Advances in Mathematics **139**, 322–353 (1998)
Article No. AI981761

A New Approach to the Word and Conjugacy Problems in the Braid Groups

Joan Birman*

Department of Mathematics, Columbia University, New York, New York 10027; Barnard College of Columbia University
E-mail: jb@math.columbia.edu

and

Ki Hyoung Ko† and Sang Jin Lee

Department of Mathematics, Korea Advanced Institute of Science and Technology, Taejon 305-701, Korea
E-mail:knot@knot.kaist.ac.kr; sangjin@knot.kaist.ac.kr

Received November 23, 1997; accepted April 30, 1998

A new presentation of the n-string braid group B_n is studied. Using it, a new solution to the word problem in B_n is obtained which retains most of the desirable features of the Garside–Thurston solution, and at the same time makes possible certain computational improvements. We also give a related solution to the conjugacy problem, but the improvements in its complexity are not clear at this writing.
© 1998 Academic Press

1. INTRODUCTION

In the foundational manuscript [3] Emil Artin introduced the sequence of braid groups B_n, $n=1, 2, 3, \ldots$ and proved that B_n has a presentation with $n-1$ generators $\sigma_1, \sigma_2, \ldots, \sigma_{n-1}$ and defining relations:

$$\sigma_t \sigma_s = \sigma_s \sigma_t \qquad \text{if} \quad |t-s| > 1. \tag{1}$$

$$\sigma_t \sigma_s \sigma_t = \sigma_s \sigma_t \sigma_s \qquad \text{if} \quad |t-s| = 1. \tag{2}$$

* Partially supported by NSF Grant 94-02988. This paper was completed during a visit by the first author to MSRI. She thanks MSRI for its hospitality and partial support, and thanks Barnard College for its support under a Senior Faculty Research Leave.

† This work was initiated during the second author's sabbatical visit to Columbia University in 1995-6. He thanks Columbia University for its hospitality during that visit.

0001-8708/98 $25.00
Copyright © 1998 by Academic Press
All rights of reproduction in any form reserved.

The word problem in B_n was solved by Artin in [3]. His solution was based on his knowledge of the structure of the kernel of the map ϕ from B_n to the symmetric group Σ_n which sends the generator σ_i to the transposition $(i, i+1)$. He used the group-theoretic properties of the kernel of ϕ to put a braid into a normal form called a "combed braid". While nobody has investigated the matter, it seems intuitively clear that Artin's solution is exponential in the length of a word in the generators $\sigma_1, \ldots, \sigma_{n-1}$.

The conjugacy problem in B_n was also posed in [3], also its importance for the problem of recognizing knots and links algorithmically was noted, however it took 43 years before progress was made. In a different, but equally foundational manuscript [9] F. Garside discovered a new solution to the word problem (very different from Artin's) which then led him to a related solution to the conjugacy problem. In Garside's solution one focusses not on the kernel of ϕ, but on its image, the symmetric group Σ_n. Garside's solutions to both the word and conjugacy problem are exponential in both word length and braid index.

The question of the speed of Garside's algorithm for the word problem was first raised by Thurston. His contributions, updated to reflect improvements obtained after his widely circulated preprint appeared, are presented in Chapter 9 of [8]. In [8] Garside's algorithm is modified by introducing new ideas, based upon the fact that braid groups are biautomatic, also that B_n has a partial ordering which gives it the structure of a lattice. Using these facts it is proved in [8] that there exists an algorithmic solution to the word problem which is $\mathcal{O}(|W|^2 n \log n)$, where $|W|$ is word length. See, in particular, Proposition 9.5.1 of [8], our discussion at the beginning of Sect. 4 below, and Remark 4.2 in Sect. 4. While the same general set of ideas apply equally well to the conjugacy problem [7], similar sharp estimates of complexity have not been found because the combinatorics are very difficult.

A somewhat different question is the *shortest word problem*, to find a representative of the word class which has shortest length in the Artin generators. It was proved in [13] that this problem in B_n is at least as hard as an NP-complete problem. Thus, if one could find a polynomial time algorithm to solve the shortest word problem one would have proved that $P = NP$.

Our contribution to this set of ideas is to introduce a new and very natural set of generators for B_n which includes the Artin generators as a subset. Using the new generators we will be able to solve the word problem in much the same way as Garside and Thurston solved it, moreover our solution generalizes to a related solution to the conjugacy problem which is in the spirit of that of [7]. The detailed combinatorics in our work are, however, rather different from those in [7] and [8]. Our algorithm solves the word problem in $\mathcal{O}(|W|^2 n)$. Savings in actual running time (rather than complexity) also occur, because a word written in our generators is generally shorter by a factor of n than a word in the standard generators which represents the same element. (Each generator a_{ts}, in our work replaces a word of length $2(t-s)-1$, where $n>t-s>0$ in the Artin generators). Also the positive part is shorter by a factor of n because the new generators lead to a new and shorter "fundamental word" δ which replaces Garside's famous Δ.

Our solution to both the word and conjugacy problems generalizes the work of Xu [17] and of Kang, Ko and Lee [10], who succeeded in finding polynomial time algorithms for the word and conjugacy problems and also for the shortest word problem in B_n for $n=3$ and 4. The general case appears to be more subtle than the cases $n=3$ and 4, however polynomial time solutions to the three problems for every n do not seem to be totally out of reach, using our generators.

In the three references [7], [8] and [9] a central role is played by *positive braids*, i.e. braids which are positive powers of the generators. Garside introduced the *fundamental braid* Δ:

$$\Delta=(\sigma_1\sigma_2\cdots\sigma_{n-1})(\sigma_1\sigma_2\cdots\sigma_{n-2})\cdots(\sigma_1\sigma_2)(\sigma_1). \tag{3}$$

He showed that every element $\mathscr{W}\in B_n$ can be represented algorithmically by a word W of the form $\Delta^r P$, where r is an integer and P is a positive word, and r is maximal for all such representations. However his P is non-unique up to a finite set of equivalent words which represent the same element $\mathscr{P}$. These can all be found algorithmically, but the list is very long. Thus instead of a unique normal form one has a fixed r and a finite set of positive words which represent $\mathscr{P}$. Thurston's improvement was to show that $\mathscr{P}$ can in fact be factorized as a product $P_1P_2\cdots P_s$, where each P_j is a special type of positive braid which is known as "permutation braid". Permutation braids are determined uniquely by their associated permutations, and Thurston's normal form is a unique representation of this type in which the integer s is minimal for all representations of $\mathscr{P}$ as a product of permutation braids. Also, in each subsequence $P_iP_{i+1}\cdots P_s$, $i=1, 2, ..., s-1$, the permutation braid P_i is the longest possible permutation braid in a factorization of this type. The subsequent work of Elrifai and Morton [7] showed that there is a related algorithm which simultaneously maximizes r and minimizes s within each conjugacy class. The set of all products $P_1P_2\cdots P_s$ which do that job (the *super summit set*) is finite, but it is not well understood.

Like Artin's, our generators are braids in which exactly one pair of strands crosses, however the images of our generators in Σ_n are *arbitrary* transpositions (s, t) instead of simply adjacent transpositions $(s, s+1)$. For each t, s with $n \geqslant t > s \geqslant 1$ we consider the element of B_n which is defined by:

$$a_{ts} = (\sigma_{t-1}\sigma_{t-2} \cdots \sigma_{s+1})\, \sigma_s(\sigma_{s+1}^{-1} \cdots \sigma_{t-2}^{-1}\sigma_{t-1}^{-1}). \tag{4}$$

The braid a_{ts}, is depicted in Figure 1(a). Notice that a_{21}, a_{321}... coincide with $\sigma_1, \sigma_2, \ldots$. The braid a_{ts}, is an elementary interchange of the tth and sth strands, with all other strands held fixed, and with the convention that the strands being interchanged pass in front of all intervening strands. We call them *band generators* because they suggest a disc-band decomposition of a surface bounded by a closed braid. Such decompositions have been studied extensively by L. Rudolph in several papers, e.g., see [15]. The bands which we use are his "embedded bands."

