Composition problems for braids: Membership, Identity and Freeness

Sang-Ki Ko Igor Potapov

July 27, 2017

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 three strands, B_3, have polynomial time solutions we prove that a very natural question for braid composition, the membership problem, is NP-complete for braids with only three strands. The membership problem is decidable in NP 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 five strands, but decidability of these problems for B_4 remains open. Finally we show that the freeness problem for semigroups of braids from B_3 is also decidable in NP.

The paper introduces 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.

1 Introduction

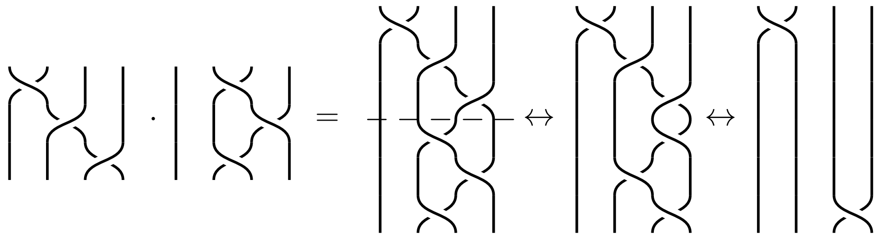
In this paper we investigate the decidability and complexity for a number of problems related to braid composition. Braids are classical topological objects that attracted a lot of attention due to their connections to topological knots and links as well as their applications to polymer chemistry, molecular biology, cryptography, quantum computations and robotics [14, 15, 23].

The discovery of 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 [17, 18, 24, 25, 26]. 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 [28].

Non-minimal Braid Problem: Given a word ω in the generators $\sigma_1, \ldots, \sigma_{n-1}$ and their inverses, determine whether there is a shorter word ω' 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 NP-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 $B = \beta_1, \dots, \beta_k \in B_n$. 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 β, i.e. is β 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 $\beta \in B$ we can construct the inverse of β by composition of braids from B, i.e. is a semigroup $\langle B \rangle$ a group?
- **Freeness Problem.** Check whether any two different concatenations of braids from B are not isotopic, i.e. is a semigroup of braids $\langle B \rangle$ free?

	B_3	B_4	B_5
Membership Problem	NP-complete	?	Undecidable
Group/Identity Problem	NP	?	Undecidable
Freeness Problem	NP	?	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 [3, 6] and algebraic properties of braids [14]. Then we prove that the membership problem for B_3 is decidable in NP, which is the first non-trivial case where composition is associative, but it is non-commutative. The main idea of the NP algorithm is to reduce the membership problem for B_3 into the emptiness problem for context-free valence grammars, which is already known to be an NP-complete problem. Note that this improves the first decidability result shown in [29]. The membership problem for braids in B_3 has a very close connection with other non-trivial computational problems in matrix semigroups. For instance, the braid group B_3 has a close relationship with the modular group $\mathrm{PSL}(2, \mathbb{Z})$ since the braid group B_3 is the universal central extension of $\mathrm{PSL}(2, \mathbb{Z})$. Recently, the problem of deciding whether a finitely generated matrix semigroup in $\mathrm{PSL}(2, \mathbb{Z})$ contains the identity matrix is proven to be NP-complete [4]. Note that the proposed NP algorithm for the membership problem for B_3 was inspired by the work of several authors on the membership problem for 2×2 matrix semigroups [3, 6, 13, 20]. We also show that fundamental problems about the braid compositions such as the identity and freeness problems are undecidable for braids with at least five strands, but decidability

[1] Note that proposed NP-hardness construction is not directly applicable for the identity problem in B_3.

of these problems for B_4 remains open. It is worth mentioning that there is no embedding from a set of pairs of words into B_4 [1]. Hence, these problems might be decidable for B_4 since our undecidability proofs for B_5 essentially rely on the embedding from a set of pairs of words into B_5.

Recently, there have been several papers on games on braids [9, 10, 21] where one player called the *attacker* tries to reach the trivial braid and the other player called the *defender* tries to keep the attacker from reaching the trivial braid based on the composition of braids from a finite set. Halava et el. [21] proved that it is undecidable to check for the existence of a winning strategy in B_3 from a given non-trivial braid and in B_5 from the trivial braid.

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 ε 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 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, Σ' 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 δ 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 Σ, $|\Sigma| = 2n$, with $n + 2$ states and $2n$ edges such that the only word $w \in L(P_n)$ and $r(w) = \varepsilon$, 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 Σ, 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 $2^n - 2$. 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 $v = i \cdot u \cdot \overline{i}$ 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.

