Journal of Algebra 345 (2011) 324–342

Contents lists available at ScienceDirect

Journal of Algebra

www.elsevier.com/locate/jalgebra

The Word Problem in the Baumslag group with a non-elementary Dehn function is polynomial time decidable

Alexei Myasnikov[a,*,1], Alexander Ushakov[a,1], Dong Wook Won[b]

[a] Department of Mathematics, Stevens Institute of Technology, Hoboken, NJ, USA
[b] Department of Mathematics, CUNY/LAGCC, Long Island City, NY, USA

ARTICLE INFO

Article history:
Received 10 May 2011
Available online 25 August 2011
Communicated by Derek Holt

MSC:
20F10
11Y16

Keywords:
Word Problem
Baumslag group
One-relator group
Magnus breakdown algorithm
Power circuit
Computational complexity

ABSTRACT

We prove that the Word Problem in the Baumslag group $G_{(1,2)} = \langle a, b;\ a^{a^b} = a^2 \rangle$ which has a non-elementary Dehn function is decidable in polynomial time.

1. Introduction

One-relator groups form a very interesting and very mysterious class of groups. In 1910 Dehn proved that the Word Problem for the standard presentation of the fundamental group of a closed oriented surface of genus at least two is solvable by what is now called Dehn's algorithm (see [21] for details). In 1932 Magnus developed a general powerful approach to one-relator groups [19], nowadays known as Magnus break-down procedure (see [20,18]). In particular, he solved the Word Problem

[*] Corresponding author.
E-mail addresses: amiasnikov@gmail.com (A. Myasnikov), sasha.ushakov@gmail.com (A. Ushakov), dwwon@hotmail.com (D.W. Won).

[1] The work was partially supported by the NSF grant DMS-0914773.

0021-8693/$ – see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jalgebra.2011.07.024

(WP) in an arbitrary one-relator group. The decision algorithm is quite complicated and its time complexity is unknown. In fact, we show here that the time function of the Magnus decision algorithm on the Baumslag group

$$G_{(1,2)} = \langle a, b \mid b^{-1}a^{-1}bab^{-1}ab = a^2 \rangle$$

is not bounded by any finite tower of exponents. Furthermore, it is unknown whether there exists any feasible general (uniform) algorithm that solves WP in all one-relator groups, and at present it seems implausible that such algorithm exists. However, it is quite possible that the Word Problem in every fixed one-relator group is tractable. In the Magnus collection of open problems in groups theory [5] the following question is posted.

Problem 1.1. (See [5], (OR3).) Is it true that WP in every given one-relator group G is decidable in polynomial time?

The current state of affairs on WP in one-relator groups can be described as follows. On one hand, there are several large classes of one-relator groups where WP is well understood and is decidable in polynomial time (hyperbolic, automatic, linear, etc). On the other hand, there are several sporadic examples of one-relator groups where WP requires a special treatment, though at the end is polynomial time decidable. Finally, there are a few one-relator groups where WP seems especially hard and the time complexity is unknown. These are the most interesting ones in this context.

One of the principal unsolved mysteries on one-relator groups is which of them have a hard WP and why. More precisely, the problem is to determine the "general classes" of one-relator groups and divide the rest (the sporadic, exceptional ones) into some well-defined families.

There are several conjectures that describe large general classes of one-relator groups which we would like to mention here.

1.1. Hyperbolic groups

Notice, that if G is hyperbolic, in particular, if it satisfies the small cancelation condition $C'(\frac{1}{6})$, then WP in G is decidable in linear time by Dehn's algorithm [14]. Since the asymptotic density of the set of words $w \in F(X)$ for which the symmetrized one-relator presentation $\langle X \mid w \rangle$ is $C'(\frac{1}{6})$ small cancelation is equal to 1, one may say that for generic one-relator groups the answer to the question above is affirmative. One can check in polynomial (at most quadratic) time if a one-relator presentation, when symmetrized, is $C'(\frac{1}{6})$ or not. Hence, it is possible to run in parallel the Magnus break-down process and Dehn's algorithm for symmetrized $C'(\frac{1}{6})$ presentations and obtain a correct uniform total algorithm that solves WP in one-relator groups, and has Ptime complexity on the set of one-relator groups of asymptotic density 1. Unfortunately, such an algorithm will not be feasible on the most interesting examples of one-relator groups. Some interesting examples of hyperbolic one-relator groups can be found in [15].

Of course, not all one-relator groups are hyperbolic. The famous Baumslag–Solitar one-relator groups

$$B_{(m,n)} = \langle a, b \mid b^{-1}a^m b = a^n \rangle, \quad m, n \geqslant 1,$$

introduced in [6] are not hyperbolic, since the groups $B_{(1,n)}$ are infinite metabelian, and the other ones contain $F_2 \times \mathbb{Z}$ as a subgroup.

The following outstanding conjecture (see [5]) describes, if true, one-relator hyperbolic groups.

Problem 1.2. Is every one-relator group without Baumslag–Solitar subgroups hyperbolic?

Independently of the above, it is very interesting to know which one-relator groups contain groups $B_{(m,n)}$.

Problem 1.3. Is there an algorithm to recognize if a given one-relator group contains a subgroup $B_{(m,n)}$ for some $m, n \geqslant 1$?

Notice, that in 1968 B.B. Newman in [24] showed that all one-relator groups with torsion are hyperbolic and, hence, the Word Problem for them is decidable in linear time.

1.2. Automatic groups

Automatic groups form another class where WP is easy. It is known that every hyperbolic group is automatic and WP is decidable in at most quadratic time in a given automatic group. Furthermore, the Dehn function in automatic groups is quadratic. We refer to [11] for more details on automatic groups. Observe, that the group $B_{(m,n)}$ is not automatic provided $m \neq n$, since its Dehn function is exponential.

The main challenge in this area is to describe one-relator automatic groups. Answering the following questions would help to understand which one-relator groups are automatic.

Problem 1.4. Is it true that one-relator groups with a quadratic Dehn function are automatic?

Problem 1.5. Is it true that one-relator groups with no subgroups isomorphic to $B_{(m,n)}, m \neq n$ are automatic?

Problem 1.6. (See [5], (OR8).) Is the one-relator group $\langle X \mid [u, v]\rangle$ automatic for all words $u, v \in F(X)$?

1.3. Linear and residually finite groups

Lipton and Zalstein in [17] proved that WP in linear groups is polynomial time decidable, so one-relator linear groups provide a general subclass of one-relator groups where WP is easy. Until recently, not much was known about linearity of one-relator groups. We refer to [3] for an initial discussion that formed the area for years to come. The real breakthrough came in 2009 when Wise announced in [28] that if a hyperbolic group G has a quasi-convex hierarchy then it is virtually a subgroup of a right angled Artin group and, hence, is linear. This result covers a lot of one-relator groups, in particular all one-relator groups with torsion. There are two interesting cases that we would like to mention here. In [1] Baumslag introduced *cyclically pinched* one-relator groups as those ones that can be presented as a free product of free groups with cyclic amalgamation

$$\langle X \cup Y \mid u = v\rangle = F(X) *_{u=v} F(Y)$$

where $u \in F(X)$ and $v \in F(Y)$ are non-trivial non-primitive elements in the corresponding factors. Similarly, one can define *conjugacy pinched* one-relator groups as HNN extensions of free groups with cyclic associated subgroups:

$$\left\langle F(X), t \;\middle|\; t^{-1}ut = v\right\rangle.$$

Wehrfritz proved in [27] that if neither u nor v is a proper power in $F(X)$ then the group $F(X) *_{u=v} F(Y)$ is linear. However, it was shown in [7,16] that if either u or v is not a proper power then the group $F(X) *_{u=v} F(Y)$ is hyperbolic, so WP in these groups is linear time decidable. Similar results hold for conjugacy pinched one-relator groups as well. Observe, that cyclically and conjugacy pinched one-relator hyperbolic groups have quasi-convex hierarchy, so their linearity follows from Wise's result. On the other hand, WP in hyperbolic groups is easy anyway, so linearity in this case does not give much in terms of the efficiency of WP.

The general problem which one-relator non-hyperbolic groups are linear is wide open. Recall, that every finitely generated linear group is residually finite. Hence, to see that a given one-relator groups is not linear it suffices to show that it is not residually finite.

Notice that there is a special decision algorithm for WP in residually finite finitely presented groups. The algorithm when given such a group $\langle X \mid R\rangle$ and a word $w \in F(X)$ runs two procedures in parallel: the first one enumerates all the consequences of the relators R until the word w occurs, in which case $w = 1$ in G; while the second one checks if w is non-trivial in some finite quotient of G. Since R is finite and G is residually finite, one of the two procedures eventually stops and gives the solution of WP for w. However, this algorithm is extremely inefficient. This is why we do not discuss residually finite one-relator groups as a separate class here, but only briefly mention the results that are related to linearity.