We introduce a new *fundamental word*:

$$\delta = a_{n(n-1)} a_{(n-1)(n-2)} \cdots a_{21} = \sigma_{n-1}\sigma_{n-2} \cdots \sigma_2\sigma_1. \tag{5}$$

The reader who is familiar with the mathematics of braids will recognize that $\Delta^2 = \delta^n$ generates the center of B_n. Thus Δ may be thought of as the "square root" of the center, whereas δ is the "nth root" of the center. We will prove that each element $\mathcal{W} \in B_n$ may be represented (in terms of the band generators) by a unique word W of the form:

$$W = \delta^j A_1 A_2 \cdots A_k, \tag{6}$$

where $A = A_1 A_2 \cdots A_k$ is positive, also j is maximal and k is minimal for all such representations, also the A_i's are positive braids which are determined uniquely by their associated permutations. We will refer to Thurston's braids P_i as *permutation braids*, and to our braids A_i as *canonical factors*.

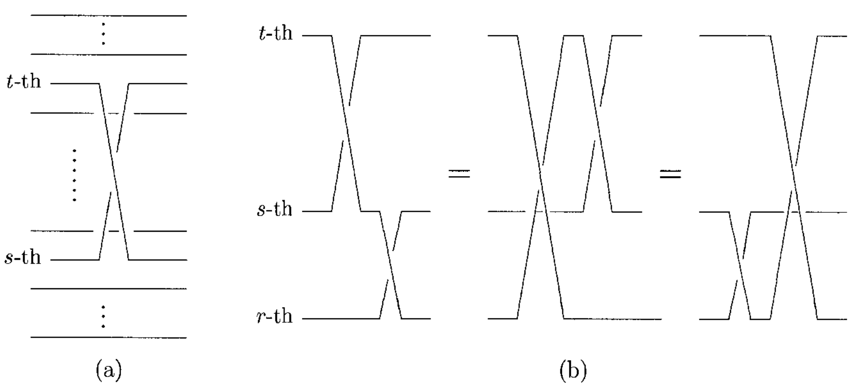

FIG. 1. The band generators and relations between them.

Let $\mathscr{W}$ be an arbitrary element of B_n and let W be a word in the band generators which represents it. We are able to analyze the speed of our algorithm for the word problem, as a function of both the word length $|W|$ and braid index n. Our main result is a new algorithmic solution to the word problem (see Sect. 4 below). Its computational complexity, which is analysed carefully in Sect. 4 of this paper, is an improvement over that given in [8] which is the best among the known algorithms. Moreover our work offers certain other advantages, namely:

1. The number of distinct permutation braids is $n!$, which grows faster than k^n for any $k \in \mathbb{R}^+$. The number of distinct canonical factors is the nth Catalan number $\mathscr{C}_n = (2n)!/n!\,(n+1)!$, which is bounded above by 4^n. The reason for this reduction is the fact that the canonical factors can be decomposed nicely into parallel, descending cycles (see Theorem 3.4). The improvement in the complexity of the word algorithm is a result of the fact that the canonical factors are very simple. We think that they reveal beautiful new structure in the braid group.

2. Since our generators include the Artin generators, we may assume in both cases that we begin with a word W of length $|W|$ in the Artin generators. Garside's Δ has length $(n-1)(n-2)/2$, which implies that the word length $|P|$ of the positive word $P = P_1 P_2 \cdots P_q$ is roughly $n^2\,|W|$. On the other hand, our δ has word length $n-1$, which implies that the length $|A|$ of the product $A = A_1 A_2 \cdots A_k$ is roughly $n\,|W|$.

3. Our work, like that in [8], generalizes to the conjugacy problem. We conjecture that our solution to that problem is polynomial in word length, a matter which we have not settled at this writing.

4. Our solution to the word problem suggests a related solution to the shortest word problem.

5. It has been noted in conversations with A. Ram that our work ought to generalize to other Artin groups with finite Coxeter groups. This may be of interest in its own right.

Here is an outline of the paper. In Sect. 2 we find a presentation for B_n in terms of the new generators and show that there is a natural semigroup B_n^+ of positive words which is determined by the presentation. We prove that every element in B_n can be represented in the form $\delta^t A$, where A is a positive word. We then prove (by a long computation) that B_n^+ embeds in B_n, i.e. two positive words in B_n represent the same element of B_n if and only if their pullbacks to B_n^+ are equal in B_n^+. We note (see Remark 2.8) that our generators and Artin's are the only ones in a class studied in [16] for which such an embedding theorm holds. In Sect. 3 we use these ideas to find normal forms for words in B_n^+, and so also for words in B_n. In Sect. 4 we give our algorithmic solution to the word problem and study its complexity. In Sect. 5 we describe very briefly how our work generalizes to the conjugacy problem.

Remark 1.1. In the article [6] P. Dehornoy gives an algorithmic solution to the word problem which is based upon the existence, proved in a different paper by the same author, of an order structure on B_n. His methods seem quite different from ours and from those in the other papers we have cited, and not in a form where precise comparisons are possible. Dehornoy does not discuss the conjugacy problem, and indeed his methods do not seem to generalize to the conjugacy problem.

2. THE SEMIGROUP OF POSITIVE BRAIDS

We begin by finding a presentation for B_n in terms of the new generators. We will use the symbol a_{ts}, whenever there is no confusion about the two subscripts, and symbols such as $a_{(t+2)(s+1)}$ when there might be confusion distinguishing between the first and second subscripts. Thus $a_{(t+1)t} = \sigma_t$.

PROPOSITION 2.1. *B_n has a presentation with generators $\{a_{ts}; n \geqslant t > s \geqslant 1\}$ and with defining relations*

$$a_{ts} a_{rq} = a_{rq} a_{ts} \qquad \text{if} \quad (t-r)(t-q)(s-r)(s-q) > 0 \tag{7}$$

$$a_{ts} a_{sr} = a_{tr} a_{ts} = a_{sr} a_{tr} \qquad \text{for all} \quad t, s, r \quad \text{with } n \geqslant t > s > r \geqslant 1. \tag{8}$$

Remark 2.2. Relation (7) asserts that a_{ts} and a_{rq} commute if t and s do not separate r and q. Relation (8) expresses a type of "partial" commutativity in the case when a_{ts}, and a_{rq} share a common strand. It tells us that if the product $a_{ts} a_{sr}$ occurs in a braid word, then we may move a_{ts} to the right (resp. move a_{sr} to the left) at the expense of increasing the first subscript of a_{sr} to t (resp. decreasing the second subscript of a_{ts} to r.)

Proof. We begin with Artin's presentation for B_n, using generators $\sigma_1, ..., \sigma_{n-1}$ and relations (1) and (2). Add the new generators a_{ts}, and the relations (4) which define them in terms of the σ_i's. Since we know that relations (7) and (8) are described by isotopies of braids, depicted in Figure 1(b), they must be consequences of (1) and (2), so we may add them too.

In the special case when $t = s + 1$ relation (4) tells us that $a_{(i+1)i} = \sigma_i$, so we may omit the generators $\sigma_1, ..., \sigma_{n-1}$, to obtain a presentation with generators a_{ts}, as described in the theorem. Defining relations are now (7), (8) and:

$$a_{(t+1)t}a_{(s+1)s} = a_{(s+1)s}a_{(t+1)t} \qquad \text{if} \quad |t-s|>1 \tag{9}$$

$$a_{(t+1)t}a_{(s+1)s}a_{(t+1)t} = a_{(s+1)s}a_{(t+1)t}a_{(s+1)s} \qquad \text{if} \quad |t-s|=1 \tag{10}$$

$$a_{ts} = (a_{t(t-1)}a_{(t-1)(t-2)}\cdots a_{(s+2)(s+1)})\, a_{(s+1)s} \times (a_{t(t-1)}a_{(t-1)(t-2)}\cdots a_{(s+2)(s+1)})^{-1} \tag{11}$$

Our task is to prove that (9), (10) and (11) are consequences of (7) and (8).