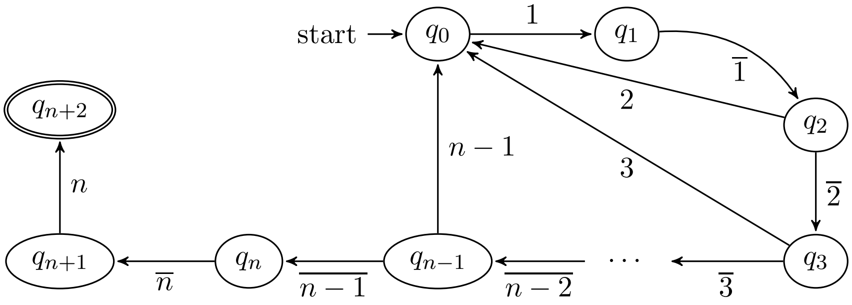


Figure 1: A deterministic finite automaton where the length of minimum non empty word w such that $r(w) = \varepsilon$ is 2^n.

Let us assume that the path leading to a state q_i with an empty reduced word, i.e $r(w) = \varepsilon$ 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 $w \cdot \overline{i} \cdot ui$ where $r(u) = \varepsilon$. Since w is the only path to reach q_i from q_0 then we have the required path has a form $w \cdot \overline{i} \cdot wi$ 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 Σ, $|\Sigma| = O(m^2)$, with $O(m^2)$ states such that the only word $w \in L(M_s)$ and $r(w) = \varepsilon$, 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 α_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 α_1, α_2 and α_3 we define three different automata P_1, P_2, P_3 having a linear structure with one ε transition, two consecutive ε transitions and four consecutive ε 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 $r(w) = \varepsilon$, has a length $|w| = s$.

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

2.2 Context-Free Valence Grammar

A *(context-free) valence grammar* over $\mathbb{Z}^k$ is a context-free grammar in which every production has an associated value from $\mathbb{Z}^k$ [16, 22]. A string in the language of the grammar can be derived in the usual way under the additional constraint that the sum of the associated values of the productions used in the derivation add up to $\mathbf{0} \in \mathbb{Z}^k$.

Formally, a valence grammar G is specified as a quadruple (N, Σ, R, S), where N is a set of *nonterminals*, Σ is a set of *terminals*, $R \subseteq N \times (N \cup T)^* \times \mathbb{Z}^k$ is a set of *productions*, and $S \in N$ is the *axiom*. For an element $(A, w, \boldsymbol{x}) \in R$, we write $A \xrightarrow{\boldsymbol{x}} w$, where $A \to w$ is the underlying production and $\boldsymbol{x} \in \mathbb{Z}^k$ is the associated value of the production.

Let $\alpha A \beta$ be a word over $N \cup \Sigma$, where $A \in N$ and $A \xrightarrow{\boldsymbol{x}} \gamma \in R$. Then, we say that A can be rewritten as γ and the corresponding derivation step is denoted $(\alpha A \beta, \boldsymbol{r}) \Rightarrow (\alpha \gamma \beta, \boldsymbol{r} + \boldsymbol{x})$. The reflexive, transitive closure of $\Rightarrow$ is denoted by $\overset{*}{\Rightarrow}$ and the (context-free) valence language generated by G is $L(G) = \{w \in \Sigma^* \mid (S, \mathbf{0}) \overset{*}{\Rightarrow} (w, \mathbf{0})\}$.

Lemma 4. *The emptiness problem for context-free valence grammars is* NP*-complete.*

Proof. It is known that the reachability problem in integer vector addition systems ($\mathbb{Z}$-VAS) is NP-complete [12]. Thus, NP-hardness follows from the fact that a valence grammar $G = (N, \Sigma, R, S)$ is a $\mathbb{Z}$-VAS if $R \subseteq N \times (N \cup \{\varepsilon\}) \times \mathbb{Z}^k$ and $N = \{S\}$.

Moreover, the NP upper bound of the emptiness problem for context-free commutative grammars with integer counters ($\mathbb{Z}$-CFCGs) [12] applies to the valence grammars since we can ignore the order of nonterminals and terminals when we consider the emptiness of the grammars. □

2.3 Braids

The braid groups can be defined in many ways including geometric, topological, algebraic and algebro-geometrical definitions [19]. Here we provide algebraic definition of the braid group.

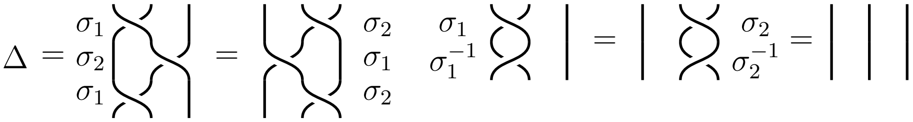
Definition 1. *The n-strand braid group B_n is the group given by the presentation with $n-1$ generators $\sigma_1, \dots, \sigma_{n-1}$ and the following relations $\sigma_i \sigma_j = \sigma_j \sigma_i$, for $|i - j| \geq 2$ and $\sigma_i \sigma_{i+1} \sigma_i = \sigma_{i+1} \sigma_i \sigma_{i+1}$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

Definition 2. *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 σ_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, Δ_n. The fundamental braid of the group B_n (also known as Garside element) can be written with $\frac{n(n-1)}{2}$ 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 Δ_n is denoted by Δ_n^{-1}.