Meskin in [22] studied residual finiteness of the following special class of one-relator groups:

$$B(u, v, m, n) = \left\langle X \;\middle|\; u^{-1}v^m u = v^n\right\rangle, \quad m, n \geqslant 1,$$

where u and v are arbitrary non-commuting elements in $F(X)$. He showed that if $m \neq 1$, $n \neq 1$, $m \neq n$ then the group $B(u, v, m, n)$ is not residually finite. It follows that the group $B_{(m,n)}$ is residually finite if and only if $m = 1$, or $n = 1$, or $m = n$.

Later Vol'vachev in [26] found linear representations for all residually finite groups $B(u, v, m, n)$. Sapir and Drutu constructed in [10] the first example of residually finite non-linear one-relator groups. They showed that the group

$$DS = \left\langle a, t \;\middle|\; t^{-2}at^2 = a^2\right\rangle$$

is residually finite and non-linear.

The general classes of one-relator groups described above are the only known ones where WP is polynomial time decidable. Now we describe the known sporadic one-relator groups where WP is presumably hard or requires a special approach.

1.4. Baumslag–Solitar groups

Gersten showed that the groups $B_{(m,n)}$, where $m \neq n$, have exponential Dehn functions [12] (see also [11] and [9]), so they are neither hyperbolic nor automatic. As we mentioned above the metabelian groups $B_{(1,n)}$ are linear, so WP in them is polynomial time decidable. The non-metabelian groups $B_{(m,n)}$ are not linear, so WP in them requires a special approach. Nevertheless, WP in these groups is polynomial time decidable (see Section 2). It would be interesting to study WP in the groups $B(u, v, m, n)$ which are similar to the Baumslag–Solitar groups.

Problem 1.7. What is complexity of WP in the groups $B(u, v, m, n)$?

1.5. Baumslag group $G_{(1,2)}$

The group $G_{(1,2)} = \langle a, b \mid b^{-1}a^{-1}bab^{-1}ab = a^2\rangle$ is truly remarkable. Baumslag introduced this group in [2] and showed that all its finite quotients are cyclic. In particular, the group $G_{(1,2)}$ is not residually finite and, hence, is not linear. In [13] Gersten showed that the Dehn function for $G_{(1,2)}$ is not elementary, since it has the lower bound $tower_2(\log_2(n))$ and later Platonov in [25] proved that $tower_2(\log_2(n))$ is exactly the Dehn function for $G_{(1,2)}$. This shows that $G_{(1,2)}$ is not hyperbolic, not automatic, not asynchronously automatic. It was conjectured by Gersten that $G_{(1,2)}$ has the highest Dehn function among all one-relator groups. As we have mentioned above the time function for the Magnus break-down algorithm on $G_{(1,2)}$ is not elementary. Taking this into account it was believed until recently that WP in $G_{(1,2)}$ is the hardest to solve among all one-relator groups. In this paper we show that the Word Problem for $G_{(1,2)}$ can be solved in polynomial time. To this end we develop a new technique to compress general exponential polynomials in the base 2 by algebraic circuits (straight-line programs) of a very special type, termed *power circuits* [23]. We showed that one can do many standard algebraic manipulations (operations $x + y, x - y, x \cdot 2^y, x \leqslant y$) over the values of exponential polynomials, whose standard binary length is not bounded by a fixed towers of exponents, in polynomial time if it is kept in the compressed form. This enables us to perform some variations of the standard algorithms in HNN extensions (or similar groups) keeping the actual rewriting in the compressed form. The resulting algorithms are of polynomial time, even though the standard versions are non-elementary.

1.6. Baumslag groups $G_{(m,n)}$

The approach outlined above is quite general and we believe it can be useful elsewhere. In particular, it works for groups of the type $G_{(m,n)}$, where m divides n. Here the groups $G_{(m,n)}$ are defined by the following presentations:

$$G_{(m,n)} = \langle a, b \mid b^{-1}a^{-1}ba^m b^{-1}ab = a^n \rangle.$$

Unfortunately, we do not have any compression techniques for the case when m does not divide n and n does not divide m. So the following problem seems currently as the main challenge regarding WP in one-relator groups.

Problem 1.8. What is the time-complexity of the Word Problem for $G_{(2,3)}$?

1.7. Generalized Baumslag groups

In [4] Baumslag, Miller and Troeger studied another series of one-relator groups $G(r, w)$ which are similar to the group $G_{(1,2)}$. Namely, if r, w are two non-commuting words in $F(X)$ then put

$$G(r, w) = \langle X \mid r^{r^w} = r^2 \rangle.$$

The group $G(r, w)$ is not residually finite (neither linear nor hyperbolic), it has precisely the same finite quotients as the group $\langle X \mid r \rangle$. These groups are surely among the ones with non-easy WP.

Problem 1.9. Let r, w be two non-commuting elements in $F(X)$.

1) What is the Dehn function of $G(r, w)$?
2) What is time complexity of WP in $G(r, w)$?

Going a bit further one can consider WP in the following groups

$$G(r, w, m, n) = \langle X \mid (r^m)^{r^w} = r^n \rangle.$$

The paper is organized as follows. In Section 2 we discuss algorithmic properties of elements of $G_{(1,2)}$ as an HNN extension of the Baumslag–Solitar group, set up the notation, and outline the difficulty of solvig the Word Problem using the standard methods for HNN extensions. In Section 3 we define the main tool in our method, namely the power circuits, and present techniques for working with them. In Section 4 we define a representation for words over some alphabet which we call a power sequence. In Section 5 we present the algorithm for solving the Word Problem in $G_{(1,2)}$ and prove that its time-complexity is $O(n^7)$.

2. The group $G_{(1,2)}$

In this section we represent the group $G_{(1,2)}$ as an HNN extension of the Baumslag–Solitar group $B_{(1,2)}$ and describe two rewriting systems $\mathcal{R}$ and $\mathcal{R}'$ to solve WP in $G_{(1,2)}$. The system $\mathcal{R}$ represents the classical Magnus breakdown algorithm for $G_{(1,2)}$. To study complexity of rewriting with $\mathcal{R}$ we construct an infinite sequence of words $\{w_k\}$ such that

- $|w_k| \leqslant 2^{k+2}$;
- it takes at least $tower_2(k-1)$ steps for $\mathcal{R}$ to rewrite w_k;
- $\mathcal{R}$ rewrites w_k into a unique word of length $tower_2(k)$.

This shows, in particular, that the time function of the Magnus breakdown algorithm on $G_{(1,2)}$ is not bounded by any finite tower of exponents. Our strategy to solve WP in $G_{(1,2)}$ can be roughly described as follows. We combine many elementary steps in rewriting by $\mathcal{R}$ into a single giant step and make it an elementary rewrite of a new system $\mathcal{R}'$. It is not hard to see that now it takes only polynomially many steps for $\mathcal{R}'$ to solve WP in $G_{(1,2)}$. In the rest of the paper we show that every elementary rewrite in $\mathcal{R}'$ (the giant step) can be done in polynomial time in the length of the input, thus proving that WP in $G_{(1,2)}$ is decidable in polynomial time.

2.1. HNN extensions

The purpose of this section is to introduce notations and the technique that we use throughout the paper.

Let H be a group with two isomorphic subgroups A and B, and $\varphi : A \to B$ an isomorphism. Then the group

$$G = \langle H, t \mid a^t = \varphi(a) \text{ for each } a \in A \rangle = \langle H, t \mid A^t = B \rangle$$

is called the HNN extension of H relative to φ. We refer to [18] for general facts on HNN extensions. The letter t is called the *stable letter*. If H is generated by a set Y then $Y \cup \{t\}$ generates G and any word w in the alphabet $(Y \cup \{t\})^{\pm 1}$ can be written in the syllable form:

$$w(H,t) = h_0 t^{\varepsilon_1} h_1 t^{\varepsilon_2} h_2 \dots t^{\varepsilon_n} h_n$$

where $\varepsilon_i = \pm 1$ for each $i = 1, \dots, n$ and h_i are words in the alphabet $Y^{\pm 1}$. The number n is called the *syllable length* of $w = w(H,t)$ and denoted by $|w|_t$. A *pinch* in w is a subword of the type $t^{-1}ht$ with $h \in A$ or a subword tht^{-1} where $h \in B$. A word w is *reduced* if it is freely reduced and contains no pinches.

Theorem *(Britton's lemma). (See [8].) Let $G = \langle H, t \mid A^t = B \rangle$. If a word*

$$w(H,t) = h_0 t^{\varepsilon_1} h_1 t^{\varepsilon_2} h_2 \dots t^{\varepsilon_n} h_n$$

represents the trivial element of G then either $n = 0$ and $w =_H 1$ or $w(H,t)$ has a pinch.

Corollary 2.1. *Let $G = \langle H, t \mid A^t = B \rangle$. Assume that:*

(G1) *The Word Problem is solvable in H.*
(G2) *The Membership Problem is solvable for A and B in H.*
(G3) *The isomorphisms φ and φ^{-1} are effectively computable.*

Then the Word Problem in G is solvable.