Relation (9) is nothing more than a special case of (7). As for (10), by symmetry we may assume that $t=s+1$. Use (8) to replace $a_{(s+2)(s+1)}a_{(s+1)s}$ by $a_{(s+1)s}a_{(s+2)s}$, thereby reducing (10) to $a_{(s+2)s}a_{(s+2)(s+1)} = a_{(s+2)(s+1)}$ $a_{(s+1,s)}$, which is a special case of (8). Finally, we consider (11). If $t=s+1$ this relation is trivial, so we may assume that $t>s+1$. Apply (8) to change the center pair $a_{(s+2)(s+1)}a_{(s+2)s}$ to $a_{(s+2)s}a_{(s+2)(s+1)}$. If $t>s+2$ repeat this move on the new pair $a_{(s+3)(s+2)}a_{(s+2)s}$. Ultimately, this process will move the original center letter $a_{(s+1)s}$ to the leftmost position, where it becomes a_{ts}. Free cancellation eliminates everything to its right, and we are done. ∎

A key feature which the new presentation shares with the old is that the relations have all been expressed as relations between positive powers of the generators, also the relations all preserve word length. Thus our presentation also determines a presentation for a semigroup. A word in positive powers of the generators is called a *positive word*. Two positive words are said to be *positively equivalent* if one can be transformed into the other by a sequence of positive words such that each word of the sequence is obtained from the preceding one by a single direct application of a defining relation in (7) or (8). For two positive words X and Y, write $X \doteq Y$ if they are positively equivalent. Positive words that are positively equivalent have the same word length since all of defining relations preserve the word length. We use the symbol B_n^+ for the monoid of positive braids, which can be defined by the generators and relations in Theorem 2.1. Thus B_n^+ is the set of positive words modulo positive equivalence. Our next goal is to prove that the principal theorem of [9] generalizes to our new presentation, i.e. that the monoid of positive braids embeds in the braid group B_n. See Theorem 2.7 below.

Before we can begin we need to establish key properties of the fundamental braid δ. Let τ be the inner automorphism of B_n which is induced by conjugation by δ.

LEMMA 2.3 *Let δ be the fundamental braid. Then*:

(I) $\delta = a_{n(n-1)}a_{(n-1)(n-2)}\cdots a_{21}$ *is positively equivalent to a word that begins or ends with any given generator* a_{ts}, $n \geqslant t > s \geqslant 1$. *The explicit expressions are*:

$$\delta \doteq (a_{ts})(a_{n(n-1)} \cdots a_{(t+2)(t+1)} a_{(t+1)s} a_{s(s-1)} \cdots a_{21})(a_{t(t-1)} \cdots a_{(s+2)(s+1)})$$

$$\delta \doteq (a_{n(n-1)} a_{(n_{-1})(n-2)} \cdots a_{(t+1)t} \times a_{t(s-1)} a_{(s-1)(s-2)} \cdots a_{21})(a_{(t-1)(t-2)} \cdots a_{(s+1)s})(a_{ts})$$

(II) *Let* $A = a_{n_m n_{m-1}} a_{n_{m-1} n_{m-2}} \cdots a_{n_2 n_1}$ *where* $n \geqslant n_m > n_{m-1} > \cdots > n_1 \geqslant 1$. *Then A is positively equivalent to a word which begins or ends with* $a_{n_t n_s}$, *for any choice of* n_t, n_s *with* $n \geqslant n_t > n_s \geqslant 1$.

(III) $a_{ts}\delta \doteq \delta a_{(t+1)(s+1)}$, *where subscripts are defined mod n.*

Proof. (I) With Remark 2.2 in mind, choose any pair of indices t, s with $n \geqslant t > s \geqslant 1$. We need to show that δ can be represented by a word that begins with a_{ts}. Focus first on the elementary braid $a_{(s+1)s}$ in the expression for δ which is given in (5), and apply the first of the pair of relations in (8) repeatedly to move $a_{(s+1)s}$ to the left (increasing its first index as you do so) until its name changes to a_{ts}. Then apply the second relation in the pair to move it (without changing its name) to the extreme left end, vis:

$$\begin{aligned}
\delta &\doteq a_{n(n-1)} a_{(n-1)(n-2)} \cdots a_{(s+2)(s+1)} a_{(s+1)s} a_{s(s-1)} \cdots a_{21} \\
&\doteq a_{n(n-1)} a_{(n-1)(n-2)} \cdots a_{(t+1)t} a_{ts} a_{t(t-1)} \cdots a_{(s+2)(s+1)} a_{s(s-1)} \cdots a_{21} \\
&\doteq a_{ts} a_{n(n-1)} \cdots a_{(t+2)(t+1)} a_{(t+1)s} a_{t(t-1)} \cdots a_{(s+2)(s+1)} a_{s(s-1)} \cdots a_{21} \\
&\doteq (a_{ts})(a_{n(n-1)} \cdots a_{(t+2)(t+1)} a_{(t+1)s} a_{s(s-1)} \cdots a_{21})(a_{t(t-1)} \cdots a_{(s+2)(s+1)})
\end{aligned}$$

We leave it to the reader to show that the proof works equally well when we move letters to the right instead of to the left.

(II) The proof of (II) is a direct analogy of the proof of (I).

(III) To establish (III), we use (I):

$$\begin{aligned}
&\delta a_{(t+1)(s+1)} \\
&\doteq a_{ts} a_{n(n-1)} \cdots a_{(t+2)(t+1)} a_{(t+1)s} a_{t(t-1)} \cdots a_{(s+2)(s+1)} a_{s(s-1)} \cdots a_{21} a_{(t+1)(s+1)} \\
&\doteq a_{ts} a_{n(n-1)} \cdots a_{(t+2)(t+1)} a_{(t+1)s} a_{(t+1)t} a_{t(t-1)} \cdots a_{(s+2)(s+1)} a_{s(s-1)} \cdots a_{21}) \\
&\doteq a_{ts} a_{n(n-1)} \cdots a_{(t+2)(t+1)} a_{(t+1)t} a_{t(t-1)} \cdots a_{(s+2)(s+1)} a_{(s+1)s} a_{s(s-1)} \cdots a_{21}) \\
&\doteq a_{ts}\, \delta. \quad \blacksquare
\end{aligned}$$

We move on to the main business of this section, the proof that the semigroup B_n^+ embeds in B_n. We will use Lemma 2.3 in the following way:

the inner automorphism defined by conjugation by δ determines an index-shifting automorphism τ of B_n and B_n^+ which is a useful tool to eliminate repetitious arguments. We define:

$$\tau(a_{ts}) = a_{(t+1)(s+1)} \qquad \text{and} \qquad \tau^{-1}(a_{ts}) = a_{(t-1)(s-1)}.$$

Following the ideas which were first used by Garside [9], the key step is to establish that there are right and left cancellation laws in B_n^+. We remark that even though Garside proved this for Artin's presentation, it does not follow that it's still true when one uses the band generator presentation. Indeed, counterexamples were discovered by Xu [17] and given in [10].

If $X \doteq Y$ is obtained by a sequence of t single applications of the defining relations in (7) and (8):

$$X \equiv W_0 \to W-1 \to \cdots \to W_t \equiv Y,$$

then the transformation which takes X to Y will be said to be of *chain-length* t.

THEOREM 2.4 (Left "cancellation"). *Let $a_{ts}X \doteq a_{rq}Y$ for some positive words X, Y. Then X and Y are related as follows*:

(I) *If there are only two distinct indices, i.e. $t = r$ and $s = q$, then $X \doteq Y$.*

(II) *If there are three distinct indices*:

(i) *If $t = r$ and $q < s$, then $X \doteq a_{sq}Z$ and $Y \doteq a_{ts}Z$ for some $Z \in B_n^+$*,

(ii) *If $t = r$ and $s < q$, then $X \doteq a_{tq}Z$ and $Y \doteq a_{qs}Z$ for some $Z \in B_n^+$*,

(iii) *If $t = q$, then $X \doteq a_{rs}Z$ and $Y \doteq a_{ts}Z$ for some $Z \in B_n^+$*,

(iv) *If $s = r$, then $X \doteq a_{sq}Z$ and $Y \doteq a_{tq}Z$ for some $Z \in B_n^+$*,

(v) *If $s = q$ and $r < t$, then $X \doteq a_{tr}Z$ and $Y \doteq a_{ts}Z$ for some $Z \in B_n^+$*,

(vi) *If $s = q$ and $t < r$, then $X \doteq a_{rs}Z$ and $Y \doteq a_{rt}Z$ for some $Z \in B_n^+$*,

(III) *If the four indices are distinct and if $(t-r)(t-q)(s-r)(s-q) > 0$, then $X \doteq a_{rq}Z$ and $Y \doteq a_{ts}Z$ for some $Z \in B_n^+$.*

(IV) *If the four indices are distinct, then*:

(i) *If $q < s < r < t$, then $X \doteq a_{tr}a_{sq}Z$ and $Y \doteq a_{tq}a_{rs}Z$ for some $Z \in B_n^+$*,

(ii) *If $s < q < t < r$, then $X \doteq a_{tq}a_{rs}Z$ and $Y \doteq a_{rt}a_{qs}Z$ for some $Z \in B_n^+$*, *Proof.* The proof of the theorem for positive words X, Y of word length j that are positively equivalent via a transformation of chain-length k will be referred to as $T(j, k)$. The proof will be proceeded by an induction on (j, k) ordered lexicographically. This induction makes sense because $T(j, 1)$ holds for any j. Assume that $T(j, k)$ holds for all pairs $(j, k) < (l, m)$, that is,

$(*)$ $T(j, k)$ is true for $0 \leqslant j \leqslant l-1$ and any k.

