will be either positive or negative.

The set $\mathcal{S}^+$ consists of 10 subsets $\mathcal{S}^+(\omega)$, $\omega \in \{1, 12, 2, 23, 3, 34, 4, 45, 5, 51\}$.

The set $\mathcal{S}^+(1)$ consists of rules $\tau(1, \emptyset, i)$, where $i = 1, \dots, \bar{m}$. The rule $\tau(1, \emptyset, i)$ has the form

$$[(L_j)_- \xrightarrow{\ell} (L_j)_-, P_j \to a_i P_j a_i^{-1}, R_j \xrightarrow{\ell} R_j, j = 1, \dots, N; \emptyset \to \emptyset, 1 \to 1].$$

The meaning of this set of rules is that the state letter P_j can freely move (if the $(\bar{\mathcal{E}}, \Omega)$-coordinates are $(\emptyset, 1)$, and L_j-letters and R_j-letters stay next to K_j-letters).

The set $\mathcal{S}^+(12)$ consists of one rule $\tau(12, r)$ for each $r \in \bar{\mathcal{E}} \backslash \{\emptyset\}$:

$$[(L_j)_- \xrightarrow{\ell} (L_j)_-, R_j \xrightarrow{\ell} R_j, j = 1, \dots, N; \emptyset \to r, 1 \to 2].$$

This rule does not insert any tape letters; it simply changes the $\bar{\mathcal{E}}$- and Ω-coordinates of state letters. This rule prepares the machine for step 2.

The set $\mathcal{S}^+(2)$ consists of rules $\tau(2, r, i)$, where $r \in \bar{\mathcal{E}}$, $i = 1, \dots, \bar{m}$:

$$\tau(2, r, i) = [L_j \to a_i L_j a_i^{-1}, R_j \xrightarrow{\ell} R_j, j = 1, \dots, N; r \to r, 2 \to 2].$$

The meaning of these rules is that they allow the state letters L_j to move freely while R_j stays next to $(R_j)_+$.

The set $\mathcal{S}^+(23)$ consists of one rule for each $r \in \bar{\mathcal{E}}$:

$$\tau(23, r) = [L_j \xrightarrow{\ell} L_j, R_j \xrightarrow{\ell} R_j; r \to r, 2 \to 3].$$

The meaning of this rule is that the machine can start step 3 when L_j and P_j meet (that is when the inner parts of $L_j z$-sectors are empty).

The set $\mathcal{S}^+(3)$ consists of one rule for each $r \in \bar{\mathcal{E}}$ and each i from 1 to $\bar{m}$:

$$\tau(3, r, i) = [L_j \xrightarrow{\ell} L_j, R_j \to a_i^{-1} R_j a_i, j = 1, \dots, N; r \to r, 3 \to 3].$$

These rules allow the state letter R_j to move freely between P_j and $(R_j)_+$.

The set $\mathcal{S}^+(34)$ consists of one rule for each nonempty $r \in \bar{\mathcal{E}}$:

$$\tau(34, r) = [L_j \xrightarrow{\ell} r L_j, P_j \xrightarrow{\ell} P_j, j = 1, \dots, N; r \to r, 3 \to 4].$$

This rule can be applied when the state letters L_j, P_j, R_j meet together; it inserts r to the left of the state letters L_j, and prepares the machine for step 4.

The set $\mathcal{S}^+(4)$ consists of rules $\tau(4, r, i)$, $r \in \bar{\mathcal{E}}, i = 1, \dots, \bar{m}$:

$$\tau(4, r, i) = [L_j \to a_i L_j a_i^{-1}, P_j \xrightarrow{\ell} P_j, j = 1, \dots, N; r \to r, 4 \to 4].$$

These rules allow the state letter L_j to move freely between $(L_j)_-$ and P_j.

The set $\mathcal{S}^+(45)$ consists of one rule for each $r \in \bar{\mathcal{E}}$:

$$\tau(45, r) = [(L_j)_- \xrightarrow{\ell} (L_j)_-, P_j \xrightarrow{\ell} P_j, j = 1, \dots, N; r \to r, 4 \to 5].$$

The machine can start step 5 when L_j and $(L_j)_-$ meet.