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



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 Δ 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

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

Lemma 5 ([27])**.** *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 Δ if and only if it contains the subword $\sigma_1\sigma_2\sigma_1$ or $\sigma_2\sigma_1\sigma_2$.*

Lemma 6 (Garside normal form [19, 14])**.** *- Every braid word $w \in B_n$ can be written uniquely as $\Delta^k\beta$, where k is an integer and β is a positive braid of which Δ is not a left divisor.*

Two braids are isotopic if their braid words can be translated one into each other via the relations from the Definition 1 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.

The *membership problem* asks, given a braid $\beta \in B_n$ and a finite set of braids $B \subseteq B_n$, whether there exists a composition $Y_1Y_2\cdots Y_r$, with each $Y_i \in B$ such that $Y_1Y_2\cdots Y_r = \beta$. In other words, is $\beta \in \langle B\rangle$? In the membership problem, when braid β 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 called the *group problem.* The problem is, given a finitely generated semigroup S, to decide whether a subset of the generator of S generates a non-trivial group [13]. Finally, the *freeness problem* is to decide whether the given semigroup of braids $\langle B\rangle$ is free.

3 Membership Problem in the Braid Group B_3 is NP-complete

In this section, we show that the membership problem for braids in the braid group B_3 is NP-complete. We also prove that the freeness problem in B_3 can be decided in NP based on the NP algorithm for the membership problem.

3.1 NP-hardness of the Membership Problem in B_3

First we show that the membership problem is NP-hard for braids in B_3. Our reduction will use the the the subset sum problem which is a famous NP-complete problem. In the subset sum problem, we are given a positive integer x and a finite set of positive integer values $S = \{s_1, s_2, \ldots, s_k\}$ and asked whether there exists 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 7. *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 α is a monomorphism [3] *(see [8] for more details). Note that α can be extended to domain Σ'^* in the usual way.*

Lemma 8 ([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 α to allow us to argue about the size of braid words constructed by $f \circ \alpha$.

Corollary 1. *Let α and f be mappings as defined in Lemma 7 and Lemma 8, 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 to the symbol $z_j \in \Sigma'$ is $8j + 4$.

Now we prove that the membership problem for braid semigroups in B_3 is NP-hard.

Lemma 9. *The membership problem is* NP*-hard for braids from B_3*

Proof. We shall use an encoding of the *subset sum problem* (SSP) 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 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 Δ^{2s} will be substituted by a set of words of a polynomial size of i, j and s that will generate a unique word $i \cdot \Delta^{2s} \cdot j$.

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

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 start from node 1 of the graph and choose either a^{s_1} or ε 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 nondeterministic 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

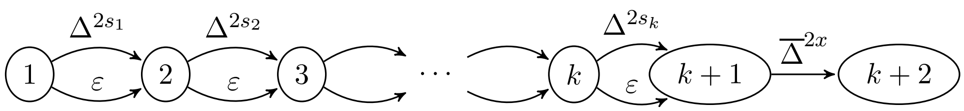


Figure 2: The initial structure of a product which forms the identity on labels.

[3] A monomorphism is an injective homomorphism.

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 $w' \neq \varepsilon$.

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 1. We use $\Delta^2 = (\sigma_1\sigma_2\sigma_3)^2$ rather then Δ to have unchanged structure of words since Δ^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 Δ^{2s_i} transitions by M_{2s_i} will be encoded with the finite number of words of the order $O(k \cdot log^2 s)$, 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 $O(k \cdot log^2 s)$. In addition to that we add k words representing ε transitions.

Using Lemma 7, we encode the set of words W into a set of braid 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 if the braid $f(\alpha(1)) \cdot f(\alpha(\overline{k+2}))$ belongs to the defined semigroup of braid words. □

3.2 NP algorithm for the Membership Problem in B_3

In this section, we show that the membership problem in B_3 is decidable in NP. Note that the decidability of the membership problem has already been solved in [29] but the time complexity of the proposed algorithm is exponential. The main idea of the algorithm proposed in [29] is as follows. Let us suppose that we are given a set $B = \{\beta_1, \beta_2, \ldots, \beta_n\}$ of braid words and a braid word β for which we need to decide whether β can be generated by the set B. We first convert the given braids β_i for $1 \leq i \leq n$ into the unique Garside normal form $\Delta^{k_i}\beta_i^+$ where $k_i \in \mathbb{N}$ is an integer and β_i^+ is a positive braid word by Lemma 6.

First, we construct an automaton which accepts a regular language $L_B = \{\Delta^{k_i}\beta_i^+ \mid 1 \leq i \leq n\}^+$ over the alphabet $\{\sigma_1, \sigma_2, \Delta, \Delta^{-1}\}$ which is the set of non-empty products of braid words from B in the Garside normal form. Then, we iteratively insert transitions labelled by any power of the fundamental braid Δ whenever we find a sequence of transitions from the automaton corresponding to Δ^x for any $x \in \mathbb{N}$. As we may have cycles in the process of inserting transitions, some transitions are labelled by $\Delta^{Expr(x_1,x_2,\ldots,x_m)}$, where $expr(x_1, x_2, \ldots, x_m)$ is a linear expression over the variables $x_1, x_2, \ldots, x_m$. Now we solve the membership problem by nondeterministically choosing a sequence of transitions and solving the system of linear Diophantine equations that consist of the equations labeling the chosen transitions from the automaton. Hence we show that the problem is decidable but in exponential time as the construction of the automaton takes exponential time.