$(**)$ $T(l, k)$ is true for $k \leqslant m-1$.

Now suppose that X and Y are positive words of length l and $a_{ts}X \doteq a_{rq}Y$ via a transformation of chain-length $m \geqslant 2$. Let $a_{\beta\alpha}W$ be the first intermediate word in the sequence of transformation from $a_{ts}X$ to $a_{rq}Y$. We can assume that $a_{\beta\alpha} \neq a_{ts}$ and $a_{\beta\alpha} \neq a_{rs}$, otherwise we apply the induction hypotheses $(**)$ to complete the proof. Furthermore, since $a_{\beta\alpha}W$ must be obtained from $a_{ts}X$ by a single application of a defining relation, we see by using $(**)$ that $X \doteq aU$ and $W \doteq bU$ for some distinct generators a, b and a positive word U.

For case (I), we see again by using $(**)$ that $W \doteq bV$ and $Y \doteq aV$ for a positive word V. Then $W \doteq bU \doteq bV$ implies $U \doteq V$ by $(*)$. Thus $X \doteq aU \doteq aV \doteq Y$.

It remains to prove cases II, III, IV(i) and IV(ii). We fix notation as follows: Using $(**)$, $W \doteq BV$ and $Y \doteq AV$ for a positive word V and two distinct positive words A, B of word length 1 or 2 depending on $a_{\beta\alpha}$ and a_{rq}. When the word length of A and B is 1, we apply $(*)$ to $W \doteq bU \doteq BV$. If $b = B$, $U \doteq V$ and so $X \doteq aU$ and $Y \doteq AU$ are the required form. If $b \neq B$, we obtain $U \doteq CQ$ and $V \doteq DQ$ for a positive word Q and two distinct word C and D of word length 1 or 2 depending on b and B. Then we apply some defining relations to $X \doteq aCQ$ and $Y \doteq ADQ$ to achieve the desired forms.

When the word lengths of A and B are 2, we rewrite $B = b'b''$ in terms of generators. Notice that $b'b'' = b''b'$. Apply $(*)$ to either $W \doteq bU \doteq b'(b''V)$ or $W \doteq bU \doteq b''(b'V)$ to obtain $U \doteq CQ$ and either $b''V \doteq DQ$ or $b'V \doteq DQ$ for a positive word Q and two distinct generators C and D. In the tables below, we will use the symbols so that we have $b''V \doteq DQ$. Apply $(*)$ to $b''V \doteq DQ$. If $b'' = D$, we obtain $V \doteq Q$ and then we apply defining relations to $X \doteq aCV$ and $Y \doteq AV$ to get the required forms. If $b'' \neq D$, we obtain $V \doteq EP$ and $Q \doteq FP$ for a positive word P and distinct words E, F of word length 1 or 2 depending on b'' and D. Then we apply some defining relations to $X \doteq aCFP$ and $Y \doteq AEP$ to achieve the desired forms.

The four tables below treat cases II, III, IV(i) and IV(ii). The first column covers the possible relative positions of $q, r, s, t, \alpha, \beta$. The second column contains one of 4 possible forms aU, aCQ, aCV, $aCFP$ of the word

X as explained above and similarly the third column contains one of 4 possible forms AU, ADQ, AV, AEP of Y. Finally the fourth and fifth columns contain the values of b and B, respectively.

In case (II), it is enough to consider the subcases (i) and (ii) because the other subcases can be obtained from (i) or (ii) by applying the automorphism τ. But we may also assume that $q < s$, otherwise we switch the roles of X and Y.

In case (III), there are actually 4 possible positions of q, r, s, t but they can be obtained from one position by applying τ. Thus we only consider case $q < r < s < t$. The table shows all possible cases required in the induction.

In case (IV), it is again enough to consider the case $q < s < r < t$ and the table covers all possible inductive steps.

