# Composition Problems for Braids

**Igor Potapov**

**University of Liverpool, Computer Science Department, Liverpool, UK**
potapov@liverpool.ac.uk

## Abstract

In this paper we investigate the decidability and complexity of problems related to braid composition. While all known problems for a class of braids with 3 strands, $B_3$, have polynomial time solutions we prove that a very natural question for braid composition, the membership problem, is NP-hard for braids with only 3 strands. The membership problem is decidable for $B_3$, but it becomes harder for a class of braids with more strands. In particular we show that fundamental problems about braid compositions are undecidable for braids with at least 5 strands, but decidability of these problems for $B_4$ remains open. The paper introduce a few challenging algorithmic problems about topological braids opening new connections between braid groups, combinatorics on words, complexity theory and provides solutions for some of these problems by application of several techniques from automata theory, matrix semigroups and algorithms.

**1998 ACM Subject Classification** F.2.2 Nonnumerical Algorithms and Problems, F.4.2 Grammars and Other Rewriting Systems, F.4.3 Formal Languages

**Keywords and phrases** Braid group, automata, group alphabet, combinatorics on words, matrix semigroups, NP-hardness, decidability

**Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2013.175

## 1 Introduction

In this paper we investigate the decidability and complexity for a number of problems related to braid composition. Braids are classical topological objects that attracted a lot of attention due to their connections to topological knots and links as well as their applications to polymer chemistry, molecular biology, cryptography, quantum computations and robotics [1, 11, 14].

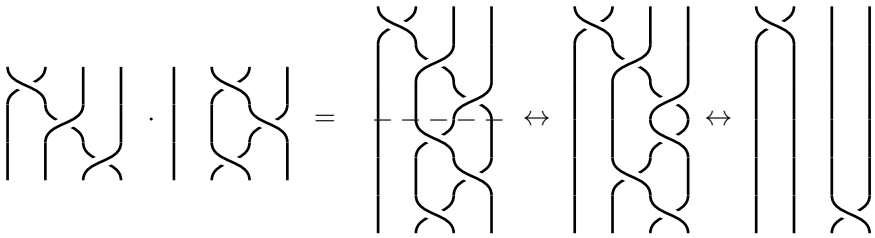
The discovery of a various cryptosystems based on the braid group inspired a new line of research about the complexity analysis of decision problems for braids, including the word problem, the generalized word problem, root extraction problem, the conjugacy problem and the conjugacy search problem. For many problems the polynomial time solutions were found, but it was surprisingly shown by M. S. Paterson and A. A. Razborov in 1991 that another closely related problem, the *non-minimal braid problem*, to be NP-complete [16]

> **Non-minimal braid problem:** Given a word $\omega$ in the generators $\sigma_1, \ldots, \sigma_{n-1}$ and their inverses, determine whether there is a shorter word $\omega'$ in the same generators which represents the same element of the $n$-strand braid group $B_n$?

The main result of this paper is to show another hard problem for braids in $B_3$, i.e. with only three strands. The problem can be naturally formulated in terms of composition (or concatenation) of braids which is one of the fundamental operations for the Braid Group.

Given two geometric braids, we can compose them, i.e. put one after the other making the endpoints of the first one coincide with the starting points of the second one. There is a neutral element for the composition: it is the trivial braid, also called identity braid, i.e. the class of the geometric braid where all the strings are straight. Two geometric braids are


isotopic if there is a continuous deformation of the ambient space that deforms one into the other, by a deformation that keeps every point in the two bordering planes fixed.



In this paper we study several computational problems related to composition of braids: Given a set of braids with $n$ strands $B = b_1, \ldots, b_k \in B_n$. Let us denote a semigroup of braids, generated by $B$ and the operation of composition, by $\langle B \rangle$.

- **Membership problem.** Check whether exist a composition of braids from a set $B$ that is isotopic to a given braid $b$. I.e. is $b$ in $\langle B \rangle$ ?
- **Identity problem.** Check whether exist a composition of braids from a set $B$ that is isotopic to a trivial braid.
- **Group problem.** Check whether for any braid $b \in B$ we can construct the inverse of $b$ by composition of braids from $B$. I.e. is a semigroup $\langle B \rangle$ a group?

| | $B_3$ | $B_4$ | $B_5$ |
|---|---|---|---|
| Membership | Decidable, NP-hard | ? | Undecidable |
| Group/Identity | Decidable | ? | Undecidable |

In contrast to many polynomial time problems we show that the Membership problem for $B_3$ is NP-hard[1] by using a combination of new and existing encoding techniques from automata theory, group theory, matrix semigroups [4, 5] and algebraic properties of braids [1]. Then we prove decidability result for the membership problem for $B_3$ which is the first non-trivial case where composition is associative, but it is non-commutative. The membership problem for braids in $B_3$ has a very close connection with other non-trivial computational problems in matrix semigroups. since the braid group $B_3$ is the universal central extension of the modular group $PSL(2, \mathbb{Z})$. The idea of decidability in $B_3$ was inspired by the work of several authors on the membership problem for $2 \times 2$ matrix semigroups [9, 13, 4, 5]. We also show that fundamental problems about the braid compositions are undecidable for braids with at least 5 strands, but decidability of these problems for $B_4$ remains open.

## 2 Preliminaries

### 2.1 Words and Automata

Given an alphabet $\Gamma = \{1, 2, \ldots, m\}$, a word $w$ is an element $w \in \Gamma^*$. We denote the concatenation of two words $u$ and $v$ by either $u \cdot v$ or $uv$ if there is no confusion. For a letter $a \in \Gamma$, we denote by $\overline{a}$ or $a^{-1}$ the inverse letter of $a$, such that $a\overline{a} = \varepsilon$ where $\varepsilon$ is the empty word. We also denote $\overline{\Gamma} = \Gamma^{-1} = \{\overline{1}, \overline{2}, \ldots, \overline{m}\}$ and for a word $w = w_1 w_2 \cdots w_n$, we denote $\overline{w} = w^{-1} = w_n^{-1} \cdots w_2^{-1} w_1^{-1}$.

The free group over a generating set $H$ is denoted by $\mathrm{FG}(H)$, i.e., the free group over two elements $a$ and $b$ is denoted as $\mathrm{FG}(\{a, b\})$. For example, the elements of $\mathrm{FG}(\{a, b\})$ are all

[1] Note that proposed NP-hardness construction is not directly applicable for Identity Problem.

the words over the alphabet $\{a, b, a^{-1}, b^{-1}\}$ that are reduced, i.e., that contain no subword of the form $x \cdot x^{-1}$ or $x^{-1} \cdot x$ (for $x \in \{a, b\}$). Note that $x \cdot x^{-1} = x^{-1} \cdot x = \varepsilon$.