Here we tackle the membership problem in B_3 in a slightly different way to obtain the NP upper bound. We first construct a (context-free) valence grammar generating every braid word corresponding to the input braid β and compute the intersection of the valence grammar and the regular language L_B which is the set of non-empty products of braid words from B. Now we can see that the membership problem in B_3 can be reduced to the emptiness problem for context-free valence grammars in polynomial time and it follow from Lemma 4 that the membership problem in B_3 is also in NP. In the following we first prove that there exists a context-free valence grammar that generates the set of all braid words equal to the given braid β.

Lemma 10. *Given a braid word $\beta \in B_3$, there exists a context-free valence grammar G over the alphabet $\Sigma = \{\sigma_1, \sigma_2, \Delta, \Delta^{-1}\}$ such that $L(G)$ is the set of all braid words over Σ which are equal* *to β.*

Proof. First, we convert the given braid β into the Garside normal form $\beta^+\Delta^k$. Note that $k \in \mathbb{Z}$ is an integer and $\beta^+ = \sigma_{i_1}\sigma_{i_2}\sigma_{i_3}\cdots\sigma_{i_n} \in B_3^+$ is a positive braid of length n where $i_j = \{1,2\}$ for $1 \leq j \leq n$.

We define a valence grammar $G = (N, \Sigma, R, S)$, where

- $N = \{S\} \cup \{S_1, S_2, \ldots, S_{n-1}\} \cup \{\overline{S_1}, \overline{S_2}, \ldots, \overline{S_{n-1}}\} \cup \{A_{\mathsf{even}}, A_{\mathsf{odd}}, A_{\mathsf{any}}\}$ is a finite set of nonterminals,
- $\Sigma = \{\sigma_1, \sigma_2, \Delta, \Delta^{-1}\}$ is a set of terminals,
- R is a finite set of productions, and
- S is the axiom.

We define R to contain the following production rules:

- $S \xrightarrow{-k} A_{\mathsf{even}}\sigma_{i_1}S_1 \mid A_{\mathsf{odd}}\mathfrak{r}(\sigma_{i_1})\overline{S_1}$,
- $S_j \xrightarrow{0} A_{\mathsf{even}}\sigma_{i_{j+1}}S_{j+1} \mid A_{\mathsf{odd}}\mathfrak{r}(\sigma_{i_{j+1}})\overline{S_{j+1}}$ for $1 \leq j \leq n-2$,
- $\overline{S_j} \xrightarrow{0} A_{\mathsf{even}}\mathfrak{r}(\sigma_{i_{j+1}})\overline{S_j} \mid A_{\mathsf{odd}}\sigma_{i_{j+1}}S_{j+1}$ for $1 \leq j \leq n-2$,
- $S_{n-1} \xrightarrow{0} A_{\mathsf{even}}\sigma_{i_n}A_{\mathsf{any}} \mid A_{\mathsf{odd}}\mathfrak{r}(\sigma_{i_n})A_{\mathsf{any}}$,
- $\overline{S_{n-1}} \xrightarrow{0} A_{\mathsf{even}}\mathfrak{r}(\sigma_{i_n})A_{\mathsf{any}} \mid A_{\mathsf{odd}}\sigma_{i_n}A_{\mathsf{any}}$,
- $A_{\mathsf{even}} \xrightarrow{0} \varepsilon \mid A_{\mathsf{even}}A_{\mathsf{even}} \mid A_{\mathsf{odd}}A_{\mathsf{odd}}$,
- $A_{\mathsf{even}} \xrightarrow{1} \sigma_2A_{\mathsf{odd}}\sigma_2A_{\mathsf{even}}\sigma_1 \mid \sigma_1A_{\mathsf{even}}\sigma_2A_{\mathsf{odd}}\sigma_2 \mid \sigma_1A_{\mathsf{odd}}\sigma_1A_{\mathsf{even}}\sigma_2 \mid \sigma_2A_{\mathsf{even}}\sigma_1A_{\mathsf{odd}}\sigma_1$,
- $A_{\mathsf{odd}} \xrightarrow{1} \Delta$,
- $A_{\mathsf{odd}} \xrightarrow{-1} \Delta^{-1}$,
- $A_{\mathsf{odd}} \xrightarrow{0} A_{\mathsf{even}}A_{\mathsf{odd}} \mid A_{\mathsf{odd}}A_{\mathsf{even}}$,
- $A_{\mathsf{odd}} \xrightarrow{1} \sigma_1A_{\mathsf{even}}\sigma_2A_{\mathsf{even}}\sigma_1 \mid \sigma_1A_{\mathsf{odd}}\sigma_1A_{\mathsf{odd}}\sigma_1 \mid \sigma_2A_{\mathsf{even}}\sigma_1A_{\mathsf{even}}\sigma_2 \mid \sigma_2A_{\mathsf{odd}}\sigma_2A_{\mathsf{odd}}\sigma_2$, and
- $A_{\mathsf{any}} \xrightarrow{0} A_{\mathsf{even}} \mid A_{\mathsf{odd}}$.