Proof. The decision algorithm that easily comes from Britton's lemma can be described as rewriting with the following infinite rewriting system $\mathcal{R}_{HNN}$:

$$\begin{aligned} &\{t^{-1}ht \to \phi(h) \mid h \in F(Y) \text{ and } h \in A\} \cup \{tht^{-1} \to \phi^{-1}(h) \mid h \in F(Y) \text{ and } h \in B\} \\ &\quad \cup \{h \to \varepsilon \mid h \in F(Y) \text{ and } h =_H 1\} \end{aligned} \tag{1}$$

where ε is the empty word. □

2.2. The group $G_{(1,2)}$ and Magnus breakdown

Proposition 2.2. *Let $G_{(1,2)} = \langle a, b \mid b^{-1}a^{-1}bab^{-1}ab = a^2\rangle$. Then the following hold:*

1) *The group $G_{(1,2)}$ is a conjugacy pinched HNN extension of the Baumslag–Solitar group $B_{(1,2)} = \langle a, t \mid t^{-1}at = a^2\rangle$ with the stable letter b:*

$$G_{(1,2)} = \langle B_{(1,2)}, b \mid b^{-1}ab = t\rangle.$$

2) *An infinite rewriting system $\mathcal{R}$:*

$$\begin{aligned}&\{t^{-1}a^kt \to a^{2k} \mid k \in \mathbb{Z}\} \cup \{ta^{2k}t^{-1} \to a^k \mid k \in \mathbb{Z}\}\\ &\quad\cup \{b^{-1}a^kb \to t^k \mid k \in \mathbb{Z}\} \cup \{bt^kb^{-1} \to a^k \mid k \in \mathbb{Z}\}\\ &\quad\cup \{aa^{-1} \to \varepsilon,\ a^{-1}a \to \varepsilon,\ bb^{-1} \to \varepsilon,\ b^{-1}b \to \varepsilon\}\end{aligned} \tag{2}$$

is terminating and for any $w = w(a, b)$,

$$w =_G 1 \quad \Leftrightarrow \quad w \to^*_{\mathcal{R}} \varepsilon.$$

In particular, $\mathcal{R}$ gives a decision algorithm for the Word Problem in $G_{(1,2)}$.

Proof. Notice that

$$\begin{aligned}G_{(1,2)} &= \langle a, b \mid b^{-1}a^{-1}bab^{-1}ab = a^2\rangle = \langle a, t, b \mid t^{-1}at = a^2,\ b^{-1}ab = t\rangle\\ &= \langle B_{(1,2)}, b \mid b^{-1}ab = t\rangle\end{aligned}$$

which proves 1).

To prove 2) observe first that the groups $B_{(1,2)}$ and $G_{(1,2)}$ are HNN extension, which satisfy the properties (G1), (G2), and (G3) from Corollary 2.1. Hence WP in both groups can be solved by the corresponding rewriting systems of the type $\mathcal{R}_{HNN}$ from the proof of Corollary 2.1. Combining these rewriting systems into one we obtain the system $\mathcal{R}$. It follows that for any $w = w(a, b)$,

$$w =_G 1 \quad \Leftrightarrow \quad w \to^*_{\mathcal{R}} \varepsilon.$$

It remains to be seen that $\mathcal{R}$ is terminating. To see this associate with each word $w = w(a, b, t)$ a triple (α, β, γ) where α is a total number of b symbols in w, β is the total number of t symbols in w, and $\gamma = |w|$. It is easy to see that any rewrite from $\mathcal{R}$ strictly decreases (α, β, γ) as an element of $\mathbb{N}^3$ in the (left) lexicographical order. Since the lexicographical order is a well-ordering it follows that the rewriting system $\mathcal{R}$ is terminating. □

Notice, that the system $\mathcal{R}$ is not confluent in general.

Proposition 2.2 states that $\mathcal{R}$ solves the Word Problem for $G_{(1,2)}$, but it does not give any estimate on the time-complexity of the rewriting procedure. To estimate the complexity of rewriting with $\mathcal{R}$ consider a sequence of words over the alphabet of $G_{(1,2)}$ defined as follows

$$\begin{aligned} w_0 &= a, \\ w_1 &= \left(b^{-1}w_0b\right)^{-1}a\left(b^{-1}w_0b\right), \\ &\vdots \\ w_{i+1} &= \left(b^{-1}w_ib\right)^{-1}a\left(b^{-1}w_ib\right). \end{aligned} \tag{3}$$

Lemma 2.3. *Let $G = G_{(1,2)} = \langle a, b \mid b^{-1}a^{-1}bab^{-1}ab = a^2\rangle$. Then (in the notation above) the following hold:*

1) *for any $i \in \mathbb{N}$, $w_i =_G a^{\left.2^{2^{\cdots^{2}}}\right\}i \text{ times}} = a^{tower_2(i)}$;*
2) *$a^{tower_2(i)}$ is the only $\mathcal{R}$-reduced form of w_i;*
3) *it takes at least $tower_2(i-1)$ elementary rewrites for $\mathcal{R}$ to rewrite w_i into $a^{tower_2(i)}$.*

Proof. By induction on k

$$w_{k+1} = \left(b^{-1}w_kb\right)^{-1}a\left(b^{-1}w_kb\right) =_G t^{-\left.2^{2^{\cdots^{2}}}\right\}k} a t^{\left.2^{2^{\cdots^{2}}}\right\}k} =_G a^{\left.2^{2^{\cdots^{2}}}\right\}k+1}$$

which proves 1). Now 2) and 3) are easy. □

Theorem 2.4. *The time function of the Magnus breakdown algorithm on $G_{(1,2)}$ is not bounded by any finite tower of exponents.*

Proof. Since the presentation $\langle a, b \mid b^{-1}a^{-1}bab^{-1}ab = a^2\rangle$ for $G_{(1,2)}$ has a unique stable letter b in its relator, the Magnus procedure represents $G_{(1,2)}$ as the HNN extension

$$G_{(1,2)} = \left\langle B_{(1,2)}, b \;\middle|\; b^{-1}ab = t\right\rangle.$$

Similarly, since the presentation $\langle a, t \mid t^{-1}at = a^2\rangle$ has a unique stable letter t in its relator, the Magnus procedure represents $B_{(1,2)}$ as the HNN extension of $\mathbb{Z} = \langle a\rangle$

$$B_{(1,2)} = \left\langle \langle a\rangle, t \;\middle|\; t^{-1}at = a^2\right\rangle.$$

Now, to determine if a given word $w = w(a, b)$ represents the identity of $G_{(1,2)}$ the Magnus process applies Britton's lemma to the constructed HNN extensions. The rewriting system $\mathcal{R}$ describes precisely the applications of Britton's lemma to the word w, when one first eliminates all the pinches related to b and then all the pinches related to t. Independently of how one realizes the rewriting in Magnus breakdown (rewriting with $\mathcal{R}$) in a deterministic fashion the rewriting of the words w_i of (3) is essentially unique and takes at least $tower_2(i-1)$ elementary rewrites to finish. Notice that for every $i \in \mathbb{N}$,

$$|w_i| = 2|w_{i-1}| + 5 \quad \text{and} \quad |w_0| = 1.$$

Hence, $|w_i| = 6 \cdot 2^n - 1$ and, as Lemma 2.3 shows, reducing the word w_i produces the word of length $tower_2(i)$. Hence the result. □

2.3. Large scale rewriting in $G_{(1,2)}$

To make the rewriting by $\mathcal{R}$ efficient one must be able to:

- work with huge numbers that appear as exponents in the rewriting process;
- perform rewrites at bulk, i.e., perform many similar rewrites at once.

In Section 5 we will use the rewriting system

$$\begin{aligned}\mathcal{R}' = {} & \{b^{-1}a^m b \to t^m,\ bt^m b^{-1} \to a^m \mid m \in \mathbb{Z}\} \\ & \cup \{t^k a^m \to a^{m2^{-k}} t^k \mid m, k \in \mathbb{Z},\ m2^{-k} \in \mathbb{Z}\} \\ & \cup \{x^k x^m \to x^{k+m} \mid k, m \in \mathbb{Z},\ x \in \{a, b, t\}\} \end{aligned} \tag{4}$$

instead of the system (2). To perform such rewrites efficiently it suffices to perform efficiently the following arithmetic operations on exponents that occur in rewriting:

(O1) addition and subtraction;
(O2) multiplication and division by a power of 2.

In the next section we introduce a representation of integer numbers over which the sequences of operations (O1) and (O2) can be performed efficiently.

3. Power circuits

In this section we define a presentation of integers which we refer to as *power circuit presentation* and show how one can perform some arithmetic operations over power circuits. See [23] for more details on circuits.

A power circuit is a quadruple $(\mathcal{P}, \mu, M, \nu)$ satisfying the conditions below:

- $\mathcal{P} = (V(\mathcal{P}), E(\mathcal{P}))$ a directed graph with no multiple edges and no directed cycles;
- $\mu : E(\mathcal{P}) \to \{1, -1\}$ a function called *the edge labelling function*;
- $M \subseteq V(\mathcal{P})$ a set of vertices called *the set of marked vertices*;
- and $\nu : M \to \{-1, 1\}$ a function called *the sign function*.