The set $\mathcal{S}^+(5)$ consists of one rule for each $r \in \bar{\mathcal{E}}, i = 1, \dots, \bar{m}$:

$$\tau(5, r, i) = [(L_j)_- \xrightarrow{\ell} (L_j)_-, R_j \to a_i^{-1} R_j a_i, j = 1, \dots, N; r \to r, 5 \to 5].$$

These rules allow R_j move freely between P_j and $(R_j)_+$.

Finally the set $\mathcal{S}^+(51)$ consists of one rule for each $r \in \bar{\mathcal{E}}$:

$$\tau(51, r) = [(L_j)_- \xrightarrow{\ell} (L_j)_-, R_j \xrightarrow{\ell} R_j, j = 1, \dots, N; r \to \emptyset, 5 \to 1].$$

The cycle is complete, the machine can start step 1 again when $(L_j)_-$ meets L_j, and R_j meets $(R_j)_+$.

The S-machine $\bar{\mathcal{S}}$ is similar to $\mathcal{S}$, so we only present the differences between $\mathcal{S}$ and $\bar{\mathcal{S}}$.

We introduce disjoint copies of sets $\mathcal{K}$, $\tilde{\mathcal{K}}$, $\mathcal{A}$, and denote them by $\bar{\mathcal{K}}$, $\bar{\tilde{\mathcal{K}}}$, $\bar{\mathcal{A}}$, respectively. In order to make $\mathcal{S}$ and $\bar{\mathcal{S}}$ "communicate", we identify $z_j(\emptyset, 1)$ with $\bar{z}_j(\emptyset, 1)$, $z \in \{K, L, P, R\}$, $j = 1, \dots, N$, and $a_i(P_j)$ with $\bar{a}_i(\bar{P}_j)$, $i = 1, \dots, m$, $j = 1, \dots, N$. Notice that we do not identify $a_{m+1}, \dots, a_{\bar{m}}$ with their "bar"-brothers $\bar{a}_{m+1}, \dots, \bar{a}_{\bar{m}}$, and we do not identify $a_i(z)$ with $\bar{a}_i(z)$ for $z \neq P_j$. Notice also that this identification of letters makes the word $\bar{\Sigma}$ coincide with the hub Σ.

Admissible words of $\bar{\mathcal{S}}$ have the same form as admissible words of $\mathcal{S}$ except for the following differences:

- All letters are replaced by their "bar"-brothers.
- We drop the restriction that certain parts of admissible words are positive (negative).
- The $\bar{K}_1 z$-, $\bar{L}_1 z$-, $\bar{P}_1 z$- and $\bar{R}_1 z$-sectors must be empty ($z \in \bar{\mathcal{K}}$). Thus in every admissible word of $\bar{\mathcal{S}}$ the part between $\bar{K}_1$ and $\bar{K}_2^{-1}$ contains no $\bar{\mathcal{A}}$-letters.

The rules of $\bar{\mathcal{S}}$ are obtained from the corresponding rules of $\mathcal{S}$ by replacing every letter by its "bar"-brother and by removing letters from $\bar{\mathcal{A}}(\bar{z}_1)$, $z \in \{K, L, P, R\}$. Thus rules of $\bar{\mathcal{S}}$ do not insert any $\bar{\mathcal{A}}$-letters in the interval between $\bar{K}_1$ and $\bar{K}_2^{-1}$, and work in other parts of the admissible words just as $\mathcal{S}$.

In order to convert the machine $\mathcal{S}$ into a list of defining relations, we need to introduce one more set of letters $\mathcal{X} = \{x(a, \tau) \mid a \in \mathcal{A} \backslash \bigcup_j \mathcal{A}(P_j), \tau \in \mathcal{S}^+\}$.