Note that we can derive every braid word corresponding to the Δ^k where k is an even (respectively, odd) integer from the nonterminal A_{even} (respectively, A_{odd}). In particular, the following derivation relation holds:

$$(A_{\mathsf{even}}, 0) \overset{*}{\Rightarrow} (\omega, k),$$

where ω is a braid word corresponding to Δ^k for an even integer k. Similarly, A_{odd} can be replaced by every braid word corresponding to Δ^m where m is an odd integer.

Now it remains to prove that the valence grammar G actually generates every braid word which is equal to the given braid β by the relations of the braid group B_3. First, we show that the every braid word generated by G is equal to the given braid β. Since the 'S'-nonterminals should be substituted by regular type productions of G (containing at most one 'S'-nonterminal on the right-hand side) to derive words consisting of terminals, we see that the following derivation should be performed in any case:

$$(S, 0) \overset{*}{\Rightarrow} (A_0 \mathfrak{r}^{p(1)}(\sigma_{i_1}) A_1 \mathfrak{r}^{p(2)}(\sigma_{i_2}) A_2 \cdots A_{n-1} \mathfrak{r}^{p(n)}(\sigma_{i_n}) A_n, -k), \tag{2}$$

where

$$p(x) = \begin{cases} 0, & \text{if } |\{k \mid 0 \leq k < x, \ \ A_k = A_{\mathsf{odd}}\}| \equiv 0 \mod 2, \\ 1, & \text{otherwise.} \end{cases}$$

In other words, $p(x)$ has a value of 0 if the number of A_{odd} appearing in front of the xth terminal symbol of the positive braid β^+ is even. Now we move the word generated by 'A'-nonterminals to the right by Equation (1). After moving every 'A'-nonterminals to the right, we obtain the following braid word:

$$\sigma_{i_1} \sigma_{i_2} \cdots \sigma_{i_n} A_0 A_1 \cdots A_n.$$

It is easy to see that 'A'-nonterminals can be substituted by some braid word which is equal to Δ^k as claimed above, and therefore, we prove that any braid word generated by G is equal to the given braid β.

Lastly, we show that the valence grammar G generates every braid word equal to the braid β. First, we define the following two sets

- $C_{\mathsf{even}} = \{\omega \mid \omega = \Delta^k, k \equiv 0 \mod 2\}$ and
- $C_{\mathsf{odd}} = \{\omega \mid \omega = \Delta^k, k \equiv 1 \mod 2\}$

such that C_{even} (respectively, C_{odd}) is the set of all braid words equal to the composition of an even (respectively, odd) number of the Garside element Δ. Then, every braid word equal to β is captured by the following set of words:

$$C_0 \cdot \{\mathfrak{r}^{p'(1)}(\sigma_{i_1})\} \cdot C_1 \cdot \{\mathfrak{r}^{p'(2)}(\sigma_{i_2})\} \cdot C_2 \cdots C_{n-1} \cdot \{\mathfrak{r}^{p'(n)}(\sigma_{i_n})\} \cdot C_n,$$

where

$$p'(x) = \begin{cases} 0, & \text{if } |\{k \mid 0 \leq k < x, \ \ C_k = C_{\mathsf{odd}}\}| \equiv 0 \mod 2, \\ 1, & \text{otherwise.} \end{cases}$$

Following the derivation described in (2), we can see that every braid word equal to β can be derived by the valence grammar G. □

Now we are ready to present our NP algorithm for the membership problem in the braid group B_3.

Lemma 11. *The membership problem can be decided in* NP *for braids from* B_3.

Proof. Let us suppose that we are given a set $B = \{\beta_1, \beta_2, \ldots, \beta_n\}$ of braid words and a braid word β for which we need to decide whether β can be generated by the set B. We first convert the given braids β_i for $1 \leq i \leq n$ into the unique Garside normal form $\Delta^{k_i}\beta_i^+$ where $k_i \in \mathbb{N}$ is an integer and β_i^+ is a positive braid word by Lemma 6.

Let us define the regular language $L_B = \{\Delta^{k_i}\beta_i^+ \mid 1 \leq i \leq n\}^+$ over the alphabet $\{\sigma_1, \sigma_2, \Delta, \Delta^{-1}\}$. Clearly, L_B should contain a braid word β' which is equal to β by the relations of the braid group B_3 if and only if the given set B of braid words generates the target braid β.

By Lemma 10, there exists a valence grammar G generating every braid word in B_3 equal to the target braid β and definable over the alphabet $\{\sigma_1, \sigma_2, \Delta, \Delta^{-1}\}$. Therefore, the problem of checking whether β can be generated by the set B reduces to the problem of checking whether the intersection of L_B and $L(G)$ is empty.