For an edge $e = v_1 \to v_2$ in $\mathcal{P}$ denote its origin v_1 by $\alpha(e)$ and its terminus v_2 by $\beta(e)$. For a vertex v in $\mathcal{P}$ define sets

$$In_v = \{e \in E(\mathcal{P}) \mid \beta(e) = v\} \quad \text{and} \quad Out_v = \{e \in E(\mathcal{P}) \mid \alpha(e) = v\}.$$

A vertex v in $\mathcal{P}$ is called a *source* if $In_v = \emptyset$. Inductively define a function $\mathcal{E} : V(\mathcal{P}) \to \mathbb{R}$ ($\mathcal{E}$ stands for evaluation) as follows: for $v \in V(\mathcal{P})$ define

$$\mathcal{E}(v) = \begin{cases} 0 & \text{if } Out_v = \emptyset; \\ 2^{\sum_{e \in Out_v} \mu(e)\mathcal{E}(\beta(e))} & \text{otherwise.} \end{cases}$$

We are interested in presentations of integer numbers only and hence we assume that $\mathcal{E}(v) \in \mathbb{Z}$ for each $v \in \mathcal{P}$. Such circuits are called *proper*. Since $\mathcal{P}$ contains no cycles the function $\mathcal{E}$ is well defined. Finally, assign a number $\mathcal{N}$ to the quadruple $(\mathcal{P}, \mu, M, \nu)$ as follows

$$\mathcal{N} = \mathcal{N}(\mathcal{P}, \mu, M, \nu) = \sum_{v \in M} \nu(v)\mathcal{E}(v).$$

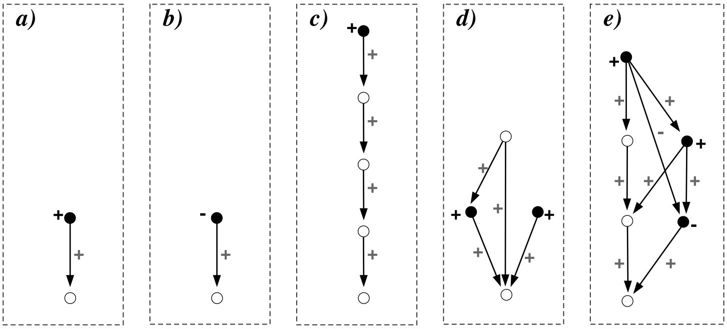

Fig. 1. Power circuits representing integers 1, −1, 16, 2 and 35. Black vertices denote the marked vertices. An edge e is labeled with $+$ if $\mu(e)=1$, a marked vertex v is labeled with $+$ if $\nu(v)=1$.

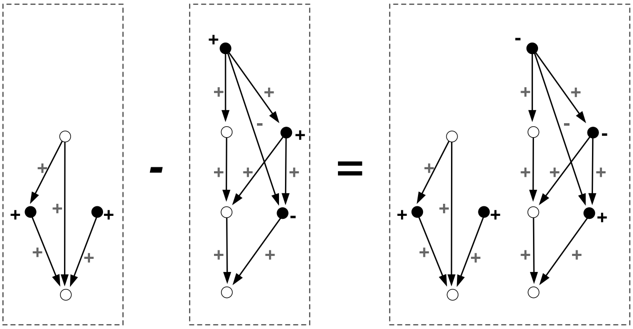

Fig. 2. Taking difference of circuits d) and e) from Fig. 1.

If $\mathcal{N}=\mathcal{N}(\mathcal{P},\mu,M,\nu)$ then we say that $(\mathcal{P},\mu,M,\nu)$ is a *power circuit presentation* of the number $\mathcal{N}\in\mathbb{R}$, or that $\mathcal{N}$ is represented by $(\mathcal{P},\mu,M,\nu)$. Throughout the paper we denote the quadruple $(\mathcal{P},\mu,M,\nu)$ simply by $\mathcal{P}$.

For a circuit $\mathcal{P}$ denote by $|\mathcal{P}|$ the number $|V(\mathcal{P})|+|E(\mathcal{P})|$ called the *size* of the circuit and by $\mathcal{N}(\mathcal{P})$ the integer represented by $\mathcal{P}$.

3.1. Zero vertices in power circuits

A vertex z in $\mathcal{P}$ is called *zero* if $Out_z=\emptyset$. It follows from the definition of the function $\mathcal{E}$ that z is a zero vertex in $\mathcal{P}$ if and only if $\mathcal{E}(z)=0$. Clearly, each non-trivial circuit has at least one zero vertex. If $\mathcal{P}$ has more than one zero vertex then its size can be reduced. The next lemma is obvious.

Lemma 3.1. *Let z_1 and z_2 be distinct zero vertices of a circuit $\mathcal{P}$ and $\mathcal{P}'$ a circuit obtained from $\mathcal{P}$ by gluing z_1 and z_2 together. Then $|V(\mathcal{P})|=|V(\mathcal{P}')|+1$ and $\mathcal{N}(\mathcal{P})=\mathcal{N}(\mathcal{P}')$.*

3.2. Addition and subtraction

Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be two circuits. To compute a circuit $\mathcal{P}_+$ such that $\mathcal{N}(\mathcal{P}_+)=\mathcal{N}(\mathcal{P}_1)+\mathcal{N}(\mathcal{P}_2)$ one can take a union of $\mathcal{P}_1$ and $\mathcal{P}_2$ leaving the labeling functions the same. Clearly the obtained result satisfies the equality $\mathcal{N}(\mathcal{P}_+)=\mathcal{N}(\mathcal{P}_1)+\mathcal{N}(\mathcal{P}_2)$. Similarly, to compute a circuit $\mathcal{P}_-$ such that $\mathcal{N}(\mathcal{P}_-)=\mathcal{N}(\mathcal{P}_1)-\mathcal{N}(\mathcal{P}_2)$ one can take a union of $\mathcal{P}_1$ and $\mathcal{P}_2$ leaving the labeling functions on $\mathcal{P}_1$ the same and changing the labeling function on $M(\mathcal{P}_2)$ to the opposite. Clearly the obtained result satisfies the required equality. See Fig. 2 for an example of difference of two circuits.

Proposition 3.2. *(See [23], Proposition 7.2.) Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be power circuits and $\mathcal{P}_+ = \mathcal{P}_1 + \mathcal{P}_2$. Then $\mathcal{N}(\mathcal{P}_+) = \mathcal{N}(\mathcal{P}_1) + \mathcal{N}(\mathcal{P}_2)$, $|V(\mathcal{P}_+)| = |V(\mathcal{P}_1)| + |V(\mathcal{P}_2)|$, and $|E(\mathcal{P}_+)| = |E(\mathcal{P}_1)| + |E(\mathcal{P}_2)|$. Moreover, $\mathcal{P}_+$ and $\mathcal{P}_-$ are computed in time $O(|\mathcal{P}_1| + |\mathcal{P}_2|)$.*

3.3. Comparison (circuit reduction)

We say that $\mathcal{P}$ is *reduced* if it is proper and for every $v_1, v_2 \in V(\mathcal{P})$

$$\mathcal{E}(v_1) = \mathcal{E}(v_2) \quad \Leftrightarrow \quad v_1 = v_2.$$

For more detail see [23]. Here we outline the main results about reduced circuits.

Theorem 3.3. *(See [23], Propositions 5.16 and 5.17.) There exists an algorithm which for every power circuit $\mathcal{P}$ constructs an equivalent reduced circuit $\mathcal{P}'$ such that*

$$\left|V\left(\mathcal{P}'\right)\right| \leqslant \left|V(\mathcal{P})\right| + 1 \quad \textit{and} \quad \left|M\left(\mathcal{P}'\right)\right| \leqslant \left|M(\mathcal{P})\right|,$$

and orders vertices of $\mathcal{P}'$ according to their $\mathcal{E}$ values. Moreover, the time complexity of the procedure is $O(|V(\mathcal{P})|^3)$.

Proposition 3.4. *(See [23], Proposition 7.11.) There exists a deterministic algorithm which for every power circuit $\mathcal{P}$ computes*

$$Sign(\mathcal{P}) = \begin{cases} -1, & \textit{if}\ \mathcal{N}(\mathcal{P}) < 0; \\ 0, & \textit{if}\ \mathcal{N}(\mathcal{P}) = 0; \\ 1, & \textit{if}\ \mathcal{N}(\mathcal{P}) > 0. \end{cases}$$

Moreover, the time complexity of that procedure is bounded above by $O(|V(\mathcal{P})|^3)$.