Let $\Sigma = \Gamma \cup \overline{\Gamma}$. Using the notation of [2], we shall also introduce a reduction mapping which removes factors of the form $a\overline{a}$ for $a \in \Sigma$. To that end, we define the relation $\vdash \subseteq \Sigma^* \times \Sigma^*$ such that for all $w, w' \in \Sigma^*$, $w \vdash w'$ if and only if there exists $u, v \in \Sigma^*$ and $a \in \Sigma$ where $w = ua\overline{a}v$ and $w' = uv$. We may then define by $\vdash^*$ the reflexive and transitive closure of $\vdash$.

▶ **Lemma 1** ([2]). *For each $w \in \Sigma^*$ there exists exactly one word $r(w) \in \Sigma^*$ such that $w \vdash^* r(w)$ does not contain any factor of the form $a\overline{a}$, with $a \in \Sigma$.*

The word $r(w)$ is called the reduced representation of word $w \in \Sigma^*$. As an example, we see that if $w = 132\overline{2}1\overline{1}\,\overline{3}\,\overline{1} \in \Sigma^*$, then $r(w) = \varepsilon$.

Using standard notations, a deterministic finite automaton (DFA) is given by quintuple $(Q, \Sigma', \delta, q_0, F)$ where $Q$ is the set of states, $\Sigma'$ is the *input alphabet*, $\delta : Q \times \Sigma' \to Q$ is the *transition function*, $q_0 \in Q$ is the initial state and $F \subseteq Q$ is the set of final states of the automaton. We may extend $\delta$ in the usual way to have domain $Q \times \Sigma'^*$. Given a deterministic finite automaton $A$, the language recognized by $A$ is denoted by $L(A) \subseteq \Sigma'^*$, i.e. for all $w \in L(A)$, it holds that $\delta(q_0, w) \in F$.

▶ **Lemma 2.** *For any given $n \in \mathbb{Z}, n \geq 3$ there is a DFA $P_n$ over a group alphabet $\Sigma$, $|\Sigma| = 2n$, with $n + 2$ states and $2n$ edges such that the only word $w \in L(P_n)$ and $r(w) = \epsilon$, has length $|w| = 2^n$.*

**Proof.** We adapt the proof of a related result over *deterministic finite automata* (DFA) recently shown in [2]. Define alphabets $\Gamma = \{1, 2, \ldots, n\}$, $\overline{\Gamma} = \{\overline{1}, \overline{2}, \ldots, \overline{n}\}$ and $\Sigma = \Gamma \cup \overline{\Gamma}$. It is shown in [2] that for any $n \geq 3$, there exists a DFA $A_n$, with $n + 1$ states over $\Sigma$, such that for any word $w \in \Sigma^*$ where $w \in L(A_n)$ and $r(w) = \varepsilon$ then $|w| \geq 2^{n-1}$. Their proof is constructive and we shall now show an adaption of it. Let $Q = \{q_0, \ldots, q_{n+2}\}$ and $q_0$ be the initial state and $\{q_{n+2}\}$ is the final state. We define the transition function $\delta : Q \times \Sigma^* \to Q$ of the DFA such that:

$$\delta(q_a, c) = \begin{cases} q_1, & \text{if } c = 1 \text{ and } a = 0; \\ q_{a+1}, & \text{if } c = \overline{a} \text{ and } 1 \leq a \leq n; \\ q_0, & \text{if } c = a \text{ and } 2 \leq a \leq n-1, \\ q_{n+2}, & \text{if c = n and } a = n+1; \end{cases}$$

All other transitions are not defined. The structure of this DFA can be seen in Figure 1. The only path leading to a state $q_n$, for any $n \geq 3$ with an empty reduced word has length

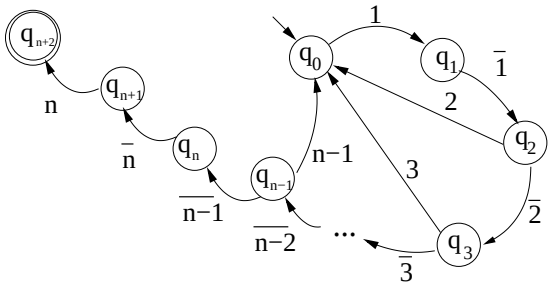



**Figure 1** A deterministic finite automaton such that the minimal non empty word $w$ such that $r(w) = \varepsilon$ and $\delta(q_0, w) \in F$ is of length $2^n$.

$2^n - 2$. The path for reaching state $q_2$ with an empty reduced word has length 2 and there are no other paths leading to $q_2$ with an empty reduced word. Let us assume that another path is leading to $q_2$ via a path where the larger index of a reachable state on this path is $j$. Then at least one symbol $j$ is not canceled in the reduced word leading to $q_2$. Consider a path from $q_i$ to $g_{i+1}$ which corresponds to reduced word $v$ then it should be of the form $v = i \cdot u \cdot \bar{i}$ where a word $u$ is an empty word and it corresponds to a path from a state $q_0$ to $q_i$ otherwise the reduced word of $v$ is not empty.

Let us assume that the path leading to a state $q_i$ with an empty reduced word, i.e $r(w) = \epsilon$ has length $2^i - 2$. Then the path for reaching state $i + 1$ with a reduced word equal to the empty word can be represented as a path $w \cdot \bar{i} \cdot ui$ where $r(u) = \epsilon$. Since $w$ is the only path to reach $q_i$ from $q_0$ then we have the required path has a form $w \cdot \bar{i} \cdot wi$ and its length is $(2^i - 2) + 1 + (2^i - 2) + 1 = 2^{i+1} - 2$. Finally we add two extra transitions to make the length of a path to be $2^n$. ◀

▶ **Lemma 3.** *For any given $s \in \mathbb{Z}$ which has a binary representation of size $m$, i.e. $m = \lceil log_2(s) \rceil$, there is a DFA $M_s$ over a group alphabet $\Sigma$, $|\Sigma| = O(m^2)$, with $O(m^2)$ states such that the only word $w \in L(M_s)$ and $r(w) = \epsilon$, has a length $|w| = s$.*

**Proof.** Let us represent $s$ as the following power series

$$\alpha_m 2^m + \alpha_{m-1} 2^{m-1} + \ldots + \alpha_2 2^1 + \alpha_1 2^0, \text{ where } \alpha_i \in \{0, 1\}.$$

For each non-zero $\alpha_i$ and $i \geq 3$ we will contract the automaton $P_i$ from Lemma 2 using unique non-intersecting alphabets for each automaton to avoid any possible cancellation of words between different parts of our final automaton. Also for non-zero $\alpha_1$, $\alpha_2$ and $\alpha_3$ we define three different automata $P_1$, $P_2$, $P_3$ having a linear structure with one $\epsilon$ transition, two consecutive $\epsilon$ transitions and four consecutive $\epsilon$ transitions, which will give us paths of length $2^0$, $2^1$ and $2^2$.