It is known that we can convert a given valence grammar over $\mathbb{Z}^k$ into a pushdown automaton (PDA) equipped with k additional blind counters in polynomial time [16]. Since we are using an integer weight of dimension one, the valence grammar G can be converted into a PDA with a blind counter in polynomial time and construct a new PDA with a blind counter recognizing the intersection $L_B \cap L(G)$ by constructing the Cartesian product of two automata. It is easy to see that the resulting automaton is still a PDA with a blind counter of size polynomial in the input size. By Lemma 4, we conclude that the membership problem in the braid group B_3 can be decided in NP. □

Following Lemma 9 and Lemma 11, we establish the following complexity result for the membership problem in the braid group B_3.

Theorem 1. *The membership problem for braids from the braid group B_3 is* NP*-complete.*

3.3 Freeness Problem in the Braid Group B_3

In the proof of Lemma 11, we construct a finite state automaton recognizing the regular language L_B with n multi-states loops representing braid words in Garside normal form from the set B. Then, a path from the initial state to itself in this automaton represents a braid that can be constructed by a semigroup generator $\{\beta_1, \beta_2, \dots, \beta_n\}$. Note that the rest of the proof is not based on the structure of this automata and the same algorithm can be applied to check the membership for any other finite graph, where labels are braids from B_3. Hence, we can immediately establish the following result.

Corollary 2. *Given a directed graph G with labels from the braid group B_3, u and v are two nodes from G and β is a braid from B_3. Then, the problem of deciding whether exists a path P from u and v such that a direct sum of braids on labels along a path P is isotopic to a braid β can be decided in* NP*.*

Note that the NP algorithm for the membership problem can exploited for decidability of the freeness problem in braid semigroups with generators from B_3.

Theorem 2. *The freeness problem for braids from the braid group B_3 can be decided in* NP*.*

Proof. Let us consider a set $B = \{\beta_1, \beta_2, \dots \beta_n\}$ of braids from B_3 and a braid semigroup $\langle B \rangle$ which is finitely generated by the set B. If the semigroup $\langle B \rangle$ is not free, then there are two products of the form $A_1 \cdot X \cdot A_2$ and $C_1 \cdot Y \cdot C_2$ such that

$$A_1 \cdot X \cdot A_2 = C_1 \cdot Y \cdot C_2, \tag{3}$$

where $A_1 \neq C_1$, $A_2 \neq C_2$, $A_1, A_2, C_1, C_2 \in B$, and $X, Y \in \langle B \rangle$.

Now it is not difficult to see that we can check whether the semigroup $\langle B \rangle$ is free if we can decide whether there exist two products as in Equation (3) since we can iteratively run the same procedure for each pairs of braids from the set B. Indeed, we can decide whether there exist two products as in Equation (3) for the chosen braids A_1, A_2, C_1, C_2 from the set B by checking whether the following equation can be satisfied for some $X, Y \in \langle B \rangle$:

$$A_1 X A_2 {C_2}^{-1} Y^{-1} {C_1}^{-1} = I.$$

Then, we can construct a finite-state automaton recognizing all the sequences of braids of the form on the left-hand side of the equation and further construct an automaton that recognizes the following regular language over braids from B_3:

$$L_B = \{A_1 w_1 A_2 C_2^{-1} w_2^{-1} C_1^{-1} \mid A_1 \neq C_1, \;\; A_2 \neq C_2, \;\; A_1, A_2, C_1, C_2 \in B, \\ w_1, w_2 \in B^*\}.$$

It should be noted that the construction of the automaton recognizing L_B takes polynomial time. We can see that the braid semigroup $\langle B \rangle$ is not free if and only if the regular language L_B contains any braid word corresponding to the trivial braid, which can be checked in NP by Corollary 2. Hence, we conclude that the freeness problem for braid semigroups in B_3 can be decided in NP. □

4 Undecidability of Decision Problems in the Braid Group B_5

The composition problems become harder with a larger number of strands. Since the braids group B_5 contain the direct product of two free groups, it is possible to show that most of the composition problems are undecidable in B_5. We first provide the following property of B_5 which will be used later in the undecidability results in B_5.

Lemma 12 ([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 $d = \sigma_4\sigma_3\sigma_2\sigma_1^2\sigma_2\sigma_3\sigma_4$.*

We can prove the undecidability of the identity problem and the group problem by relying on the embedding from B_5 into the direct product of two free groups.

Theorem 3. *The identity problem and the group problem are undecidable for braids in B_5.*

Proof. Bell and Potapov [5] has proven the undecidability of the *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. Let $\Sigma = \{a, b\}$ be a binary alphabet and $\Pi = \{(s_1, t_1), (s_2, t_2), \ldots, (s_m, t_m)\} \subseteq \mathrm{FG}(\Sigma) \times \mathrm{FG}(\Sigma)$. Formally speaking, the ICP is 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 ε is the empty word (identity).