Proposition 3.5. *Let $\mathcal{P}$ be a reduced power circuit, $v \in M(\mathcal{P})$ the marked vertex satisfying $\mathcal{E}(v) = \min\{\mathcal{E}(u) \mid u \in M(\mathcal{P})\}$, and $m \in \mathbb{N}$. Then $\mathcal{N}(\mathcal{P})$ is divisible by 2^m if and only if $\mathcal{E}(v_i)$ is.*

Proof. Follows from the definition of $\mathcal{N}(\mathcal{P})$ and the definition of a reduced circuit. □

3.4. Multiplication and division by a power of two

Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be power circuits. Assume that $\mathcal{N}(\mathcal{P}_2) > 0$. In this section we outline a procedure for constructing circuits $\mathcal{P}_\bullet$ and $\mathcal{P}_\circ$ satisfying

$$\mathcal{N}(\mathcal{P}_\bullet) = \mathcal{N}(\mathcal{P}_1) \bullet \mathcal{N}(\mathcal{P}_2) := \mathcal{N}(\mathcal{P}_1) \cdot 2^{\mathcal{N}(\mathcal{P}_2)}$$

and

$$\mathcal{N}(\mathcal{P}_\circ) = \mathcal{N}(\mathcal{P}_1) \circ \mathcal{N}(\mathcal{P}_2) := \frac{\mathcal{N}(\mathcal{P}_1)}{2^{\mathcal{N}(\mathcal{P}_2)}}.$$

Recall that $\mathcal{N}(\mathcal{P}_1) = \sum_{v \in M_1} \nu(v)\mathcal{E}(v)$, where $\mathcal{E}(v) = 2^{\sum_{e \in Out_v} \mu(e)\mathcal{E}(\beta(e))}$ is a power of 2. Hence, to multiply $\mathcal{N}(\mathcal{P}_1)$ by $2^{\mathcal{N}(\mathcal{P}_2)}$ one can multiply the values of $\mathcal{E}(v)$ by $2^{\mathcal{N}(\mathcal{P}_2)}$ for each $v \in M(\mathcal{P}_1)$ which

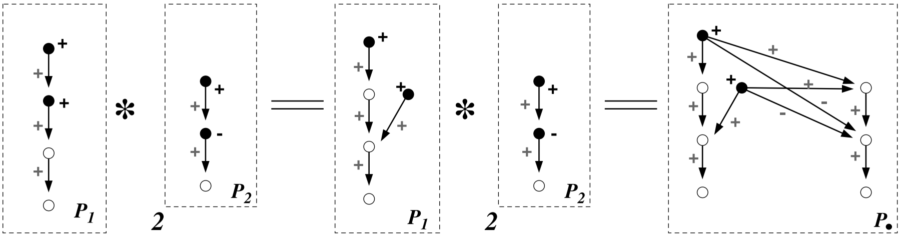

Fig. 3. Multiplication by a power of 2.

corresponds to increase of the value of the sum $\sum_{e \in Out_v} \mu(e)\mathcal{E}(\beta(e))$ by $\mathcal{N}(\mathcal{P}_2)$. Thus, to multiply $\mathcal{N}(\mathcal{P}_1)$ by $2^{\mathcal{N}(\mathcal{P}_2)}$ one can perform the following steps:

(1) make each marked vertex v in $\mathcal{P}_1$ a source;
(2) take a union of $\mathcal{P}_1$ and $\mathcal{P}_2$;
(3) for each $v_1 \in M_1$ and $v_2 \in M_2$ add an edge $e = v_1 \to v_2$ and put $\mu(e) = \nu(v_2)$;
(4) unmark all vertices of $\mathcal{P}_2$.

See Fig. 3 for an example.

In this paper we work with integer numbers only. Hence the operation $\circ$ is not always defined for all pairs $\mathcal{P}_1$, $\mathcal{P}_2$ of circuits. To check if $\mathcal{P}_1 \circ \mathcal{P}_2$ is defined one can reduce the presentation of $\mathcal{P}_1$ and check the conditions of Proposition 3.5. To actually multiply $\mathcal{P}_1$ by $2^{-\mathcal{N}(\mathcal{P}_2)}$ one needs to 1) reduce $\mathcal{P}_1$, 2) invert the value of $\mathcal{P}_2$ and, 3) apply the algorithm outlined above to compute $\mathcal{P}_\circ$.

Proposition 3.6. *Let $\mathcal{P}_1$ and $\mathcal{P}_2$ be power circuits. Assume that $\mathcal{P}_\bullet$ and $\mathcal{P}_\circ$ are obtained by the outlined above procedures. Then*:

1) $\mathcal{N}(\mathcal{P}_\bullet) = \mathcal{N}(\mathcal{P}_1)2^{\mathcal{N}(\mathcal{P}_2)}$ *and* $\mathcal{N}(\mathcal{P}_\circ) = \frac{\mathcal{N}(\mathcal{P}_1)}{2^{\mathcal{N}(\mathcal{P}_2)}}$.
2) $|V(\mathcal{P}_\bullet)|, |V(\mathcal{P}_\circ)| \leqslant |V(\mathcal{P}_1)| + |V(\mathcal{P}_2)| + |M_1|$.
3) *The time required to construct $\mathcal{P}_\bullet$ is bounded by $O(|\mathcal{P}_1| + |\mathcal{P}_2|)$.*
4) *The time required to construct $\mathcal{P}_\circ$ and to check that it properly defines an integer is bounded by $O(|V(\mathcal{P}_1)|^3 + |\mathcal{P}_2| + |M_1| \cdot |M_2|)$.*

Proof. Straightforward to check. □

3.5. *Normal forms of power circuits*

A *binary sum* is an algebraic expression

$$\varepsilon_1 2^{q_1} + \cdots + \varepsilon_k 2^{q_k}, \tag{5}$$

where $\varepsilon_i \in \{-1, 1\}$ and $q_1 > q_2 > \cdots > q_k$ are natural numbers. A binary sum (5) is called *compact* if $q_i - q_{i+1} \geqslant 2$ for every $i = 1, \ldots, k-1$.

The following lemma shows that compact binary sums give some kind of a normal form for natural numbers.

Lemma 3.7. *(See [23], Lemma 2.8.) The following hold*:

(1) *For any $n \in \mathbb{N}$ there exists a unique compact binary sum $P_n = \varepsilon_1 2^{q_1} + \cdots + \varepsilon_k 2^{q_k}$ representing n. Furthermore, $k, q_1, \ldots, q_k \leqslant \log_2 n$ and P_n can be found in linear time $O(\log_2 n)$.*
(2) *A compact binary sum representation of a given number $n \in \mathbb{N}$ involves the least possible number of terms among all other binary sums representing n.*

Let $\mathcal{P}$ be a power circuit. We say that $\mathcal{P}$ is in the *normal form* if:

(N1) $\mathcal{P}$ is proper and reduced.
(N2) For every vertex $v \in V(\mathcal{P})$ the binary sum $\sum_{e \in Out_v} \mu(e)\mathcal{E}(\beta(e))$ is compact (after proper enumeration of children of v).
(N3) The binary sum $\mathcal{E}(\mathcal{P}) = \sum_{v \in M} \nu(v)\mathcal{E}(v)$ is in the compact form.

Power circuits $\mathcal{P}_1$ and $\mathcal{P}_2$ are *isomorphic* if there exists a graph isomorphism $\varphi : \mathcal{P}_1 \to \mathcal{P}_2$ mapping $M(\mathcal{P}_1)$ bijectively onto $M(\mathcal{P}_2)$ and preserving the values of μ, ν, and γ.

Theorem 3.8 *(Normal forms). (See [23], Theorem 4.9.) Two power circuits in the normal form represent the same number if and only if they are isomorphic.*

The following theorem gives an important algorithmic result about normal power circuits that we use in the sequent.

Theorem 3.9 *(Computing normal forms). There exists an algorithm which for any $n \in \mathbb{N}$ computes the unique normal power circuit $\mathcal{P}_n$ representing n in time $O(\log_2 n \log_2 \log_2 n)$. Furthermore, $|V(\mathcal{P}_n)| \leqslant \lceil \log_2 n \rceil + 2$.*

Proof. It follows from Lemma 3.7 that the required circuit $\mathcal{P}_n$ contains at most $2 + \log_2 n$ vertices with $\mathcal{E}$-values $0, 2^0, 2^1, \ldots \leqslant n$. It is straightforward to process each vertex v with $\mathcal{E}(v) = 2^k$ and make sure that it satisfies the property (N2) in time bounded by $O(\log_2 k) \leqslant O(\log_2 \log_2 n)$. Therefore, $\mathcal{P}_n$ indeed can be constructed in time $O(\log_2 n \log_2 \log_2 n)$. □

4. Power sequences

Let X be an alphabet, $x_1, \ldots, x_n \in X^{\pm 1}$, and $\mathcal{P}_1, \ldots, \mathcal{P}_n$ power circuits. A sequence $\mathcal{S} = (x_1, \mathcal{P}_1), \ldots, (x_n, \mathcal{P}_n)$ is called a *power sequence*. We say that a power sequence $\mathcal{S}$ represents a word

$$W(\mathcal{S}) = x_1^{\mathcal{N}(\mathcal{P}_1)} \ldots x_n^{\mathcal{N}(\mathcal{P}_n)}.$$

If all the power circuits $\mathcal{P}_i$ are normal then the sequence $\mathcal{S}$ is termed *normal*.