For every $\tau \in \mathcal{S}$, we also consider a map α_τ from $\mathcal{A}$ to the free group generated by $\mathcal{A}$ and $\mathcal{X} \cup \mathcal{X}^{-1}$. Let $\tau \in \mathcal{S}^+$, $a \in \mathcal{A}(z), z \in \tilde{\mathcal{K}}$. Then

$$\alpha_\tau(a) = \begin{cases} x(a, \tau)a & \text{if } z = K_j, \ j = 1, \dots, N, \\ x(a, \tau)a & \text{if } z = L_j, \ j = 1, \dots, N, \\ a & \text{if } z = P_j, \ j = 1, \dots, N, \\ ax(a, \tau) & \text{if } z = R_j, \ j = 1, \dots, N. \end{cases}$$

The definition of $\alpha_{\tau^{-1}}$, $\tau \in \mathcal{S}^+$, is obtained from the definition of α_τ by replacing all $\mathcal{X}$-letters with their inverses.

We expand these maps to all words in the alphabet $\mathcal{A} \cup \mathcal{A}^{-1} \cup \mathcal{K} \cup \mathcal{K}^{-1}$ in the natural way (mapping letters from $\mathcal{K} \cup \mathcal{K}^{-1}$ to themselves).

Also with every $\tau \in \mathcal{S}^+$, we associate a set of letters

$$\Theta(\tau) = \{\theta(\tau, z) \mid z \in \{K_j, L_j, P_j, R_j\}, \ j = 1, \dots, N\}.$$

Let $\Theta(z) = \{\theta(\tau, z) \mid \tau \in \mathcal{S}^+\}$, $z \in \{K_j, L_j, P_j, R_j\}$. Finally the union of all $\Theta(\tau)$ is denoted by Θ.

Let $\tau = [U_1 \to V_1, \dots, U_k \to V_k; r \to r', l \to l']$ be one of the rules of $\mathcal{S}^+$. Let $u(z), v(z)$, $z \in \tilde{\mathcal{K}}$, be the words associated with this rule as above.

Then we associate with τ the following list of *main relations*:

$$\theta(\tau, z_-)^{-1} z(r, l) \theta(\tau, z) = \alpha_{\tau^{-1}}(v_i) z(r', l') \alpha_{\tau^{-1}}(u_i), \ i = 1, \dots, k.$$

Let $z \in \{K_j, L_j, P_j, R_j\}$, $j = 1, \dots, N$, $a, b \in \mathcal{A}(z)$. Then we add *auxiliary* Θ-*relations*:

$$\theta(\tau, z)^{-1}\alpha_\tau(a)\theta(\tau, z) = \alpha_{\tau^{-1}}(a) \text{ if } \tau \text{ does not lock } thezz_+\text{-sector}$$

$$\begin{array}{ll} ax(b,\tau)a^{-1} = x(b,\tau)^4 & \text{if } z = K_j \text{ or } z = L_j, \\ a^{-1}x(b,\tau)a = x(b,\tau)^4 & \text{if } z = R_j. \end{array}$$

Finally let b' be the "brother" of b in the alphabet $\mathcal{A}(z_-)$ if $z = K_j$ or $z = L_j$. Then we need auxiliary (k, x)-*relations*:

$$\begin{array}{ll} z(r,i)x(b,\tau) = x(b',\tau)z(r,i) & \text{if } z = K_j, \\ z(r,i)x(b,\tau) = x(b',\tau)^4 z(r,i) & \text{if } z = L_j. \end{array}$$

Let $\mathcal{R}(\mathcal{S})$ be the set of all relations corresponding to the set of S-rules $\mathcal{S}^+$.

To convert $\bar{\mathcal{S}}$ into a set of relations we do not need $\mathcal{X}$-letters, but we need "bar"-brothers of the letters from Θ. Denote this set by $\bar{\Theta}$. The easiest way to explain how to convert an S-rule of $\bar{\mathcal{S}}$ to a set of relations is the following: convert it above, then add bars to all letters, replace all letters from $\mathcal{X}$ and from $\bigcup \bar{A}(z_1)$, $z \in \{K, L, P, R\}$ by 1. Let $\mathcal{R}(\bar{\mathcal{S}})$ be the set of all relations corresponding to $\bar{\mathcal{S}}^+$.

Finally the group $\mathcal{H}$ is given by the set of generators $\mathcal{K} \cup \bar{\mathcal{K}} \cup \mathcal{A} \cup \bar{\mathcal{A}} \cup \Theta \cup \bar{\Theta} \cup \mathcal{X}$ and the set of relations $\mathcal{R} = \mathcal{R}(\mathcal{S}) \cup \mathcal{R}(\bar{\mathcal{S}}) \cup \{\Sigma\}$.