We can directly use the Lemma 12 to encode the ICP in terms of braid words. We shall use a straightforward encoding to embed an instance of the ICP into a set of braids. Let $\Pi \subseteq \Sigma^* \times \Sigma^*$ be an instance of the ICP where $\Sigma = \{a, b, a^{-1}, b^{-1}\}$ generates a free group. Define two morphisms ϕ and ψ that map Σ into B_5 as follows:

$$\begin{aligned} \phi(a) &= {\sigma_1}^4, & \phi(a^{-1}) &= {\sigma_1}^{-4}, \\ \phi(b) &= {\sigma_2}^4, & \phi(b^{-1}) &= {\sigma_2}^{-4}. \\ \psi(a) &= {\sigma_4}^2, & \psi(a^{-1}) &= {\sigma_4}^{-2}, \\ \psi(b) &= \sigma_4\sigma_3\sigma_2\sigma_1^2\sigma_2\sigma_3\sigma_4, & \psi(b^{-1}) &= \sigma_4^{-1}\sigma_3^{-1}\sigma_2^{-1}\sigma_1^{-2}\sigma_2^{-1}\sigma_3^{-1}\sigma_4^{-1}. \end{aligned}$$

The domain of ϕ and ψ can be naturally extended to words as follows:

$$\phi(w_1 \ldots w_i) = \phi(w_1) \cdot \ldots \cdot \phi(w_i); \quad \psi(v_1 \ldots v_j) = \psi(v_1) \cdot \ldots \cdot \psi(v_j),$$

where $w_1 \cdots w_i, v_1 \cdots v_j \in \Sigma^*$. For each pair of words $(s, t) \in W$, define the braid word $\phi(s) \cdot \psi(t)$. Let S be a braid semigroup generated by these braid words. In other words, S is finitely generated by the set $\{\phi(s) \cdot \psi(t) \mid (s, t) \in \Pi\}$. If there exists a solution to the ICP, then we see that $\phi(\varepsilon) \cdot \psi(\varepsilon) = 1 \in$

S, where 1 is the trivial braid. Otherwise, the trivial braid does not exist in the braid semigroup S since ψ and ϕ are injective homomorphisms. Therefore, 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 identity problem is also computationally equivalent to the following problem which is called the group problem. Given a semigroup generated by a finite set of pairs of words (over a group alphabet), can we decide whether the semigroup is a group? Using the same morphisms ϕ and ψ, we can encode the group problem for words by braids, having that the group problem for braids in B_5 is also undecidable. □

Similarly, we also prove that the freeness problem is undecidable in B_5.

Theorem 4. *The freeness problem for braids from the braid group B_5 is undecidable.*

Proof. We first introduce the *mixed modification PCP* (MMPCP) [11] which is already proven to be undecidable and prove the undecidability of the freeness problem in B_5 by encoding an instance of the MMPCP.

Given a finite alphabet Σ, a binary alphabet Δ, and a pair of homomorphisms $h, g : \Sigma^* \to \Delta^*$, the MMPCP asks to decide whether or not there exists a word $w = a_1 \dots a_k \in \Sigma^+, a_i \in \Sigma$ such that

$$h_1(a_1)h_2(a_2)\dots h_k(a_k) = g_1(a_1)g_2(a_2)\dots g_k(a_k),$$

where $h_i, g_i \in \{h, g\}$ and for some $j \in [1, k]$ such that $h_j \neq g_j$.

Let $\Sigma = \{a_1, a_2, \dots, a_{n-2}\}$ and $\Delta = \{a_{n-1}, a_n\}$ be disjoint alphabets and $h, g : \Sigma^* \to \Delta^*$ be an instance of the MMPCP. Now define a morphism $\gamma : (\Sigma \cup \Delta)^* \times (\Sigma \cup \Delta)^* \to B_5$ by

$$\gamma(u, v) = \phi(u) \cdot \psi(v).$$

It is easy to see that γ is a homomorphism since $\gamma(u_1, v_1)\gamma(u_2, v_2) = \gamma(u_1u_2, v_1v_2)$. Now let S be a braid semigroup which is finitely generated by the set $\{\gamma(a_i, h(a_i)), \gamma(a_i, g(a_i)) \mid a_i \in \Sigma, 1 \leq i \leq n-2\}$. The braid semigroup S is not free if and only if the MMPCP instance has a solution. Since the MMPCP is undecidable, we conclude that the freeness problem in the braid group B_5 is also undecidable. □

5 Conclusion

The paper introduces 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 have shown that the membership problem for B_3 is decidable and actually NP-complete. The NP-hardness result is in line with the best current knowledge about similar problem in the special linear group $\mathrm{SL}(2, \mathbb{Z})$. W Finally in this paper we have proven 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 second author is grateful for many fruitful discussions with Sergei Chmutov and Victor Goryunov on the computational problems in topology.

References

[1] A. M. Akimenkov. Subgroups of the braid group B_4. *Mathematical notes of the Academy of Sciences of the USSR*, 50(6):1211–1218, 1991.