For a power sequence $\mathcal{S}$ we define the following numerical characteristics: the total number of marked vertices in its circuits

$$M(\mathcal{S}) = \sum_{(x,\mathcal{P}) \in \mathcal{S}} \bigl|M(\mathcal{P})\bigr|;$$

and the total number of vertices in its circuits

$$V(\mathcal{S}) = \sum_{(x,\mathcal{P}) \in \mathcal{S}} \bigl|V(\mathcal{P})\bigr|.$$

If $\mathcal{S}$ represents a word $w = x_1^{p_1} \ldots x_n^{p_n}$ and $g = x_i^{p_i} \ldots x_j^{p_j}$ is a subword of w denote by $\mathcal{S}_g$ the segment of $\mathcal{S}$ corresponding to g.

A power sequence is *reduced* if it does not contain:

(R1) a pair $(x, \mathcal{P})$ where $\mathcal{N}(\mathcal{P}) = 0$,
(R2) a subsequence $(x, \mathcal{P}), (x, \mathcal{P}')$.

To reduce a power sequence $\mathcal{S}$ one can consequently replace non-reduced subsequences $(x, \mathcal{P}), (x, \mathcal{P}')$ by the corresponding pairs $(x, \mathcal{P} + \mathcal{P}')$, and remove the pairs (R1). The described process is called a *reduction* of a power sequence.

Proposition 4.1. *Let $\mathcal{S}$ be a power sequence and $\mathcal{S}'$ be obtained by reducing $\mathcal{S}$. Then $\mathcal{S}$ and $\mathcal{S}'$ represent the same element of the corresponding free group $F(A)$. Furthermore, $M(\mathcal{S}') \leqslant M(\mathcal{S})$ and $V(\mathcal{S}') \leqslant V(\mathcal{S})$. The time complexity of reduction is not greater than $O(V(\mathcal{S})^3)$.*

Proof. Follows from Proposition 3.2. □

Proposition 4.2 *(Computing normal power sequences). For a given word w in an alphabet $X^{\pm 1}$ one can compute the unique reduced normal power sequence $\mathcal{S}_n$ representing w in time $O(|w| \log |w|)$.*

5. The Word Problem in $G_{(1,2)}$

In this section we describe an efficient decision algorithm for the Word Problem in $G_{(1,2)}$. It is based on the large scale rewriting system $\mathcal{R}'$ introduced in Section 2.3, which allows one to perform many similar elementary rewrites at once. Furthermore, to avoid huge numbers as exponents in the rewriting process we keep the numbers in the compressed form, representing words by power sequences. We show in Section 5.3 that the resulting algorithm has polynomial time complexity. To some extent this is a compressed version of the classical Magnus breakdown algorithm for the Word Problem in $G_{(1,2)}$.

5.1. The compressed form of the Magnus breakdown algorithm

We mentioned in Section 2.2 that the Magnus breakdown algorithm for the Word Problem in $G_{(1,2)}$ represents the group $G_{(1,2)}$ as the HNN extension

$$G_{(1,2)} = \langle B_{(1,2)}, b \mid b^{-1}ab = t \rangle, \tag{6}$$

where

$$B_{(1,2)} = \langle a, t \mid t^{-1}at = a^2 \rangle, \tag{7}$$

and then rewrites a given word eliminating all possibles pinches according to Britton's lemma.

Eliminating several similar pinches of the type $ta^m t^{-1}$, $t^{-1}a^m t$ (that correspond to the HNN extension (7)) at once, results in the large scale rewriting system $\mathcal{R}'$ (see Section 2.3):

$$\begin{aligned} &\{b^{-1}a^m b \to t^m,\ bt^m b^{-1} \to a^m \mid m \in \mathbb{N}\} \\ &\quad \cup \{t^k a^m \to a^{m2^{-k}} t^k \mid m \in \mathbb{N},\ m2^{-k} \in \mathbb{Z}\} \\ &\quad \cup \{t^{-k} a^m \to a^{m2^k} t^{-k} \mid k \in \mathbb{N}\} \\ &\quad \cup \{x^k x^m \to x^{k+m} \mid k, m \in \mathbb{Z},\ x \in \{a, b, t\}\}. \end{aligned} \tag{8}$$

A single rewrite of the form $b^{-1}a^m b \to t^m$ or $bt^m b^{-1} \to a^m$ decreases the total power of b in a given word, so the rewriting algorithm performs at most $|w|/2$ such transformations. It remains to be seen how one can execute efficiently all other rewrites in $\mathcal{R}'$. To tame large exponents we represent words in the alphabet $\{a, b, t\}^{\pm 1}$ by the power sequences from Section 4. This is done in Section 5.2. In particular, this involves an efficient checking if a word $u(a, t)$, when given as a power sequence, is equal to a^m or t^m in $B_{(1,2)}$.

The decision algorithm for the Word Problem in $G_{(1,2)}$ denoted by $\mathcal{A}$ can be roughly described as follows.

- Given a word $w(a,b,t)$ find a normal power sequence $\mathcal{S}_w$ for w.
- Eliminate all the pinches in the sequence $\mathcal{S}_w$ that come from the HNN extension (6). If the resulting sequence contains the letter b, then return *No*.
- Otherwise, eliminates all the pinches in the sequence $\mathcal{S}_w$ that come from the HNN extension (7), using the rules from $\mathcal{R}'$. If the resulting word is empty, then return *Yes*. Otherwise, return *No*.

More precise description of the algorithm $\mathcal{A}$ is given below.

Algorithm 1 Word Problem for $G_{(1,2)}$

```
Input: A word $w = w(a,b,t)$.
Output: Yes if $w$ represents the identity in $G_{(1,2)}$, No otherwise.
Represent $w$ as a product of powers

    $w(a,b,t) = g_0(a,t)b^{\varepsilon_1}g_1(a,t)b^{\varepsilon_2}g_2(a,t)\ldots b^{\varepsilon_n}g_n(a,t)$    (9)

  where

    $g_i(a,t) = a^{m_{i,0}}t^{\delta_{i,1}}a^{m_{i,1}}t^{\delta_{i,2}}a^{m_{i,2}}\ldots t^{\delta_{i,k_i}}a^{m_{i,k_i}}$    (10)

  and $\varepsilon_i, \delta_{i,j}, m_{i,j} \in \mathbb{Z}$.
Compute a power sequence $\mathcal{S}$ representing $w$.
while $\mathcal{S}$ contains a subsequence $\mathcal{S}_{g_i}$ satisfying the following do
    if $\varepsilon_i < 0$, $\varepsilon_{i+1} > 0$, and $\mathcal{S}_{g_i} =_B a^p$ for some $p \in \mathbb{Z}$ then
        Replace $(b,\varepsilon_i)\mathcal{S}_{g_i}(b,\varepsilon_{i+1})$ in $\mathcal{S}$ with $(b,\varepsilon_i+1),(t,p),(b,\varepsilon_{i+1}-1)$.
    end if
    if $\varepsilon_i > 0$, $\varepsilon_{i+1} < 0$, and $\mathcal{S}_{g_i} =_B t^p$ for some $p \in \mathbb{Z}$ then
        Replace $(b,\varepsilon_i)\mathcal{S}_{g_i}(b,\varepsilon_{i+1})$ in $\mathcal{S}$ with $(b,\varepsilon_i-1),(a,p),(b,\varepsilon_{i+1}+1)$.
    end if
end while
if $\mathcal{S}$ involves a letter $b$ then
    return No.
end if
if $\mathcal{S}$ represents the trivial element in $B_{(1,2)}$ then
    return Yes.
else
    return No.
end if
```

In the next section we describe how one can decide if $\mathcal{S}_{g_i} =_B a^p$ or $\mathcal{S}_{g_i} =_B t^p$ and how to compute the corresponding power p.

5.2. Word processing in $B_{(1,2)}$

Let

$$(a,\mathcal{P}_{m_0}),(t,\mathcal{P}_{\delta_1}),(a,\mathcal{P}_{m_1}),(t,\mathcal{P}_{\delta_2}),(a,\mathcal{P}_{m_2})\ldots(t,\mathcal{P}_{\delta_k}),(a,\mathcal{P}_{m_k}) \tag{11}$$

be a power sequence representing a word

$$g = a^{m_0}t^{\delta_1}a^{m_1}t^{\delta_2}a^{m_2}\ldots t^{\delta_k}a^{m_k}$$

over the alphabet of $B_{(1,2)}$.

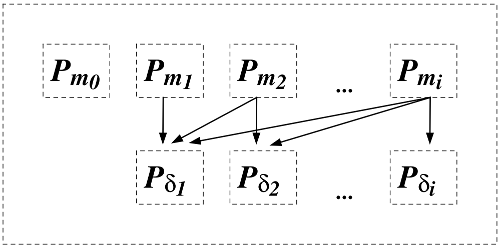

Fig. 4. A circuit representing $\sum_{i=0}^{k} \mathcal{P}_{m_i} \circ (\sum_{j=1}^{i} \mathcal{P}_{\delta_j})$.