The complete proof of Theorem 1 can be found in [OlSa02].

References

[BORS] J. C. Birget, A. Yu. Ol'shanskii, E. Rips, M. V. Sapir. Isoperimetric functions of groups and computational complexity of the word problem. Annals of Mathematics, 156, 2 (2002), 467–518.

[Cla] C. R. J. Clapham. An embedding theorem for finitely generated groups, Proc. London. Math. Soc. (3), 17 (1967), 419–430. MR **36:**5199

[Col] Donald J. Collins. Conjugacy and the Higman embedding theorem. Word problems, II (Conf. on Decision Problems in Algebra, Oxford, 1976), pp. 81–85, Stud. Logic Foundations Math., 95, North-Holland, Amsterdam-New York, 1980. MR **81m:**20051

[CM] D. J. Collins, C. F. Miller III. The conjugacy problem and subgroups of finite index. Proc. London Math. Soc. (3) 34 (1977), no. 3, 535–556. MR **55:**8187

[GK] A. V. Gorjaga, A. S. Kirkinskiĭ. The decidability of the conjugacy problem cannot be transferred to finite extensions of groups. Algebra i Logika 14 (1975), no. 4, 393–406. (Russian) MR **54:**2813

[Hi] G. Higman. Subgroups of finitely presented groups. Proc. Roy. Soc. Ser. A, 262 (1961), 455–475. MR **24:**A152

[KT] *Kourovka Notebook.* Unsolved Problems in Group Theory. 5th edition, Novosibirsk, 1976.

[Mak] G. S. Makanin. Equations in a free group. Izv. Akad. Nauk SSSR Ser. Mat. 46 (1982), no. 6, 1199–1273, 1344; English transl., Math. USSR-Izv. 21 (1983), 546–582. MR **84m:**20040

[Ma] Yu. I. Manin. The computable and the noncomputable, "Sovetskoe Radio", Moscow, 1980 MR **82i:**03002

[Mil] Charles F. Miller III. On group-theoretic decision problems and their classification. Annals of Mathematics Studies, No. 68. Princeton University Press, Princeton, N.J.; University of Tokyo Press, Tokyo, 1971. MR **46:**9147

[Ol2] A. Yu. Ol'shanskii. On distortion of subgroups in finitely presented groups. Mat. Sb. 188 (1997), no. 11, 51–98; English transl., Sb. Math. 188 (1997), no. 11, 1617–1664. MR **99a:**20038

[OlSa01] A. Yu. Ol'shanskii, M. V. Sapir. Length and area functions on groups and quasi-isometric Higman embeddings. Internat. J. Algebra Comput. 11 (2001), no. 2, 137–170. MR **2002b:**20058

[OlSa02] A. Yu. Olshanskii, M. V. Sapir. The Conjugacy Problem and Higman Embeddings. Preprint arXiv:math.GR/0212227.

[Rot] J. Rotman. *An Introduction to the Theory of Groups.* Allyn & Bacon, third edition, 1984. MR **85f:**20001

[SBR] M. V. Sapir, J. C. Birget, E. Rips. Isoperimetric and isodiametric functions of groups, Annals of Mathematics, 157, 2 (2002), 345–466.

[Va] M. K. Valiev. On polynomial reducibility of the word problem under embedding of recursively presented groups in finitely presented groups. Mathematical foundations of computer science 1975 (Fourth Sympos., Mariánské Lázně, 1975), pp. 432–438. Lecture Notes in Comput. Sci., Vol. 32, Springer, Berlin, 1975. MR **54:**413

Mathematics Department, Vanderbilt University, Nashville, Tennessee 37240, and Mechanics-Mathematics Department, Chair of Higher Algebra, Moscow State University, Moscow, Russia

E-mail address: `alexander.olshanskiy@vanderbilt.edu` `olshan@shabol.math.msu.su`

Mathematics Department, Vanderbilt University, Nashville, Tennessee 37240

E-mail address: `msapir@math.vanderbilt.edu`