Then we will use a resulting set of automata $P_{i_1}, P_{i_2}, \ldots P_{i_l}$ to build a single automaton by merging the initial state of $P_{i_t}$ with the final state of $P_{i_{t+1}}$ for all $t = 1 \ldots l - 1$ and defining the initial state of $P_{i_1}$ as the initial state of automaton $M_s$ and the final state of $P_{i_l}$ as the final state of $M_s$. It is easy to see that following the Lemma 2 each $P_{i_t}$ will reach its own final state having an empty word iff the number of executed transition is $2^{i_t}$. So finally we build a DFA $M_s$ over a group alphabet, such that the only word $w \in L(M_s)$ and $r(w) = \epsilon$, has a length $|w| = s$.

The DFA $M_s$ over a group alphabet $\Sigma$, will have $|\Sigma| = O(m^2)$, $O(m^2)$ states and $O(m^2)$ transitions, since there are no more then $m$ parts $P_{i_1}, P_{i_2}, \ldots P_{i_l}$ and each part $P_{i_t}$ has only $i_t + 2$ states. Moreover the only word $w \in L(M_s)$ and $r(w) = \epsilon$, has a length $|w| = s$. ◀

## 2.2 Braids

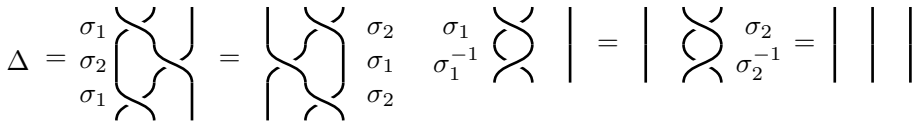
The braid groups can be defined in many ways including geometric, topological, algebraic and algebro-geometrical definitions [17]. Here we provide algebraic definition of the braid group.

▶ **Definition 4.** The $n$-strand braid group $B_n$ is the group given by the presentation with $n - 1$ generators $\sigma_1, \ldots, \sigma_{n-1}$ and the following relations $\sigma_i \sigma_j = \sigma_j \sigma_i$, for $|i - j| \geq 2$ and $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$ for $1 \leq i \leq n - 2$. These relations are called Artrin's relation.

Words in the alphabet $\{\sigma\ ,\ \sigma^{-1}\}$ will be referred to as braid words [2].

We say that a braid word $w$ is positive if no letter $\sigma_i^{-1}$ occurs in $w$. The positive braids form a semigroup denoted by $B_n^+$. There is one very important positive braid known as the fundamental $n$-braid, $\Delta_n$. The fundamental braid of the group $B_n$ (also known as Garside element) can be written with $n(n-1)/2$ Artin generators as: $\Delta_n = (\sigma_{n-1}\sigma_{n-2}\dots\sigma_1)(\sigma_{n-1}\sigma_{n-2}\dots\sigma_2)\dots\sigma_{n-1}$.

Geometrically, the fundamental braid is obtained by lifting the bottom ends of the identity braid and flipping (right side over left) while keeping the ends of the strings in a line. The inverse of the fundamental braid $\Delta_n$ is denoted by $\Delta_n^{-1}$.




Let $B_3 = \{\sigma_1, \sigma_2 | \sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2\}$ be the group with three braids. Let $\Delta$ be the Garside element: $\Delta = \sigma_1\sigma_2\sigma_1$. Let $\tau : B_3 \to B_3$ be automorphism defined by $\sigma_1 \to \sigma_2$, $\sigma_2 \to \sigma_1$. It is straightforward to check that

$$\Delta b = \tau(b)\Delta, \qquad \Delta^{-1}b = \tau(b)\Delta^{-1}, \qquad b \in B_3. \tag{1}$$

▶ **Lemma 5** ([15]). *Two positive words are equal in $B_3$ if and only if they can be obtained from each other by applying successively the relation $\sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2$ . A positive word is left or right divisible by $\Delta$ if and only if it contains the subword $\sigma_1\sigma_2\sigma_1$ or $\sigma_2\sigma_1\sigma_2$ .*

▶ **Lemma 6** ([12, 1]). *Garside normal form – Every braid word $w \in B_n$ can be written uniquely as $\Delta^k b$, where $k$ is an integer and $b$ is a positive braid of which $\Delta$ is not a left divisor.*

▶ **Definition 7.** Two braids are isotopic if their braid words can be translated one into each other via the relations from the Defintion 4 plus the relations $\sigma_i\sigma_i^{-1} = \sigma_i^{-1}\sigma_i = 1$, where 1 is the identity (trivial braid).

Let us define a set of natural problems for semigroups and groups in the context of braid composition. Given a finite set of braids $B$. A multiplicative semigroup $\langle B\rangle$ is a set of braids that can be generated by any finite composition of braids from $B$.

Membership Problem: Given a braid $b \in B_n$ and a finite set of braids $B \subseteq B_n$, does there exist a composition $Y_1Y_2\cdots Y_r$, with each $Y_i \in B$ such that $Y_1Y_2\cdots Y_r = b$? In other words, is $b \in \langle M\rangle$? In the Membership Problem, when braid $b$ is the trivial braid, we call this problem the Identity Problem.

The Identity Problem for semigroups is a well-known challenging problem which is also computationally equivalent to another fundamental problem in Group Theory: given a finitely generated semigroup $S$, decide whether a subset of the generator of S generates a nontrivial group (Group Problem) [9].

[2] Whenever a crossing of strands $i$ and $i+1$ is encountered, $\sigma_i$ or ${\sigma_i}^{-1}$ is written down, depending on whether strand $i$ moves under or over strand $i+1$.

## 3 NP-hardness of the Membership Problem in $B_3$

In this section we show that the Membership is NP-hard for braids in $B_3$. Our reduction will use the following well-known NP-complete problem. Subset Sum Problem: Given a positive integer $x$ and a finite set of positive integer values $S = \{s_1, s_2, \ldots, s_k\}$, does there exist a nonempty subset of $S$ which sums to $x$?

We will require the following encoding between words over an arbitrary group alphabet and a binary group alphabet, which is well known from the literature.