	X	Y	b	B
$q<s<t=r<\alpha<\beta$	$a_{\beta\alpha}a_{sq}Q \doteq a_{sq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q<s<t=r=\alpha<\beta$	$a_{\beta s}a_{sq}Q \doteq a_{sq}a_{\beta q}Q$	$a_{\beta q}a_{ts}Q \doteq a_{ts}a_{\beta q}Q$	a_{ts}	a_{tq}
$q<s=\alpha<t=r<\beta$	$a_{\beta s}a_{sq}V \doteq a_{sq}a_{\beta q}V$	$a_{ts}a_{\beta q}V \doteq a_{ts}a_{\beta q}V$	$a_{\beta t}$	$a_{\beta t}a_{sq}$
$q<\alpha<s<t=r<\beta$	$a_{\beta\alpha}a_{\beta s}a_{\alpha q}P$ $\doteq a_{sq}a_{s\alpha}a_{\beta q}P$	$a_{t\alpha}\beta_{\beta q}a_{ts}P$ $\doteq a_{ts}a_{s\alpha}a_{\beta q}P$	a_{ts}	$a_{\beta t}a_{\alpha q}$
$q=\alpha<s<t=r<\beta$	$a_{\beta\alpha}a_{\beta s}Q \doteq a_{sq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	$a_{\beta t}$
$q<s<\alpha<t=r=\beta$	$a_{t\alpha}a_{sq}Q \doteq a_{sq}a_{t\alpha}Q$	$a_{t\alpha}a_{\alpha s}Q \doteq a_{ts}a_{t\alpha}Q$	$a_{\alpha s}$	$a_{\alpha q}$
$q<\alpha<s<t=r=\beta$	$a_{s\alpha}a_{\alpha q}Q \doteq a_{sq}a_{s\alpha}Q$	$a_{t\alpha}a_{ts}Q \doteq a_{ts}a_{s\alpha}Q$	a_{ts}	$a_{\alpha q}$
$\alpha<q<s<t=r=\beta$	$a_{s\alpha}a_{sq}Q \doteq a_{sq}a_{q\alpha}Q$	$a_{q\alpha}a_{ts}Q \doteq a_{ts}a_{q\alpha}Q$	a_{ts}	a_{tq}
$q<s<\alpha<\beta<t=r$	$a_{\beta\alpha}a_{sq}Q \doteq a_{sq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	a_{tq}
$q<s=\alpha<\beta<t=r$	$a_{t\beta}a_{sq}Q \doteq a_{sq}a_{t\beta}Q$	$a_{\beta s}a_{ts}Q \doteq a_{ts}a_{t\beta}Q$	a_{ts}	a_{tq}
$q<\alpha<s=\beta<t=r$	$a_{s\alpha}a_{\alpha q}Q \doteq a_{sq}a_{s\alpha}Q$	$a_{s\alpha}a_{t\alpha}Q \doteq a_{ts}a_{s\alpha}Q$	$a_{t\alpha}$	a_{tq}
$q=\alpha<s=\beta<t=r$	$a_{sq}U$	$a_{ts}U$	a_{tq}	a_{tq}
$\alpha<q<s=\beta<t=r$	$a_{s\alpha}a_{sq}V \doteq a_{sq}a_{q\alpha}V$	$a_{ts}a_{q\alpha}V \doteq a_{ts}a_{q\alpha}V$	$a_{t\alpha}$	$a_{t\alpha}a_{sq}$
$q<\alpha<\beta<s<t=r$	$a_{\beta\alpha}a_{sq}Q \doteq a_{sq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	a_{tq}
$q=\alpha<\beta<s<t=r$	$a_{\beta q}a_{sq}Q \doteq a_{sq}a_{s\beta}Q$	$a_{t\beta}a_{ts}Q \doteq a_{ts}a_{s\beta}Q$	a_{ts}	a_{tq}
$\alpha<q<\beta<s<t=r$	$a_{\beta\alpha}a_{\beta q}a_{s\alpha}P$ $\doteq a_{sq}a_{s\beta}a_{q\alpha}P$	$a_{t\beta}\beta_{q\alpha}a_{ts}P$ $\doteq a_{ts}a_{s\beta}a_{q\alpha}P$	a_{ts}	$a_{\beta q}a_{t\alpha}$
$\alpha<q=\beta<s<t=r$	$a_{q\alpha}a_{s\alpha}Q \doteq a_{sq}a_{q\alpha}Q$	$a_{q\alpha}a_{ts}Q \doteq a_{ts}a_{q\alpha}Q$	a_{ts}	$a_{t\alpha}$
$q<r<s<t<\alpha<\beta$	$a_{\beta\alpha}a_{rq}Q \doteq a_{rq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q<r<s<t=\alpha<\beta$	$a_{\beta s}a_{rq}Q \doteq a_{rq}a_{\beta s}Q$	$a_{\beta t}a_{ts}Q \doteq a_{ts}a_{\beta s}Q$	a_{ts}	a_{rq}
$q<r<s=\alpha<t<\beta$	$a_{\beta s}a_{rq}Q \doteq a_{rq}a_{\beta s}Q$	$a_{\beta s}a_{\beta t}Q \doteq a_{ts}a_{\beta s}Q$	$a_{\beta t}$	a_{rq}
$q<r<\alpha<s<t<\beta$	$a_{\beta\alpha}a_{rq}Q \doteq a_{rq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q<r=\alpha<s<t<\beta$	$a_{\beta r}a_{rq}Q \doteq a_{rq}a_{\beta q}Q$	$a_{\beta q}a_{ts}Q \doteq a_{ts}a_{\beta q}Q$	a_{ts}	a_{rq}
$q<\alpha<r<s<t<\beta$	$a_{\beta\alpha}a_{\beta r}a_{\alpha q}P$ $\doteq a_{rq}a_{\beta q}a_{r\alpha}P$	$a_{\beta q}a_{r\alpha}a_{ts}P$ $\doteq a_{ts}a_{\beta q}a_{r\alpha}P$	a_{ts}	$a_{\beta r}a_{\alpha q}$
$q=\alpha<r<s<t<\beta$	$a_{\beta q}a_{\beta r}Q \doteq a_{rq}a_{\beta q}Q$	$a_{\beta q}a_{ts}Q \doteq a_{ts}a_{\beta q}Q$	a_{ts}	$a_{\beta r}$
$q<r<s<\alpha<t=\beta$	$a_{t\alpha}a_{rq}Q \doteq a_{rq}a_{t\alpha}Q$	$a_{t\alpha}a_{\alpha s}Q \doteq a_{ts}a_{t\alpha}Q$	$a_{\alpha s}$	a_{rq}
$q<r<\alpha<s<t=\beta$	$a_{s\alpha}a_{rq}Q \doteq a_{rq}a_{s\alpha}Q$	$a_{t\alpha}a_{ts}Q \doteq a_{ts}a_{s\alpha}Q$	a_{ts}	a_{rq}
$q<r=\alpha<s<t=\beta$	$a_{sr}a_{rq}Q \doteq a_{rq}a_{sq}Q$	$a_{tq}a_{ts}Q \doteq a_{ts}a_{sq}Q$	a_{ts}	a_{rq}
$q<\alpha<r<s<t=\beta$	$a_{s\alpha}a_{sr}a_{\alpha q}P$ $\doteq a_{rq}a_{sq}a_{r\alpha}P$	$a_{tq}a_{r\alpha}a_{ts}P$ $\doteq a_{ts}a_{sq}a_{r\alpha}P$	a_{ts}	$a_{tr}a_{\alpha q}$