Proposition 5.1 *(All non-positive powers). Consider a sequence* (11)*. Assume that $\delta_1 + \cdots + \delta_i \leqslant 0$ for every $i = 1, \ldots, k$. Then $g = a^M t^\sigma$ in $B_{(1,2)}$ where*

$$\sigma = \sum_{i=1}^{k} \delta_i \quad and \quad M = \sum_{i=0}^{k} m_i \cdot 2^{-\sum_{j=1}^{i} \delta_j}.$$

Furthermore, there exist power circuits $\mathcal{P}_M$ and $\mathcal{P}_\sigma$ such that $\mathcal{N}(\mathcal{P}_M) = M$ and $\mathcal{N}(\mathcal{P}_\sigma) = \sigma$ and

(a) *$|V(\mathcal{P}_\sigma)| = \sum_{j=1}^{k} |V(\mathcal{P}_{\delta_j})|$ and $|V(\mathcal{P}_M)| \leqslant \sum_{j=0}^{k}(|V(\mathcal{P}_{m_j})| + |M(\mathcal{P}_{m_j})|) + \sum_{j=1}^{k} |V(\mathcal{P}_{\delta_j})|$;*
(b) *$|M(\mathcal{P}_\sigma)| = \sum_{j=1}^{k} |M(\mathcal{P}_{\delta_j})|$ and $|M(\mathcal{P}_M)| = \sum_{j=0}^{k} |M(\mathcal{P}_{m_j})|$.*

Proof. The equality $g = a^M t^\sigma$ in $B_{(1,2)}$ is obvious. A circuit $\mathcal{P}_\sigma$ can be constructed by taking a disjoint union of the circuits $\mathcal{P}_{\delta_1}, \ldots, \mathcal{P}_{\delta_k}$. A circuit $\mathcal{P}_M$ can be obtained as follows.

- Take a disjoint union of the circuits $\mathcal{P}_{m_0}, \ldots, \mathcal{P}_{m_k}, \mathcal{P}_{\delta_1}, \ldots, \mathcal{P}_{\delta_k}$.
- Add edges between $\mathcal{P}_{m_i}$ and $\mathcal{P}_{\delta_j}$ as it is done in multiplication by a power of two (see Section 3.4).
- Unmark vertices in $\mathcal{P}_{\delta_j}$'s. See Fig. 4.

Clearly $\mathcal{P}_M$ and $\mathcal{P}_\sigma$ satisfy equalities (a) and (b). □

We call the transformation of Proposition 5.1 the (T1)-transformation. Applying (T1)-transformations to subsequences of (11) we either obtain a power sequence

$$\mathcal{S} = (a, \mathcal{P}_{M_0}), (t, \mathcal{P}_{\sigma_1}), (a, \mathcal{P}_{M_1}), \ldots, (a, \mathcal{P}_{M_{n-1}}), (t, \mathcal{P}_{\sigma_n}), (a, \mathcal{P}_{M_n}), \tag{12}$$

where $\sigma_i > 0$ for every $i = 1, \ldots, n$; or a power sequence

$$\mathcal{S} = (a, \mathcal{P}_{M_0}), (t, \mathcal{P}_{\sigma_1}), (a, \mathcal{P}_{M_1}), \ldots, (a, \mathcal{P}_{M_{n-1}}), (t, \mathcal{P}_{\sigma_n}), (a, \mathcal{P}_{M_n}), (t, \mathcal{P}_{\sigma_{n+1}}), \tag{13}$$

where $\sigma_i > 0$ for every $i = 1, \ldots, n$ and $\sigma_{n+1} < 0$. Furthermore, for every σ_i and M_i we have

$$\mathcal{P}_{\sigma_i} = \mathcal{P}_{\delta_q} + \cdots + \mathcal{P}_{\delta_r} \quad \text{and} \quad \mathcal{P}_{M_i} = \sum_{j=q-1}^{r} \mathcal{P}_{m_j} \circ \left(\sum_{m=q}^{j} \mathcal{P}_{\delta_m} \right) \tag{14}$$

for some $q < r$. The next lemma follows from Britton's lemma.

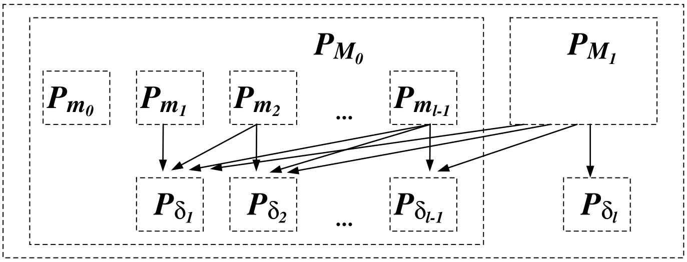

Fig. 5. A circuit representing $M_0 + 2^{-\sigma_1}M_1$.

Lemma 5.2. *If the sequence* (12) *or* (13) *represents in $B_{(1,2)}$ an element a^p or t^p for some $p \in \mathbb{Z}$ then for every $i = 1, \ldots, n$ the condition*

$$2^{-\sigma_i}\bigl(\cdots + 2^{-\sigma_{n-2}}\bigl(M_{n-2} + 2^{-\sigma_{n-1}}\bigl(M_{n-1} + 2^{-\sigma_n}M_n\bigr)\bigr)\cdots\bigr) \in \mathbb{Z}$$

is satisfied.

Proposition 5.3 *(All positive powers). Consider a sequence* (12) *or* (13) *representing an element $g \in B_{(1,2)}$. If $g = a^M t^\sigma$ in $B_{(1,2)}$, then for every $i = 1, \ldots, n$ the condition*

$$2^{-\sigma_i}\bigl(\cdots + 2^{-\sigma_{n-2}}\bigl(M_{n-2} + 2^{-\sigma_{n-1}}\bigl(M_{n-1} + 2^{-\sigma_n}M_n\bigr)\bigr)\cdots\bigr) \in \mathbb{Z} \tag{15}$$

is satisfied. Furthermore,

$$\sigma = \sigma_1 + \cdots + \sigma_n,$$
$$M = \bigl(M_0 + \cdots \bigl(M_{n-2} + (M_{n-1} + M_n \circ \sigma_n) \circ \sigma_{n-1}\bigr) \cdots \circ \sigma_1\bigr),$$

and there exist circuits $\mathcal{P}_M$ and $\mathcal{P}_\sigma$ for M and σ satisfying:

(a) $|V(\mathcal{P}_\sigma)| = \sum_{j=1}^{n} |V(\mathcal{P}_{\sigma_j})|$ *and* $|V(\mathcal{P}_M)| \leqslant \sum_{j=0}^{n}(|V(\mathcal{P}_{M_j})| + |M(\mathcal{P}_{M_j})|) + \sum_{j=1}^{n} |V(\mathcal{P}_{\sigma_j})|$;
(b) $|M(\mathcal{P}_\sigma)| = \sum_{j=1}^{n} |M(\mathcal{P}_{\sigma_j})|$ *and* $|M(\mathcal{P}_M)| = \sum_{j=0}^{k} |M(\mathcal{P}_{M_j})|$.

Proof. The equality (15) follows from Britton's lemma. To construct a circuit $\mathcal{P}_\sigma$ we take a disjoint union of $\mathcal{P}_{\sigma_1}, \ldots, \mathcal{P}_{\sigma_n}$. Clearly, $\mathcal{P}_\sigma$ satisfies (a) and (b).

We prove the existence of a required circuit $\mathcal{P}_M$ by induction on n. If $n = 0$, then $\mathcal{P}_M = \mathcal{P}_{M_0}$ and we have nothing to do. The case when $n = 1$ provides us with the induction step. In this case we need to construct a circuit representing $M_0 + 2^{-\sigma_1}M_1$. By (14) we have $\sigma_1 = \delta_1 + \cdots + \delta_l$ and $M_0 = m_0 + m_1 2^{-\delta_1} + \cdots + m_{l-1}2^{-\delta_1 \cdots -\delta_{l-1}}$. The structure of circuits $\mathcal{P}_{\sigma_1}$ and $\mathcal{P}_{M_0}$ was described in Proposition 5.1. To construct $\mathcal{P}_M$ we do the following (see Fig. 5).

- Reduce $\mathcal{P}_{M_1}$ to obtain an equivalent circuit $\mathcal{P}'_{M_1}$.
- Take a disjoint union of $\mathcal{P}_{M_0}, \mathcal{P}_{\delta_l}$ and $\mathcal{P}'_{M_1}$.
- Unmark all vertices in $\mathcal{P}_{\delta_l}$.
- By construction, $\mathcal{P}_{M_0}$ contains subgraphs corresponding to $\mathcal{P}_{\delta_0}, \ldots, \mathcal{P}_{\delta_{l-1}}$. So, we add edges from marked vertices in $\mathcal{P}_{M_1}$ to vertices in $\mathcal{P}_{\delta_0}, \ldots, \mathcal{P}_{\delta_l}$ that were marked as for operation $\circ$.
- Collapse zero-vertices in the obtained circuit.