▶ **Lemma 8.** *Let $\Sigma' = \{z_1, z_2, \ldots, z_l\}$ be a group alphabet and $\Sigma_2 = \{c, d, \overline{c}, \overline{d}\}$ be a binary group alphabet. Define the mapping $\alpha : \Sigma' \to \Sigma_2^*$ by:*

$$\alpha(z_i) = c^i d \overline{c}^i, \alpha(\overline{z_i}) = c^i \overline{d} \overline{c}^i,$$

*where $1 \leq i \leq l$. Then $\alpha$ is a monomorphism [3] (see [8] for more details). Note that $\alpha$ can be extended to domain $\Sigma'^*$ in the usual way.*

▶ **Lemma 9** ([7])**.** *Let $\Sigma_2 = \{c, d, \overline{c}, \overline{d}\}$ be a binary group alphabet and define $f : \Sigma_2^* \to B_3$ by: $f(c) = {\sigma_1}^4, f(\overline{c}) = {\sigma_1}^{-4}, f(d) = {\sigma_2}^4, f(\overline{d}) = {\sigma_2}^{-4}$. Then mapping $f$ is a monomorphism.*

The above two morphisms give a way to map words from an arbitrary sized alphabet into the set braid words in $B_3$. We will later require the following corollary concerning mappings $f$ and $\alpha$ to allow us to argue about the size of braid words constructed by $f \circ \alpha$.

▶ **Corollary 10.** *Let $\alpha$ and $f$ be mappings as defined in Lemma 8 and Lemma 9, then:*

$$f(\alpha(z_j)) = f(c^j d \overline{c}^j) = {\sigma_1}^{4j} {\sigma_2}^4 {\sigma_1}^{-4j}$$

*and the length of a braid word from $B_3$ corresponding a symbol $z_j \in \Sigma'$ is $8j + 4$.*

▶ **Theorem 11.** *The* Membership Problem *is NP-hard for braids from $B_3$*

**Proof.** We shall use an encoding of the Subset Sum Problem into a set of braids from $B_3$. Define an alphabet $\Sigma = \Sigma' \cup \{\Delta, \overline{\Delta}\}, \Sigma' = \{1, 2, \ldots, k+2, \overline{1}, \overline{2}, \ldots, \overline{k+2}\}$ that will be extended during the construction.

We now define a set of words $W$ which will encode the Subset Sum Problem (SSP) instance. Note that the length of words in the following set is not bounded by a polynomial of the size of the SSP instance, however this is only a transit step and will not cause a problem in the final encoding. In particular the unary representation of a number $s$ by a word $\Delta^{2s}$ will be substituted by a set of words of a polynomial size of $i, j$ and $s$ that will generate a unique word $i \cdot \Delta^{2s} j$.

$$W = \begin{array}{ll} \{1 \cdot \Delta^{2s_1} \cdot \overline{2}, & 1 \cdot \varepsilon \cdot \overline{2}, \\ 2 \cdot \Delta^{2s_2} \cdot \overline{3}, & 2 \cdot \varepsilon \cdot \overline{3}, \\ \vdots & \vdots \\ k \cdot \Delta^{2s_k} \cdot \overline{(k+1)}, & k \cdot \varepsilon \cdot \overline{(k+1)}, \\ (k+1) \cdot \overline{\Delta}^{2x} \cdot \overline{(k+2)}\} \subseteq \Sigma^* & \end{array}$$

Figure 2 shows the way in which the words of $W$ can be combined to give the identity for the reduced word on labels in the graph structure. The above assumption will mean that we

[3] A monomorphism is an injective homomorphism.

start from node 1 of the graph and choose either $a^{s_1}$ or $\varepsilon$ to move to node 2. This corresponds to $w_1$ being equal to either $1 \cdot \Delta^{2s_1} \cdot \overline{2}$ or $1 \cdot \varepsilon \cdot \overline{2}$. We follow such non-deterministic choices from node 1 until we reach a node $s_{k+2}$. At this point, if we chose $s_{i_1}, s_{i_2}, \ldots, s_{i_l}$, such that they sum to $x$, then the reduced representation of $w$ will equal $1 \cdot \overline{k+2}$. If there does not exist a solution to the subset sum problem, then it will not be possible to reach the empty word concatenating the labels on a graph structure so it would be possible to get a word $1 \cdot \overline{k+2}$, since it will be only $1 \cdot w' \cdot \overline{k+2}$, where $w' \neq \epsilon$.

Using the encoding idea from Lemma 3 we replace each transition from state $i$ to state $j$ labelled with $\Delta^{2s} j$ by the automaton $M_{2s}$ and then will encode each transition form $M_{2s}$ from a state $x$ to state $y$ with the label $z \in \Sigma$ by the braid word $f(\alpha(x)) \cdot (\sigma_1\sigma_2\sigma_3)^2 \cdot f(\alpha(z)) \cdot f(\alpha(\overline{y}))$ following Corollary 10. We use $\Delta^2 = (\sigma_1\sigma_2\sigma_3)^2$ rather then $\Delta$ to have unchanged structure of words since $\Delta^2$ is commutative with any word in $B_3$. Also each word of the following type $i \cdot \varepsilon \cdot \overline{j}$, where $i, j \in \Sigma'$ can be directly encoded by a braid $f(\alpha(i)) \cdot f(\alpha(\overline{i}))$ .

The number of states, the alphabet size and the number of edges for each $M_{2s_i}$ automaton are of the order $O(m^2)$, where $m$ is $log_2 s_i$. Thus we have that the whole automaton after replacing all $\Delta^{2s_i}$ transitions by $M_{2s_i}$ will be encoded with the finite number of words of the order $O(k \cdot log_2^2 2s)$, where $s$ is the maximal element of $\{s_1, s_2, \ldots, s_k\}$ and the length of each braid word is of the order $O(k \cdot log_2^2 2s)$. In addition to that we add $k$ words representing $\epsilon$ transitions.

Using Lemma 8, we encode the set of words $W$ into a set of braids words over the alphabet $\{\sigma_1, \sigma_1^{-1}, \sigma_2, \sigma_2^{-1}\}$, where the total number of letters will be only polynomially increased. So finally the SSP has a solution if and only iff the braid $f(\alpha(1)) \cdot f(\alpha(\overline{k+2}))$ belongs to the defined semigroup of braid words . ◀

## 4 Decidability of the Membership problem in $B_3$

▶ **Theorem 12.** *The membership problem is decidable for braids from $B_3$.*

**Proof.** Let us given a set of braid words $\{b_1, b_2, \ldots b_n\}$ from $B_3$. First let us convert them into Garside normal form $\Delta^k b$ where $k$ is an integer and $b$ is a positive braid of which $\Delta$ is not a left divisor.

In order to find the unique Garside decomposition we need to replace each occurrence of $\sigma_1^{-1}$ with $\sigma_2\sigma_1\Delta^{-1}$, and $\sigma_2^{-1}$ with $\sigma_1\sigma_2\Delta^{-1}$, and then push all $\Delta^{-1}$ to the right using (1). After that iteratively one should successively replace all subwords $\sigma_1\sigma_2\sigma_1$ and $\sigma_2\sigma_1\sigma_2$ with $\Delta$ and push them to the right using (1).