$q=\alpha<r<s<t=\beta$	$a_{sq}a_{sr}Q \doteq a_{rq}a_{sq}Q$	$a_{tq}a_{ts}Q \doteq a_{ts}a_{sq}Q$	a_{ts}	a_{tr}
$\alpha<q<r<s<t=\beta$	$a_{s\alpha}a_{rq}Q \doteq a_{rq}a_{s\alpha}Q$	$a_{t\alpha}a_{ts}Q \doteq a_{ts}a_{s\alpha}Q$	a_{ts}	a_{rq}
$q<r<s<\alpha<\beta<t$	$a_{\beta\alpha}a_{rq}Q \doteq a_{rq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q<r<s=\alpha<\beta<t$	$a_{t\beta}a_{rq}Q \doteq a_{rq}a_{t\beta}Q$	$a_{\beta s}a_{ts}Q \doteq a_{ts}a_{t\beta}Q$	a_{ts}	a_{rq}
$q<r<\alpha<s=\beta<t$	$a_{s\alpha}a_{rq}Q \doteq a_{rq}a_{s\alpha}Q$	$a_{s\alpha}a_{t\alpha}Q \doteq a_{ts}a_{s\alpha}Q$	$a_{t\alpha}$	a_{rq}
$q<r=\alpha<s=\beta<t$	$a_{sr}a_{rq}Q \doteq a_{rq}a_{sq}Q$	$a_{sq}a_{tq}Q \doteq a_{ts}a_{sq}Q$	a_{tr}	a_{rq}
$q<\alpha<r<s=\beta<t$	$a_{s\alpha}a_{sr}a_{\alpha q}P$ $\doteq a_{rq}a_{sq}a_{r\alpha}P$	$a_{sq}a_{r\alpha}a_{tq}P$ $\doteq a_{ts}a_{sq}a_{r\alpha}P$	$a_{t\alpha}$	$a_{sr}a_{\alpha q}$
$q=\alpha<r<s=\beta<t$	$a_{sq}a_{sr}Q \doteq a_{rq}a_{sq}Q$	$a_{sq}a_{tq}Q \doteq a_{ts}a_{sq}Q$	a_{tq}	a_{sr}
$\alpha<q<r<s=\beta<t$	$a_{s\alpha}a_{rq}Q \doteq a_{rq}a_{s\alpha}Q$	$a_{s\alpha}a_{t\alpha}Q \doteq a_{ts}a_{s\alpha}Q$	$a_{t\alpha}$	a_{rq}
$q<r<\alpha<\beta<s<t$	$a_{\beta\alpha}aQ \doteq a_{rq}a_{rq}Q$	$a_{\beta\alpha}a_{\beta\alpha}Q \doteq a_{ts}a_{rq}Q$ts.	a_{ts}	a_{rq}
$q<r=\alpha<\beta<s<t$	$a_{\beta r}a_{rq}Q \doteq a_{rq}a_{\beta q}Q$	$a_{\beta q}a_{ts}Q \doteq a_{ts}a_{\beta q}Q$	a_{ts}	a_{rq}
$q<\alpha<r<\beta<s<t$	$a_{\beta\alpha}a_{\beta r}a_{\alpha q}P$ $\doteq a_{rq}a_{\beta q}a_{r\alpha}P$	$a_{\beta q}a_{r\alpha}a_{ts}P$ $\doteq a_{ts}a_{\beta q}a_{r\alpha}P$	a_{ts}	$a_{\beta r}a_{\alpha q}$
$q=\alpha<r<\beta<s<t$	$a_{\beta q}a_{\beta r}Q \doteq a_{rq}a_{\beta q}Q$	$a_{\beta q}a_{ts}Q \doteq a_{ts}a_{\beta q}Q$	a_{ts}	$a_{\beta r}$
$\alpha<q<r<\beta<s<t$	$a_{\beta\alpha}a_{rq}Q \doteq a_{rq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q<\alpha<r=\beta<s<t$	$a_{r\alpha}a_{\alpha q}Q \doteq a_{rq}a_{r\alpha}Q$	$a_{r\alpha}a_{ts}Q \doteq a_{ts}a_{r\alpha}Q$	a_{ts}	$a_{\alpha q}$
$\alpha<q<r=\beta<s<t$	$a_{r\alpha}a_{rq}Q \doteq a_{rq}a_{q\alpha}Q$	$a_{q\alpha}a_{ts}Q \doteq a_{ts}a_{q\alpha}Q$	a_{ts}	a_{rq}
$q<\alpha<\beta<r<s<t$	$a_{\beta\alpha}a_{rq}Q \doteq a_{rq}a_{\beta\alpha}Q$	$a_{\beta\alpha}a_{ts}Q \doteq a_{ts}a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q=\alpha<\beta<r<s<t$	$a_{\beta q}a_{rq}Q \doteq a_{rq}a_{r\beta}Q$	$a_{r\beta}a_{ts}Q \doteq a_{ts}a_{r\beta}Q$	a_{ts}	a_{rq}
$\alpha<q<\beta<r<s<t$	$a_{\beta\alpha}a_{r\alpha}a_{\beta q}P$ $\doteq a_{rq}a_{r\beta}a_{q\alpha}P$	$a_{r\beta}a_{q\alpha}a_{ts}P$ $\doteq a_{ts}a_{r\beta}a_{q\alpha}P$	a_{ts}	$a_{r\alpha}a_{\beta q}$
$\alpha<q=\beta<r<s<t$	$a_{q\alpha}a_{r\alpha}Q \doteq a_{rq}a_{q\alpha}Q$	$a_{q\alpha}a_{ts}Q \doteq a_{ts}a_{q\alpha}Q$	a_{ts}	$a_{r\alpha}$
$q<s<r<t<\alpha<\beta$	$a_{\beta\alpha}(a_{tr}a_{sq})Q$ $\doteq (a_{tr}a_{sq})a_{\beta\alpha}Q$	$a_{\beta\alpha}(a_{tq}a_{rs})Q$ $\doteq (a_{rs}a_{tq})a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q<s<r<t=\alpha<\beta$	$a_{\beta s}(a_{tr}a_{sq})Q$ $\doteq (a_{tr}a_{sq})a_{\beta q}Q$	$a_{\beta t}(a_{tq}a_{rs})Q$ $\doteq (a_{rs}a_{tq})a_{\beta q}Q$	a_{ts}	a_{rq}
$q<s=\alpha<r<t<\beta$	$a_{\beta s}a_{tr}a_{\alpha q}P$ $\doteq (a_{tr}a_{sq})a_{\beta q}P$	$a_{\beta q}a_{rs}a_{\beta t}P$ $\doteq (a_{rs}a_{tq})a_{\beta q}P$	$a_{\beta t}$	$a_{\beta r}a_{\alpha q}$
$q<\alpha<s<r<t<\beta$	$a_{\beta\alpha}a_{\beta s}(a_{tr}a_{\alpha q})P$ $\doteq (a_{tr}a_{sq})a_{\beta q}a_{s\alpha}P$	$a_{\beta q}a_{r\alpha}a_{\beta t}a_{rs}P$ $\doteq (a_{rs}a_{il})a_{\beta q}a_{s\alpha}P$	a_{ts}	$a_{\beta r}a_{\alpha q}$
$q=\alpha<s<r<t<\beta$	$a_{\beta q}(a_{\beta s}a_{tr})Q$ $\doteq (a_{tr}a_{sq})a_{\beta q}Q$	$a_{\beta q}(a_{\beta t}a_{rs})Q$ $\doteq (a_{rs}a_{tq})a_{\beta q}Q$	a_{ts}	$a_{\beta r}$
$q<s<r<\alpha<t=\beta$	$a_{t\alpha}(a_{\alpha r}a_{sq})Q$ $\doteq (a_{tr}a_{sq})a_{t\alpha}Q$	$a_{t\alpha}(a_{\alpha q}a_{rs})Q$ $\doteq (a_{rs}a_{tq})a_{t\alpha}Q$	$a_{\alpha s}$	a_{rq}
$q<s<r=\alpha<t=\beta$	$a_{tr}a_{sq}Q$	$a_{rs}a_{tq}Q$	a_{rs}	a_{rq}
$q<s<\alpha<r<t=\beta$	$a_{t\alpha}a_{tr}a_{sq}P$ $\doteq (a_{tr}a_{sq})a_{r\alpha}P$	$a_{tq}a_{r\alpha}a_{\alpha s}P$ $\doteq (a_{rs}a_{tq})a_{r\alpha}P$	$a_{\alpha s}$	$a_{tr}a_{\alpha q}$
$q<\alpha<s<r<t=\beta$	$a_{s\alpha}a_{tr}a_{\alpha q}P$ $\doteq (a_{tr}a_{sq})a_{s\alpha}P$	$a_{tq}a_{r\alpha}a_{rs}P$ $\doteq (a_{rs}a_{tq})a_{s\alpha}P$	a_{ts}	$a_{tr}a_{\alpha q}$
$q=\alpha<s<r<t=\beta$	$a_{tr}a_{sq}Q$	$a_{rs}a_{tq}Q$	a_{ts}	a_{tr}
$\alpha<q<s<r<t=\beta$	$a_{s\alpha}(a_{tr}a_{sq})Q$ $\doteq (a_{tr}a_{sq})a_{q\alpha}Q$	$a_{t\alpha}(a_{\beta q}a_{rs})Q$ $\doteq (a_{rs}a_{tq})a_{q\alpha}Q$	a_{ts}	a_{rq}
$q<s<r<\alpha<\beta<t$	$a_{\beta\alpha}(a_{tr}a_{sq})Q$ $\doteq (a_{tr}a_{sq})a_{\beta\alpha}Q$	$a_{\beta\alpha}(a_{tq}a_{rs})Q$ $\doteq (a_{rs}a_{tq})a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q<s<r=\alpha<\beta<t$	$a_{\beta r}(a_{tr}a_{sq})Q$ $\doteq (a_{tr}a_{sq})a_{t\beta}Q$	$a_{\beta q}(a_{tq}a_{rs})Q$ $\doteq (a_{rs}a_{tq})a_{t\beta}Q$	a_{ts}	a_{rq}