[2] T. Ang, G. Pighizzini, N. Rampersad, and J. Shallit. Automata and reduced words in the free group. *CoRR*, abs/0910.4555, 2009.

[3] P. C. Bell, M. Hirvensalo, and I. Potapov. Mortality for 2 × 2 matrices is np-hard. In *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science*, MFCS 2012, pages 148–159, 2012.

[4] P. C. Bell, M. Hirvensalo, and I. Potapov. The identity problem for matrix semigroups in $\mathrm{SL}(2,\mathbb{Z})$ is NP-complete. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms 2017*, SODA 2017, pages 187–206, 2017.

[5] P. C. Bell and I. Potapov. On the undecidability of the identity correspondence problem and its applications for word and matrix semigroups. *International Journal of Foundations of Computer Science*, 21(6):963–978, 2010.

[6] P. C. Bell and I. Potapov. On the computational complexity of matrix semigroup problems. *Fundamenta Infomaticae*, 116(1-4):1–13, 2012.

[7] V. Bezverkhnij and I. Dobrynina. On the unsolvability of the conjugacy problem for subgroups of the group R_5 of pure braids. *Mathematical Notes*, 65(1):13–19, 1999.

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

[9] A. Bovykin and L. Carlucci. Long games on braids, 2006. Available online at `http://logic.pdmi.ras.ru/~andrey/braids_final3.pdf`.

[10] L. Carlucci, P. Dehornoy, and A. Weiermann. Unprovability results involving braids. *Proceedings of the London Mathematical Society*, 102(1):159–192, 2011.

[11] J. Cassaigne, J. Karhumäki, and T. Harju. On the decidability of the freeness of matrix semigroups. Technical report, Turku Center for Computer Science, 1996.

[12] D. Chistikov, C. Haase, and S. Halfon. Context-free commutative grammars with integer counters and resets. *Theoretical Computer Science*, 2016. In press.

[13] C. Choffrut and J. Karhumäki. Some decision problems on integer matrices. *RAIRO - Theoretical Informatics and Applications*, 39(1):125–131, 3 2010.

[14] P. Dehornoy, I. Dynnikov, D. Rolfsen, and B. Wiest. *Ordering Braids*. Mathematical surveys and monographs. American Mathematical Society, 2008.

[15] D. B. A. Epstein, M. S. Paterson, J. W. Cannon, D. F. Holt, S. V. Levy, and W. P. Thurston. *Word Processing in Groups*. A. K. Peters, Ltd., 1992.

[16] H. Fernau and R. Stiebe. Sequential grammars and automata with valances. *Theoretical Computer Science*, 276(1-2):377–405, 2002.

[17] D. Garber. Braid group cryptography. *CoRR*, abs/0711.3941, 2007.

[18] D. Garber. Braid group cryptography. In *Braids: Introductory Lectures on Braids, Configurations and Their Applications*, pages 329–403. World Scientific Publishing Company, 2010.

[19] F. A. Garside. The braid group and other groups. *The Quarterly Journal of Mathematics*, 20(1):235–254, 1969.

[20] Y. Gurevich and P. Schupp. Membership problem for the modular group. *SIAM Journal on Computing*, 37(2):425–459, 2007.

[21] V. Halava, T. Harju, R. Niskanen, and I. Potapov. Weighted automata on infinite words in the context of attacker-defender games. *Information and Computation*, 2017. Submitted.

[22] H. J. Hoogeboom. Context-free valence grammars - revisited. In *Proceedings of the 5th International Conference on Developments in Language Theory*, pages 293–303, 2002.

[23] S. J. Lomonaco and L. H. Kauffman. Quantizing braids and other mathematical structures: the general quantization procedure. In *Proceedings of SPIE - The International Society for Optical Engineering*, volume 8057, pages 805702–805702–14, 2011.

[24] K. Mahlburg. As overview of braid group cryptography, 2004. Available online at `http://www.math.wisc.edu/~boston/mahlburg.pdf`.

[25] A. Myasnikov, V. Shpilrain, and A. Ushakov. A practical attack on a braid group based cryptographic protocol. In *Proceedings of the 25th Annual International Cryptology Conference Advances in Cryptology*, CRYPTO 2005, pages 86–96, 2005.

[26] A. Myasnikov, V. Shpilrain, and A. Ushakov. Random subgroups of braid groups: An approach to cryptanalysis of a braid group based cryptographic protocol. In *Proceedings of the 9th International Conference on Theory and Practice in Public-Key Cryptography*, PKC 2006, pages 302–314, 2006.

[27] S. Orevkov. Quasipositivity problem for 3-braids. *Turkish Journal of Mathematics*, 28(1):89–94, 2004.

[28] M. Paterson and A. Razborov. The set of minimal braids is co-np-complete. *Journal of Algorithms*, 12(3):393–408, 1991.

[29] I. Potapov. Composition Problems for Braids. In *Proceedings of the 33rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, FSTTCS 2013, pages 175–187, 2013.