Then we construct a finite state automaton $A$ with $n$ multi-states loops representing $n$ braid words in the Garside normal form. For each braid word $b_i$ in the Garside form $\Delta^k b$, where $b = \sigma_{j_1}\sigma_{j_2}\ldots\sigma_{j_{|b_i|}}$ we define a sequence of $|b_i|$ transitions

$$s_{1,i} \xrightarrow{\Delta^k} s_{2,i} \xrightarrow{\sigma_{j_1}} s_{3,i} \xrightarrow{\sigma_{j_2}} s_{4,i} \to \ldots \to s_{|b_i|-1,i} \xrightarrow{\sigma_{j_{|b_i|}}} s_{|b_i|,i}$$

After that we merge all states $s_{1,i}$ and $s_{|b_i|,i}$ for all $i's$ into a single state $s_0$, which will be

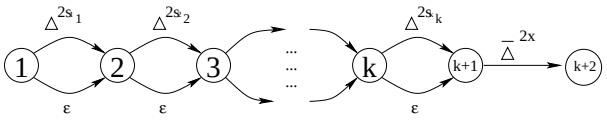



**Figure 2** The initial structure of a product which forms the identity on labels.

the initial and the final state of the automaton $A$. Thus $A$ has $n$ multi-states loops from the initial/final state $s_0$ representing $n$ braid words.

If the automaton $A$ has $q$ states we will first show that any path from a state $s$ to $t$ of the length greater than $3 \cdot 2^{(q^2-3q)}$ and equal to $\Delta^k$, for some $k \in \mathbb{Z}$ should contain a path of shorter length from $s$ to $t$ which is equal to $\Delta^{k'}$. Suppose A has a path $u$ from state $s$ to $t$ which is equal to $\Delta^k$. Then $u$ can be decomposed in at least one of two ways. Either in *Case 1* there exist two words $v_1 = \Delta^{k_1}$ and $v_2 = \Delta^{k_2}$ , $k_1, k_2 \in \mathbb{Z}$ such that $u = v_1 \cdot v_2$ or in *Case 2* there exist two words $v_1 = \Delta^{\mu_1}$ and $v_2 = \Delta^{\mu_2}$ such that

- if $\mu_1$, $\mu_2$ are even numbers then $u = \sigma_1 v_1 \sigma_2 v_2 \sigma_1$ or $u = \sigma_2 v_1 \sigma_1 v_2 \sigma_2$ or $u = \sigma_1^{-1} v_1 \sigma_2^{-1} v_2 \sigma_1^{-1}$ or $u = \sigma_2^{-1} v_1 \sigma_1^{-1} v_2 \sigma_2^{-1}$;
- if $\mu_1$, $\mu_2$ are odd numbers then $u = \sigma_1 v_1 \sigma_1 v_2 \sigma_1$ or $u = \sigma_2 v_1 \sigma_2 v_2 \sigma_2$ or $u = \sigma_1^{-1} v_1 \sigma_1^{-1} v_2 \sigma_1^{-1}$ or $u = \sigma_2^{-1} v_1 \sigma_2^{-1} v_2 \sigma_2^{-1}$;
- if $\mu_1$ is even and $\mu_2$ is odd then $u = \sigma_1 v_1 \sigma_2 v_2 \sigma_2$ or $u = \sigma_2 v_1 \sigma_1 v_2 \sigma_1$ or $u = \sigma_1^{-1} v_1 \sigma_2^{-1} v_2 \sigma_2^{-1}$ or $u = \sigma_2^{-1} v_1 \sigma_1^{-1} v_2 \sigma_1^{-1}$,
- if $\mu_1$ is odd and $\mu_2$ is even then $u = \sigma_2 v_1 \sigma_2 v_2 \sigma_1$ or $u = \sigma_1 v_1 \sigma_1 v_2 \sigma_2$ or $u = \sigma_2^{-1} v_1 \sigma_2^{-1} v_2 \sigma_1^{-1}$ or $u = \sigma_1^{-1} v_1 \sigma_1^{-1} v_2 \sigma_2^{-1}$;

Any subword $u'$ of $u$ such that $u' = \Delta^{k'}$ can also be decomposed in at least one of these two ways, so we can recursively decompose $u$ and the resulting subwords until we have decomposed $u$ into single symbols. So, we can specify a certain type of parse tree such that the automaton $A$ has a path $u$ from state $s$ to $t$ which is equal to $\Delta^k$ if and only if we can build this type of parse tree for $u$.

Let us define a parse tree for a given word $u$ equal to $\Delta^k$ from a state $s$ to $t$ as follows. Every internal node corresponds to a subword $u'$ of $u$, such that $u'$ is a power of the fundamental braid $\Delta$ and the root of the whole tree corresponds to $u$. The leaves store individual symbols $\sigma_1, \sigma_1^{-1}, \sigma_2, \sigma_2^{-1}$. When read from left to right, the symbols in the leaves of any subtree form the word that corresponds to the root of the subtree. Following Lemma 5 each internal node is

1. either the node that has two children, both of which are internal nodes that serve as roots of subtrees (corresponds to Case 1).
2. or the node that has five children, where the first, third and fifth (from the left) children are single symbols and the second and fourth children in the middle can be either empty or an internal node that is the root of another subtree which is equal to $\Delta^r$, $r \in \mathbb{Z}$ (corresponds to Case 2).

We label each internal node $\gamma$ with a pair of states $(s_{j_1}, s_{j_2})$ such that if $u'$ is the subword of $u$ that corresponds to the subtree rooted at $\gamma$, and $u = v_1 \cdot u' \cdot v_2$ then $s_{j_1} \in \delta(s, v_1)$, $s_{j_2} \in \delta(s_{j_1}, u')$. are the states reached after reading the input prefixes $v_1$ and $v_1 u'$, respectively, during the accepting computation under consideration. This implies that $s \in \delta(s_{j_2}, v_2)$ and $(s,t)$ is the label associated with the root of the tree.