$q<s<\alpha<r<\beta<t$	$a_{\beta\alpha}a_{\beta r}(a_{t\alpha}a_{sq})P$ $\doteq(a_{tr}a_{sq})a_{t\beta}a_{r\alpha}P$	$a_{\beta q}a_{r\alpha}a_{tq}a_{\alpha s}P$ $\doteq(a_{rs}a_{il})a_{t\beta}a_{r\alpha}P$	a_{ts}	$a_{\beta r}a_{\alpha q}$
$q<s=\alpha<r<\beta<t$	$a_{t\beta}a_{\beta r}a_{sq}P$ $\doteq(a_{tr}a_{sq})a_{t\beta}P$	$a_{\beta q}a_{rs}a_{tq}P$ $\doteq(a_{rs}a_{tq})a_{t\beta}P$	a_{ts}	$a_{\beta r}a_{\alpha q}$
$q<s<\alpha<r=\beta<t$	$a_{r\alpha}(a_{t\alpha}a_{sq})Q$ $\doteq(a_{tr}a_{sq})a_{r\alpha}Q$	$a_{r\alpha}(a_{tq}a_{\alpha s})Q$ $\doteq(a_{rs}a_{tq})a_{r\alpha}Q$	a_{ts}	a_{rq}
$q<s=\alpha<r=\beta<t$	$a_{tr}a_{sq}Q$	$a_{rs}a_{tq}Q$	a_{ts}	a_{sq}
$q<s<\alpha<\beta<r<t$	$a_{\beta\alpha}(a_{tr}a_{sq})Q$ $\doteq(a_{tr}a_{sq})a_{\beta\alpha}Q$	$a_{\beta\alpha}(a_{tq}a_{rs})Q$ $\doteq(a_{rs}a_{tq})a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q<s=\alpha<\beta<r<t$	$a_{t\beta}(a_{tr}a_{sq})Q$ $\doteq(a_{tr}a_{sq})a_{r\beta}Q$	$a_{\beta s}(a_{tq}a_{rs})Q$ $\doteq(a_{rs}a_{tq})a_{r\beta}Q$	a_{ts}	a_{rq}
$q<\alpha<s=\beta<r<t$	$a_{s\alpha}(a_{tr}a_{\alpha q})Q$ $\doteq(a_{tr}a_{sq})a_{s\alpha}Q$	$a_{s\alpha}(a_{tq}a_{r\alpha})Q$ $\doteq(a_{rs}a_{tq})a_{s\alpha}Q$	$a_{t\alpha}$	a_{rq}
$q=\alpha<s=\beta<r<t$	$a_{tr}a_{sq}Q$	$a_{rs}a_{tq}Q$	a_{tq}	a_{rq}
$\alpha<q<s=\beta<r<t$	$a_{s\alpha}a_{tr}a_{sq}P$ $\doteq(a_{tr}a_{sq})a_{q\alpha}P$	$a_{rs}a_{q\alpha}a_{t\alpha}P$ $\doteq(a_{rs}a_{tq})a_{q\alpha}P$	$a_{t\alpha}$	$a_{rq}a_{sq}$
$q<\alpha<\beta<s<r<t$	$a_{\beta\alpha}(a_{tr}a_{sq})Q$ $\doteq(a_{tr}a_{sq})a_{\beta\alpha}Q$	$a_{\beta\alpha}(a_{rs}a_{tq})Q$ $\doteq(a_{rs}a_{tq})a_{\beta\alpha}Q$	a_{ts}	a_{rq}
$q=\alpha<\beta<s<r<t$	$a_{\beta q}(a_{tr}a_{sq})Q$ $\doteq(a_{tr}a_{sq})a_{s\beta}Q$	$a_{r\beta}(a_{tq}a_{rs})Q$ $\doteq(a_{rs}a_{tq})a_{s\beta}Q$	a_{ts}	a_{rq}
$\alpha<q<\beta<s<r<t$	$a_{\beta\alpha}a_{\beta q}(a_{tr}a_{s\alpha})P$ $\doteq(a_{tr}a_{sq})a_{s\beta}a_{q\alpha}P$	$a_{r\beta}a_{q\alpha}a_{rs}a_{t\alpha}P$ $\doteq(a_{rs}a_{il})a_{s\beta}a_{q\alpha}P$	a_{ts}	$a_{\beta q}a_{r\alpha}$
$\alpha<q=\beta<s<r<t$	$a_{q\alpha}(a_{tr}a_{s\alpha})Q$ $\doteq(a_{tr}a_{sq})a_{q\alpha}Q$	$a_{q\alpha}(a_{t\alpha}a_{rs})Q$ $\doteq(a_{rs}a_{tq})a_{q\alpha}Q$	a_{ts}	$a_{r\alpha}$

This completes the proof that "Left Cancellation" is possible in the monoid of positive words. ∎

Similarly we can prove the following theorem.

THEOREM 2.5 (Right "cancellation"). *Let* $Xa_{ts}\doteq Ya_{rq}$ *for some positive words* X, Y. *Then* X *and* Y *are related as follows*:

(I) *If* $t=r$ *and* $s=q$, *then* $X\doteq Y$,

(II) (i) *If* $t=r$ *and* $q<s$, *then* $X\doteq Za_{tq}$ *and* $Y\doteq Za_{sq}$ *for some* $Z\in B_n^+$,

(ii) *If* $t=r$ *and* $s<q$, *then* $X\doteq Za_{qs}$ *and* $Y\doteq Za_{ts}$ *for some* $Z\in B_n^+$,

(iii) *If* $t=q$, *then* $X\doteq Za_{rt}$ *and* $Y\doteq Za_{ts}$ *for some* $Z\in B_n^+$,

(iv) *If* $s=r$, *then* $X\doteq Za_{tq}$ *and* $Y\doteq Za_{ts}$ *for some* $Z\in B_n^+$,

(v) *If* $s=q$ *and* $r<t$, *then* $X\doteq Za_{rs}$ *and* $Y\doteq Za_{tr}$ *for some* $Z\in B_n^+$,

(vi) *If* $s=q$ *and* $t<r$, *then* $X\doteq Za_{rt}$ *and* $Y\doteq Za_{ts}$ *for some* $Z\in B_n^+$,

(III) *If* $(t-r)(t-q)(s-r)(s-q)>0$, *then* $X \doteq Za_{rq}$ *and* $Y \doteq Za_{ts}$ *for some* $Z \in B_n^+$.

(IV) (i) *If* $q<s<r<t$, *then* $X \doteq Za_{tq}a_{rs}$ *and* $Y \doteq Za_{ts}a_{sq}$ *for some* $Z \in B_n^+$,

(ii) *If* $s<q<t<r$, *then* $X \doteq Za_{rt}a_{qs}$ *and* $Y \doteq Za_{tq}a_{rs}$ *for some* $Z \in B_n^+$,

The properties of δ which were worked out in Lemma 2.3 ensure that δ can take the role of the half twist Δ of the Garside's argument in [9] to show:

THEOREM 2.6 (Right reversibility). *If X, Y are positive words, then there exist positive words U, V such that $UX \doteq VY$.*

Using left and right "cancellation" and right reversibility, we obtain (as did Garside) the following embedding theorem [5].

THEOREM 2.7 (Embedding Theorem). *The natural map from B_n^+ to B_n is injective, that is, if two positive words are equal in B_n, then they are positively equivalent.*

Remark 2.8. Any time that the defining relations in a group presentation are expressed as relations between positive words in the generators one may consider the semigroup of positive words and ask whether that semigroup embeds in the corresponding group. Adjan [1] and also Remmers [14] studied this situation and showed that a semigroup is embeddable if it is "cycle-free", in their terminology. Roughly speaking, this means that the presentation has relatively few relations, so that a positive word can only be written in a small number of ways. But the fundamental words Δ and δ can be written in many many ways, and it therefore follows that large subwords of these words can too, so our presentations are almost the opposite to those considered by Adjan and Remmers.

According to Sergiescu [16], any connected planar graph with n vertices gives rise to a positive presentation of B_n in which each edge gives a generator which is a conjugate of one of Artin's elementary braids and relations are derived at each vertex and at each face. In fact one can generalize his construction as follows. Consider the elements in B_n defined by:

$$b_{ts} = (\sigma_{t-1}^{-1}\sigma_{t-2}^{-1}\cdots\sigma_{s+1}^{-1})\,\sigma_s(\sigma_{s+1}\cdots\sigma_{t-2}\sigma_{t-1}).$$

The braid b_{ts} is geometrically a positive half-twisted band connecting the tth and the sth strands, and passing behind all intermediate strands. Since $b_{t(t-1)} = \sigma_{t-1} = a_{t(t-1)}$, the set $X = \{a_{ts}, b_{ts} \mid \leqslant s < t \leqslant n\}$ contains $(n-1)^2$ elements. Then X may be described by a graph on a plane where n vertices are arranged, in order, on a line. An element $a_{ts}, b_{ts} \in X$ belongs to an edge connecting the tth and the sth vertices on one side or the other, depending on whether it is an a_{ts} or b_{ts}. In this way one obtains a planar graph in which two edges have at most one interior intersection point. It is not hard to show that a subset $Y \subset X$ is a generating set of B_n if and only if the generators in Y form a connected subgraph. Consider all presentations that have $Y \subset X$ as a set of generators and have a finite set of equations between positive words in Y as a set of relators. All Sergiescu's planar graphs are of this type. Artin's presentation corresponds to the linear graph with $n-1$ edges and our presentation corresponds to the complete graph on n vertices. One can prove [12] that the embedding theorem falls to hold in all but two presentations of this type. Those two are Artin's presentation and ours.

3. THE WORD PROBLEM

In this section we present our solution to the word problem in B_n, using the presentation of Proposition 2.1. Our approach builds on the ideas of Garside [9], Thurston [8] and Elrifai and Morton in [7]. In the next section we will translate the results of this section into an algorithm, and compute its complexity.

We begin with a very simple consequence of Lemma 2.3.

LEMMA 3.1. *Every element $\mathscr{W} \in B_n$ can be represented by a word of the form $\delta^p Q$ where p is an integer and Q is a word in the generators $a_{t,s}$ of $B_n^+ \subset B_n$.*

Proof. Choose any word which represents $\mathscr{W}$. Using (I) of Lemma 2.3 replace every generator which occurs with a negative exponent by $\delta^{-1}M$, where M is positive. Then use (III) of Lemma 2.3 to collect the factors δ^{-1} at the left. ∎

The *word length* of a (freely reduced) word W in our presentation of B_n is denoted by $|W|$. The identity word will be denoted by e, $|e|=0$. For words V, W, we write $V \leqslant W$ (or $W \geqslant V$) if $W = P_1 V P_2$ for some $P_1, P_2 \in B_n^+$. Then $W \in B_n^+$ if and only if $e \leqslant W$. Also $V \leqslant W$ if and only if $W^{-1} \leqslant V^{-1}$.