It follows from the construction that $\mathcal{N}(\mathcal{P}_M) = M$ and that properties (a) and (b) hold for $\mathcal{P}_M$. □

Proposition 5.4 *(Complexity of word processing in $B_{(1,2)}$). It takes*

$$O\left(k\left(\sum_{j=0}^{k}\left(\left|V(\mathcal{P}_{m_j})\right|+\left|M(\mathcal{P}_{m_j})\right|\right)+\sum_{j=1}^{k}\left|V(\mathcal{P}_{\delta_j})\right|\right)^3\right)$$

operations to determine if (11) *is equivalent to a sequence* $(a,\mathcal{P})$ *or a sequence* $(t,\mathcal{P})$. *If* (11) *is equivalent to a sequence* $(a,\mathcal{P})$ *then* $\mathcal{P}$ *satisfies*:

$$\left|V(\mathcal{P})\right|\leqslant\sum_{j=0}^{k}\left(\left|V(\mathcal{P}_{m_j})\right|+\left|M(\mathcal{P}_{m_j})\right|\right)+\sum_{j=1}^{k}\left|V(\mathcal{P}_{\delta_j})\right| \quad \textit{and} \quad \left|M(\mathcal{P})\right|=\sum_{j=0}^{k}\left|M(\mathcal{P}_{m_j})\right|.$$

If (11) *is equivalent to a sequence* $(t,\mathcal{P})$ *then* $\mathcal{P}$ *satisfies*:

$$\left|V(\mathcal{P})\right|=\sum_{j=1}^{k}\left|V(\mathcal{P}_{\delta_j})\right| \quad \textit{and} \quad \left|M(\mathcal{P})\right|=\sum_{j=1}^{k}\left|M(\mathcal{P}_{\delta_j})\right|.$$

Proof. The bounds on $|V(\mathcal{P})|$ and $|M(\mathcal{P})|$ for both cases follow from Propositions 5.1 and 5.3. Furthermore, at every step in the process all power circuits in the sequence (11) have the number of vertices bounded by $\sum_{j=0}^{k}(|V(\mathcal{P}_{m_j})|+|M(\mathcal{P}_{m_j})|)+\sum_{j=1}^{k}|V(\mathcal{P}_{\delta_j})|$. Hence, it takes up to

$$O\left(\left(\sum_{j=0}^{k}\left(\left|V(\mathcal{P}_{m_j})\right|+\left|M(\mathcal{P}_{m_j})\right|\right)+\sum_{j=1}^{k}\left|V(\mathcal{P}_{\delta_j})\right|\right)^3\right)$$

operations to check if conditions of Propositions 5.1 and 5.3 hold at every step. The algorithm performs $O(k)$ transformations and hence the claimed bound on complexity. □

5.3. Complexity estimate for Algorithm 1

Finally, it remains to estimate the time complexity of Algorithm 1.

Theorem 5.5. *Algorithm* 1 *solves the Word Problem for* $G_{(1,2)}$ *in time* $O(|w|^7)$.

Proof. Let $w=w(a,b,t)$ be a reduced word over the alphabet $\{a,b,t\}$. First, Algorithm 1 constructs a power sequence $\mathcal{S}$ for w. As described in [23] it is straightforward to construct circuits for numbers $m_i,\varepsilon_i,\delta_i$. Clearly, the total number of vertices for circuits $m_i,\varepsilon_i,\delta_i$ is not greater than $2|w|$. This can be done in $O(|w|)$ steps.

In the loop 3–10 Algorithm 1 determines what subsequences $\mathcal{S}_{g_i}$ can be shortened into $(a,\mathcal{P})$ or $(t,\mathcal{P})$. By Proposition 5.4 this can be done in time

$$O\left(|g_i|\cdot\left(\left|V(\mathcal{P}_{g_i})\right|+\left|M(\mathcal{P}_{g_i})\right|\right)^3\right)$$

and the obtained circuit $\mathcal{P}$ satisfies

$$\left|V(\mathcal{P})\right|\leqslant V(\mathcal{S}_{g_i})+M(\mathcal{S}_{g_i}) \quad \text{and} \quad \left|M(\mathcal{P})\right|\leqslant M(\mathcal{S}_{g_i}).$$

Hence, a single transformation on a step 5 or 8:

- does not increase the total number of marked vertices in $\mathcal{S}$;
- can increase the total number of vertices by the number of marked vertices.

Therefore, in the worst case steps 5 and 8 are performed on a sequence S_{g_i} of size $V(S_{g_i}) = O(|w|^2)$. Algorithm 1 performs up to $|w|$ steps 5 and 8. Hence the result. □

References

[1] G. Baumslag, On generalized free products, Math. Z. 78 (1962) 423–438.

[2] G. Baumslag, A non-cyclic one-relator group all of whose finite factor groups are cyclic, J. Aust. Math. Soc. 10 (1969) 497–498.

[3] G. Baumslag, Some problems on one-relator groups, in: Proceedings of the Second International Conference on the Theory of Groups, in: Lecture Notes in Comput. Sci., vol. 372, Springer, Berlin, 1974.

[4] G. Baumslag, C. Miller, D. Troeger, Reflections on the residual finiteness of one-relator groups, Groups Geom. Dyn. 1 (2007) 209–219.

[5] G. Baumslag, A.G. Myasnikov, V. Shpilrain, Open problems in combinatorial group theory, second edition, in: Combinatorial and Geometric Group Theory, in: Contemp. Math., vol. 296, American Mathematical Society, 2002, pp. 1–38.

[6] G. Baumslag, D. Solitar, Some two-generator one-relator non-Hopfian groups, Bull. Amer. Math. Soc. 68 (1962) 199–201.

[7] M. Bestvina, M. Feighn, A combination theorem for negatively curved groups, J. Differential Geom. 35 (1992) 85–101.

[8] J.L. Britton, The word problem, Ann. of Math. 77 (1963) 16–32.

[9] D. Groves, S. Hermiller, Isoperimetric inequalities for soluble groups, Geom. Dedicata 88 (2001) 239–254.

[10] C. Drutu, M. Sapir, Non-linear residually finite groups, J. Algebra 284 (2005) 174–178.

[11] D.B.A. Epstein, J.W. Cannon, D.F. Holt, S.V.F. Levy, M.S. Paterson, W.P. Thurston, Word Processing in Groups, Jones and Bartlett Publishers, 1992.

[12] S. Gersten, The double exponential theorem for isodiametric and isoperimetric functions, Internat. J. Algebra Comput. 1 (1991) 321–327.

[13] S.M. Gersten, Dehn functions and $l1$-norms of finite presentations, in: Algorithms and Classification in Combinatorial Group Theory, Springer, Berlin, 1992, pp. 195–225.

[14] M. Greendlinger, Dehn's algorithm for the word problem, Comm. Pure Appl. Math. 13 (1960) 67–83.

[15] S. Ivanov, P. Schupp, On the hyperbolicity of small cancelation groups and one-relator groups, Trans. Amer. Math. Soc. 350 (1998) 1851–1894.

[16] O. Kharlampovich, A. Myasnikov, Hyperbolic groups and free constructions, Trans. Amer. Math. Soc. 350 (1998) 571–613.

[17] R. Lipton, Y. Zalstein, Word problems solvable in logspace, J. Assoc. Comput. Mach. 24 (1977) 522–526.

[18] R. Lyndon, P. Schupp, Combinatorial Group Theory, Classics Math., Springer, 2001.

[19] W. Magnus, Das Identitätsproblem für Gruppen mit einer definierenden Relation, Math. Ann. 106 (1932) 295–307.

[20] W. Magnus, A. Karrass, D. Solitar, Combinatorial Group Theory, Springer, 1977.

[21] J. McCool, On a question of Remeslennikov, Glasg. Math. J. 43 (2001) 123–124.

[22] S. Meskin, Nonresidually finite one-relator groups, Trans. Amer. Math. Soc. 164 (1972) 105–114.

[23] A.G. Miasnikov, A. Ushakov, Dong Wook Won, Power circuits, exponential algebra, and time complexity, preprint, available at http://arxiv.org/abs/1006.2570, 2010.

[24] B.B. Newman, Some results on one-relator groups, Bull. Amer. Math. Soc. 74 (1968) 568–571.

[25] A.N. Platonov, Isoparametric function of the Baumslag–Gersten group, Vestnik Moskov. Univ. Ser. I Mat. Mekh. (2004) 12–17 (in Russian).

[26] R. Vol'vachev, Linear representation of certain groups with one relation, Vestsi Akad. Navuk BSSR Ser. Fiz.-Mat. Navuk 124 (1985) 3–11.

[27] B.A.F. Wehrfritz, Generalized free products of linear groups, Proc. Lond. Math. Soc. 27 (1973) 402–424.

[28] D. Wise, Research announcement: The structure of groups with a quasiconvex hierarchy, Electron. Res. Announc. Math. Sci. 16 (2009) 44–55.