If the parse tree of $u$ has two nodes $\xi_1$ and $\xi_2$ with the same state-pair label such that $\xi_2$ is a descendant of $\xi_1$, then there exists a word shorter than $u$ which is $\Delta^{k'}$. This is because we can replace the subtree rooted at $\xi_1$ with the subtree rooted at $\xi_2$. Furthermore, if an internal node $\xi_1$ is labeled with a pair $(p,q)$, for some $p, q \in Q$ then the subword $u'$ corresponding to the subtree rooted at $\xi_1$ can be removed from $u$, obtaining a shorter path. Therefore the height of the subtree corresponding to the shortest subword equal to a power of $\Delta$ is at most $q^2 - q$ and the number of leaves of a parse tree of height $h$ is at most $3(2^{q^2-q} - 2)$. In the maximal case we have the complete binary tree of depth $q - 1$ with three extra leaves in each internal node and on the last level every node has three leaves. Assume that we have a path

with some larger length from a state $s$ to $t$ with a braid word $w$ equal to $\Delta^k$ then according to above proof it should be two states $p$ and $q$ which will appear twice in the path with the following order $s \xrightarrow{A} p \xrightarrow{B} p \xrightarrow{C} q \xrightarrow{D} q \xrightarrow{E} t$ and decomposing it into five parts $A, B, C, D, E$, where $w = A \cdot B \cdot C \cdot D \cdot E = \Delta^k$, $C = \Delta^{k_1}$, $B \cdot C \cdot D = \Delta^{k_2}$. From this follows that any path from a state $s$ to $t$ in $A$, which is equal to $\Delta^k$, can be represented by a linear combination of shorter $\Delta$ paths each of which has length at most $3(2^{q^2-q} - 2)$. This gives us a bound on the number of values for expressing a power of $\Delta$'s during modification of automata $A$ and also will guarantee the termination of the following procedure, where we add a number of new transitions between states $s$ and $t$ to get a direct edge labelled by a power of $\Delta$:

1. For any of the following sequences, where $even_1$ and $even_2$ are any even numbers or 0, which means that in case of $\Delta^0$ the transition is not there:
   $s \xrightarrow{\sigma_1} \xrightarrow{\Delta^{even_1}} \xrightarrow{\sigma_2} \xrightarrow{\Delta^{even_2}} \xrightarrow{\sigma_1} t$; $\quad s \xrightarrow{\sigma_2} \xrightarrow{\Delta^{even_1}} \xrightarrow{\sigma_1} \xrightarrow{\Delta^{even_2}} \xrightarrow{\sigma_2} t$ we add
   $s \xrightarrow{\Delta^{even_1+even_2+1}} t$
2. For any of the following sequences, where $odd$ is any odd number:

   $$s \xrightarrow{\sigma_1} \xrightarrow{\Delta^{odd}} \xrightarrow{\sigma_1} \xrightarrow{\sigma_2} t; s \xrightarrow{\sigma_2} \xrightarrow{\Delta^{odd}} \xrightarrow{\sigma_2} \xrightarrow{\sigma_1} t; s \xrightarrow{\sigma_1} \xrightarrow{\sigma_2} \xrightarrow{\Delta^{odd}} \xrightarrow{\sigma_2} t; s \xrightarrow{\sigma_2} \xrightarrow{\sigma_1} \xrightarrow{\Delta^{odd}} \xrightarrow{\sigma_1} t$$

   we add $s \xrightarrow{\Delta^{odd+1}} t$
3. For any of the following sequences, where $odd_1$ and $odd_2$ are any odd numbers:
   $s \xrightarrow{\sigma_1} \xrightarrow{\Delta^{odd_1}} \xrightarrow{\sigma_1} \xrightarrow{\Delta^{odd_2}} \xrightarrow{\sigma_1} t$; $\quad s \xrightarrow{\sigma_2} \xrightarrow{\Delta^{odd_1}} \xrightarrow{\sigma_2} \xrightarrow{\Delta^{odd_2}} \xrightarrow{\sigma_2} t$
   we add $s \xrightarrow{\Delta^{odd_1+odd_2+1}} t$
4. For any of the following sequences, where $odd$ is any odd number and $even$ is any even number, we add $s \xrightarrow{\Delta^{odd+even}} t$:

   $$s \xrightarrow{\sigma_1} \xrightarrow{\Delta^{odd}} \xrightarrow{\sigma_1} \xrightarrow{\Delta^{even}} \xrightarrow{\sigma_2} t; s \xrightarrow{\sigma_2} \xrightarrow{\Delta^{odd}} \xrightarrow{\sigma_2} \xrightarrow{\Delta^{even}} \xrightarrow{\sigma_1} t;$$

   $$s \xrightarrow{\sigma_1} \xrightarrow{\Delta^{even}} \xrightarrow{\sigma_2} \xrightarrow{\Delta^{odd}} \xrightarrow{\sigma_2} t; s \xrightarrow{\sigma_2} \xrightarrow{\Delta^{even}} \xrightarrow{\sigma_1} \xrightarrow{\Delta^{odd}} \xrightarrow{\sigma_1} t$$

5. For any of the following sequences, where $z_1, z_2$ are integer numbers:
   $s \xrightarrow{\Delta^{z_1}} \xrightarrow{\Delta^{z_2}} t$; we add $s \xrightarrow{\Delta^{z_1+z_2}} t$
6. If $s = t$, for any of the above cases 1-5, then the new edge should make a cycle labelled by some power of $\Delta$, i.e $\Delta^a$, where $a \in \mathbb{Z}$. In this case if there are no other cyclic edges from a state $s$ then we add a cyclic edge from state with an expression $\Delta^{x_{new} \cdot a}$, where $x_{new}$ is a new symbol from some infinite alphabet $\{x_i | i \in \mathbb{N}\}$, which will be used later as unknown in a system of equations. Assume that there is a cyclic edge with $\Delta^z$ where $z$ is represented by a linear expression $Expr(x_{i_1}, x_{i_2}, ..., x_{i_j})$ then no extra cyclic edges are added, but the expression $Expr(x_{i_1}, x_{i_2}, ..., x_{i_j})$ will be replaces by $Expr(x_{i_1}, x_{i_2}, ..., x_{i_j}) + x_{new} \cdot a$.

Now we will generalize the cases 1-5 to incorporate the idea of expressions into the new set of rules in the straightforward way: for any $\Delta^x$, $x$ is an expressions $Expr()$ that can be a constant of a linear function. In case 6, if $a$ in $\Delta^a$ is already some expression of the form $Expr_1(...) + c$, where $c \in \mathbb{Z}$, we replace the expression on the original cycle by $Expr(x_{i_1}, x_{i_2}, ..., x_{i_j}) + Expr_1(...) + x_{new} \cdot c$. [4]

[4] Formally we should also multiply $Expr_1(...)$ by $x_{new}$ which can be avoided since the variables in the linear expression $Expr_1(...)$ are independent from $x_{new}$ and in this case can be simply renamed.

7. For $s \xrightarrow{\Delta^{Expr_1()}} p \xrightarrow{\Delta^{Expr_2()}} p \xrightarrow{\Delta^{Expr_3()}} t$; we add $s \xrightarrow{\Delta^{Expr_1()+Expr_2()+Expr_3()}} t$, where any of the expressions $Expr_i()$ can be a constant.
8. For any of the following sequences, where $\alpha, \beta \in \mathbb{Z}$ $s \xrightarrow{\sigma_{j_1}} p_0 \xrightarrow{\Delta^{Expr_0()}} p_0 \xrightarrow{\Delta^{\alpha}} p_1 \xrightarrow{\Delta^{Expr_1()}} p_1 \xrightarrow{\sigma_{j_2}} p_2 \xrightarrow{\Delta^{Expr_2()}} p_2 \xrightarrow{\Delta^{\beta}} p_3 \xrightarrow{\Delta^{Expr_3()}} p_4 \xrightarrow{\sigma_{j_3}} t$ we add the transition $s \xrightarrow{\Delta^{Expr_0()+Expr_1()+Expr_2()+Expr_3()+\alpha+\beta+1}} t$ if the values of the $\sigma$'s indices and even/odd constrains for $\Delta$ in between them will match with one of the cases 1-4. In order to record the information about even/odd case for the expressions we can add one of the following equations:
   - if $Expr()$ should have an even value then we add: $Expr() = 2 \cdot x_{new}$
   - if $Expr()$ should have an odd value then we add: $Expr() = 2 \cdot x_{new} + 1$

Now let us modify the automaton $A$ following rules 1-8 in the following way. We apply rules 1-5 in any order until it is possible then if no cycles are created then the process will terminate since there will be only a finite number of such sequences. In this case no other rules 1-8 are applicable. If there is at least one cycle after applying rules 1-5 we iteratively apply rules 6, 7 and 8: as many times as possible each, then the process is starting again. When no extra transition can be added according to the above rules the process will terminate.

Finally in order to check the membership for a braid $b = \Delta^k \cdot w$ we will need to find a path from the initial to the final state of $A$ that consists of a set of $|w|$ edges in the automaton $A$ connected by $\Delta$'s such that total sum for powers of $\Delta$'s will be equal to $k$ and the positive word after moving all $\Delta$'s to the left will be equal to $w$.

If the length of a braid word $b$ is equal to $l$ then in the canonical form $\Delta^k \cdot w$ that is isotopic to $b$ we have that the length of a positive word $w$ , i.e $h = |w|$, is limited by $2 \cdot l$, following the process of rewriting. Then the absolute value of the negative power of $\Delta$ is $2 \cdot l$ and the number of positive $\Delta$'s that can be derived from $w$ is bounded by $\frac{2l}{3}$. So the absolute value of $k$ is bounded by $|\frac{2l}{3} - l| = \frac{l}{3}$.

Let $w$ be a positive word $\sigma_{i_1} \cdot \sigma_{i_2} \cdot \ldots \cdot \sigma_{i_h}$. If $b$ is in the semigroup of braids generated by $b_1, b_2, \ldots, b_n$ then it should be a word in $A$ that consists of a set of $h$ edges with $\sigma$ labels which are connected by $\Delta$'s such that total sum for powers of $\Delta$'s will be equal to $k$ and the positive word after moving all $\Delta$'s to the left will be equal to $w$:

$$\Delta^{j_1}\sigma_{i_1}\Delta^{j_2}\sigma_{i_2}\ldots\Delta^{j_h}\sigma_{i_h} = \Delta^k w; i \in \{1,2\}, j \in \mathbb{Z}, \sum_{d=1}^{h} j_d = k$$

Now we nondeterministically choose a particular pattern of $\sigma$'s and even/odd values of $j's$ and checking whether the resulting positive word after moving all $\Delta$'s to the left will be equal $w$. Obviously it can be converted into deterministic algorithm since any of the non-deterministic guesses will be made from some finite sets. Then we also define a system of linear Diophantine equations that will correspond to the $\Delta$ transitions which are connecting $\sigma_{i_1}, \sigma_{i_2} \ldots \sigma_{i_h}$ transitions. Let us assume that the subset of $\sigma$'s transitions are connected via a $\Delta$ transitions in modified automaton $A$ with the following expressions $R_1, R_2, \ldots, R_{h+1}$, where each $R_l$ can be

- $\epsilon$, i.e. empty, which corresponds to direct connection of $\sigma_{l-1}$ and $\sigma_l$
- $\Delta_l^k$ for some $k_l \in \mathbb{Z}$
- $\Delta^{Expr_l(x_{z_1}, x_{i_2}, \ldots, x_{z_\mu})}$ - which is a linear expression with $\mu$ variables, $x_i \in \mathbb{N}$ .

The system of linear Diophantine equations and inequalities will consists of the equation $\sum_{d=1}^{h} R_d = k$ and several equations representing a number of constraints for even/odd properties as well as the restrictions on variables to be positive integers. Since this system is known to be decidable [3] we can decide the membership problem by checking the existence of a solution for our system of linear Diophantine equations and inequalities. ◀

The composition problems become harder with a larger number of strands.

Here we illustrate the technique to show undecidability of the fundamental problem whether a semigroup of braids is a group.

▶ **Lemma 13.** *[7] Subgroups* $\langle {\sigma_1}^4, {\sigma_2}^4\rangle$, $\langle {\sigma_4}^2, d\rangle$ *of the group* $B_5$ *are free and* $B_5$ *contains the direct product* $\langle {\sigma_1}^4, {\sigma_2}^4\rangle \times \langle {\sigma_4}^2, d\rangle$ *of two free groups of rang 2 as a subgroup, where* $d = \sigma_4\sigma_3\sigma_2\sigma_1^2\sigma_2\sigma_3\sigma_4$.

▶ **Theorem 14.** *The Identity problem and the Group Problem are undecidable for braids in* $B_5$.

**Proof.** It was recently proved in [6] the undecidability of the following Identity Correspondence Problem (ICP) which asks whether a finite set of pairs of words (over a group alphabet) can generate an identity pair by a sequence of concatenations:

Identity Correspondence Problem (ICP) - Let $\Sigma = \{a, b\}$ be a binary alphabet and $\Pi = \{(s_1, t_1), (s_2, t_2), \ldots, (s_m, t_m)\} \subseteq \mathrm{FG}(\Sigma) \times \mathrm{FG}(\Sigma)$. Is it decidable to determine if there exists a nonempty finite sequence of indices $l_1, l_2, \ldots, l_k$ where $1 \leq l_i \leq m$ such that $s_{l_1}s_{l_2}\cdots s_{l_k} = t_{l_1}t_{l_2}\cdots t_{l_k} = \varepsilon$, where $\varepsilon$ is the empty word (identity)?

We can directly use the Lemma 13 to encode Identity Correspondence Problem in terms of braid words. We shall use a straightforward encoding to embed an instance of the Identity Correspondence Problem into a set of braids. Given an instance of ICP say $W \subseteq \Sigma^* \times \Sigma^*$ where $\Sigma = \{a, b, a^{-1}, b^{-1}\}$ generates a free group. Define two morphisms $\phi$ and $\psi$, $\Sigma \to B_5$:

$$\phi(a) = {\sigma_1}^4, \phi(b) = {\sigma_2}^4, \phi(a^{-1}) = {\sigma_1}^{-4}, \phi(b^{-1}) = {\sigma_2}^{-4}.$$

$$\psi(a) = {\sigma_4}^2, \psi(b) = \sigma_4\sigma_3\sigma_2\sigma_1^2\sigma_2\sigma_3\sigma_4,$$

$$\psi(a^{-1}) = {\sigma_4}^{-2}, \psi(b^{-1}) = \sigma_4^{-1}\sigma_3^{-1}\sigma_2^{-1}\sigma_1^{-2}\sigma_2^{-1}\sigma_3^{-1}\sigma_4^{-1}.$$

Both $\phi$ and $\psi$ can be naturally extended for words $\Sigma^* \to B_5$:

$$\phi(a_{i_1} \ldots a_{i_j}) = \phi(a_{i_1}) \cdot \ldots \cdot \phi(a_{i_j}); \quad \psi(a_{i_1} \ldots a_{i_j}) = \psi(a_{i_1}) \cdot \ldots \cdot \psi(a_{i_j})$$

For each pair of words $(s, t) \in \Pi$, define the braid word $\phi(s) \cdot \psi(t)$. Let $S$ be a semigroup generated by these braid words. If there exists a solution to ICP, i.e., $(\varepsilon, \varepsilon)$, then we see that $\phi(\varepsilon) \cdot \psi(\varepsilon) = 1 \in S$ where 1 is the trivial braid. Otherwise, since $\psi$ and $\phi$ are injective homomorphisms, $1 \notin S$.

Thus we have that the problem whether a trivial braid can be expressed by any finite length composition of braids from $B_5$ is undecidable. The ICP problem is also computationally equivalent to the following Group Problem: is the semigroup generated by a finite set of pairs of words (over a group alphabet) a group. Using the same morphisms $\phi$ and $\psi$ we can encode the Group Problem for words by braids, having that the Group Problem for braids in $B_5$ is also undecidable. ◀

## 5 Conclusion

The paper introduce a few challenging algorithmic problems about topological braids opening new connections between braid groups, combinatorics on words, complexity theory and provides solutions for some of these problems by application of several techniques from automata theory, matrix semigroups and algorithms.

We show that the membership problem for $B_3$ is decidable. The complexity of the problem is at least NP-hard[5] and the basic upper bound for the time complexity of proposed construction is exponential. The question about the exact complexity of the membership problem in $B_3$ is left open and may require a further study in terms of improving the lower bound or designing a more efficient algorithm. We believe that the proposed technique for deciding the membership problem in $B_3$ can also be used to design the algorithm for the Freeness Problem: Given a set of braids with $n$ strands $B = b_1, \ldots, b_k \in B_n$. Let us denote a semigroup of braids, generated by $B$ and the operation of composition, by $\langle B \rangle$. Check whether any two different concatenations of braids from $B$ are not isotopic. I.e. is a semigroup of braids $\langle B \rangle$ free? One of the possibilities for solving above problem with our techniques is to follow ideas proposed in [10], but arranging a more sophisticated procedure of dealing with powers of $\Delta$'s. Finally in this paper we show that fundamental problems about the braid compositions are undecidable for braids with at least 5 strands, but decidability of these problems for $B_4$ remains open.

**Acknowledgements.** The author is grateful for many fruitful discussions with Sergei Chmutov and Victor Goryunov on the computational problems in tolopogy.

## References

1 P. Dehornoy, I. Dynnikov, D. Rolfsen, B. Wiest. Ordering braids. Mathematical Surveys and Monographs 148, Providence, R.I.: American Mathematical Society, ISBN 978-0-8218-4431-1, MR 2463428, (2008).

2 T. Ang, G. Pighizzini, N. Rampersad, and J. Shallit. Automata and reduced words in the free group. In *arXiv:0910.4555*, 2009.

3 Farid Ajili and Evelyne Contejean. Avoiding slack variables in the solving of linear Diophantine equations and inequations. Theoret. Comput. Sci., 173:183–208, 1997.

4 Paul C. Bell, Mika Hirvensalo, Igor Potapov. Mortality for $2 \times 2$ Matrices Is NP-Hard. MFCS 2012:148–159.

5 Paul C. Bell, Igor Potapov. On the Computational Complexity of Matrix Semigroup Problems. Fundam. Inform. 116(1–4):1–13 (2012).

6 Paul C. Bell, Igor Potapov. On the Undecidability of the Identity Correspondence Problem and its Applications for Word and Matrix Semigroups. Int. J. Found. Comput. Sci. 21(6): 963-978 (2010).

7 V. N. Bezverkhnii, I. V. Dobrynina. Undecidability of the conjugacy problem for subgroups in the colored braid group $R_5$. Math. Notes, 65:1 (1999), 13–19.

8 J.-C. Birget and S. Margolis. Two-letter group codes that preserve aperiodicity of inverse finite automata. *Semigroup Forum*, 76(1):159–168, 2008.

9 Christian Choffrut, Juhani Karhumaki. Some decision problems on integer matrices. ITA 39(1):125–131 (2005).

10 Julien Cassaigne, François Nicolas: On the decidability of semigroup freeness. RAIRO – Theor. Inf. and Applic. 46(3):355–399 (2012).

11 David B. A. Epstein, M. S. Paterson, J. W. Cannon, D. F. Holt, S. V. Levy, W. P. Thurston. Word Processing in Groups. A K Peters/CRC Press, 352p, 1992.

12 F.A. Garside. The braid group and other groups. In *Quart. J. Math. Oxford Ser. (2), 20:235–254*, 1969.

[5] The NP-hardness result is in line with the best current knowledge about similar problem in $SL(2, \mathbb{Z})$.

**13** Yuri Gurevich, Paul Schupp. Membership Problem for the Modular Group. SIAM J. Comput. 37(2): 425–459 (2007).

**14** Samuel J. Lomonaco, Louis H. Kauffman. Quantizing braids and other mathematical structures: the general quantization procedure. Proc. SPIE 8057, Quantum Information and Computation IX, 805702 (June 02, 2011); doi:10.1117/12.883681.

**15** S.Yu. Orevkov. Quasipositivity problem for 3-braids Proceedings of 10th Gokova Geometry-Topology Conference 2004, pp. 89–93. Turkish Journal of Math., 28(2004), 89–93.

**16** Mike Paterson, Alexander A. Razborov. The Set of Minimal Braids is co-NP-Complete. J. Algorithms 12(3): 393–408 (1991).

**17** Braid group, Wikipedia. `http://en.wikipedia.org/wiki/Braid_group`, July 2013.