Recall that τ is the inner automorphism of B_n which is defined by $\tau(W) = \delta^{-1} W \delta$. By Lemma 2.3 the action of τ on the generators is given by $\tau(a_{ts}) = a_{(t+1)(s+1)}$.

PROPOSITION 3.2. *The relation "$\leqslant$" has the following properties*:

(I) *"$\leqslant$" is a partial order on B_n.*

(II) *If $W \leqslant \delta^u$, then $\delta^u = PW = W\tau^u(P)$ for some $P \in B_n^+$*

(III) *If $\delta^u \leqslant W$, then $W = P\delta^u = \delta^u \tau^u(P)$ for some $P \in B_n^+$*

(IV) *If* $\delta^{v_1} \leqslant V \leqslant \delta^{v_2}$ *and* $\delta^{w_1} \leqslant W \leqslant \delta^{w_2}$, *then* $\delta^{v_1+w_1} \leqslant VW \leqslant \delta^{v_2+w_2}$.

(V) *For any* W *there exist integers* u, v *such that* $\delta^u \leqslant W \leqslant \delta^v$.

Proof. See [7] or [10]. The proofs given there carry over without any real changes to the new situation. ∎

The set $\{W \mid \delta^u \leqslant W \leqslant \delta^v\}$ is denoted by $[u, v]$. For $\mathscr{W} \in B_n$, the last assertion of the previous proposition enables us to define the *infimum* and the *supremum* of $\mathscr{W}$ as $\inf(\mathscr{W}) = \max\{u \in \mathbb{Z} \mid \delta^u \leqslant W\}$ and $\sup(\mathscr{W}) = \min\{v \in \mathbb{Z} \mid W \leqslant \delta^v\}$, where W represents $\mathscr{W}$. The integer $\ell(\mathscr{W}) = \sup(\mathscr{W}) - \inf(\mathscr{W})$ is called the *canonical length* of $\mathscr{W}$.

A permutation π on $\{1, 2, ..., n\}$ is called a *descending cycle* if it is represented by a cycle $(t_j, t_{j-1}, ..., t_1)$ with $j \geqslant 2$ and $t_j > t_{j-1} > \cdots > t_1$. Given a descending cycle $\pi = (t_j, t_{j-1}, ..., t_1)$, the symbol δ_π denotes the positive braid $a_{t_j t_{j-1}} a_{t_{j-l} t_{j-2}} \cdots a_{t_2 t_1}$. A pair of descending cycles $(t_j, t_{j-1}, ..., t_1)$, $(s_i, s_{i-1}, ..., s_1)$ are said to be *parallel* if t_a and t_b never separate s_c and s_d. That is, $(t_a - s_c)(t_a - s_d)(t_b - s_c)(t_b - s_d) > 0$ for all a, b, c, d with $1 \leqslant a < b \leqslant j$ and $1 \leqslant c < d \leqslant i$. The cycles in a product of parallel, descending cycles are disjoint and non-interlacing. Therefore they commute with one-another. For pairwise parallel, descending cycles $\pi_1, \pi_2, ..., \pi_k$, the factors in the product $\delta_{\pi_1} \delta_{\pi_2} \cdots \delta_{\pi_k}$ are positive braids which commute with one-another and therefore there is a well-defined map from the set of all products of parallel descending cycles to B_n, which splits the homomorphism $\phi\colon B_n \to \Sigma_n$, $\phi(a_{ts}) = (t, s)$.

Our first goal is to prove that braids in $[0, 1]$, i.e. braids A with $e \leqslant A \leqslant \delta$ are precisely the products $\delta_{\pi_1} \delta_{\pi_2} \cdots \delta_{\pi_k}$ as above. We will also prove that each δ_{π_i} is represented by a unique word in the band generators, so that the product A also has a representation which is unique up to the order of the factors.

Let $A = BaCbD$ be a decomposition of the positive word A into subwords, where a, b are generators. Let t, s, r, q be integers, with $n \geqslant t > s > r > q \geqslant 1$. We say that the pair of letters (a, b) is *an obstructing* pair in the following cases:

case (1): $a = a_{tr}$, $b = a_{sq}$
case (2): $a = a_{sq}$, $b = a_{tr}$
case (3): $a = a_{sr}$, $b = a_{ts}$
case (4): $a = a_{ts}$, $b = a_{tr}$
case (5): $a = a_{tr}$, $b = a_{sr}$
case (6): $a = a_{ts}$, $b = a_{ts}$.

LEMMA 3.3. *A necessary condition for a positive word* A *to be in* $[0, 1]$ *is that* A *has no decomposition as* $BaCbD$, *with* $B, a, C, b, D \geqslant e$ *and* (a, b) *an obstructing pair.*

Proof. We use a geometric argument. Given a braid word W in the $a_{t,s}$'s, we associate to W a surface F_W bounded by the closure of W, as follows: F_W consists of n disks joined by half-twisted bands, with a band for each letter in W. The half-twisted band for a_{ts} is the negative band connecting the tth and the sth disks. Our defining relations in (7) and (8) correspond to isotopies sliding a half-twisted band over an adjacent half-twisted band or moving a half-twisted band horizontally. (See Figure 1(b)). Thus defining relations preserve the topological characteristics of F_W. For example the surface F_δ has one connected component and is contractible.

By the proof of Lemma 2.3, we may write $\delta = a_{ts} W$ where $W = \delta_{\pi'} \delta_{\pi''}$ for parallel descending cycles $\pi' = (t, t-1, ..., s+1)$ and $\pi'' = (n, n-1, ..., t+1, s, s-1, ..., 1)$. Thus for this W the surface F_W has two connected components, $F_{\delta'_\pi}$ and $F_{\delta''_\pi}$.

It is enough to consider the cases $(a, b) = (a_{tr}, a_{sq}), (a_{sr}, a_{ts}), (a_{ts}, a_{ts})$ since all other cases are obtained from these cases by applying the automorphism τ, which preserves δ. Since A is in $[0, 1]$ we know that $\delta = V_1 A V_2$ for some $V_1, V_2 \geqslant e$. By Proposition 3.2 (II) we see that $AE = \delta$ for some word $E \geqslant e$. So $BaCbDE = \delta$, which implies that $aCbDE\tau(B) = \delta$. If $a = a_{tr}$, $b = a_{sq}$ for $t > s > r > q$ and $\delta = a_{tr} W$, then F_W has two connected components and the sth disk and the qth disk lie on distinct components. But the sth and the qth disks lie in the same component in $F_{CbDE\tau}(B)$ since they are connected by b and this is a contradiction.

If $a = a_{sr}$, $b = a_{ts}$ and $\delta = a_{sr} W$, then the tth and sth disks lie in distinct components in F_W but they lie in the same component in $F_{CbDE\tau}(B)$ and this is again a contradiction.

If $a = b$, then F_{AE} contains a non-trivial loop but F_δ is contractible and this is a contradiction. ∎

THEOREM 3.4. *A braid word A is in $[0, 1]$ if and only if $A = \delta_{\pi_1} \delta_{\pi_2} \cdots \delta_{\pi_k}$ for some parallel, descending cycles $\pi_1, \pi_2, ..., \pi_k$ in Σ_n.*

Proof. First assume that $A = \delta_{\pi_1} \delta_{\pi_2} \cdots \delta_{\pi_k}$. We induct on the number n of braid strands to prove the necessity. The theorem is true when $n = 2$. Suppose that $\pi_1, \pi_2, ..., \pi_k$ are parallel, descending cycles in Σ_n. In view of the inductive hypothesis, we may assume without loss of generality that the index n appears in one of cycles. Since the factors $\delta_{\pi_1}, ..., \delta_{\pi_j}$ in the product commute with one-another, we may assume that for some $1 \leqslant i \leqslant k$, the cycle $\pi_i = (n, t, ..., s)$, where all of the indices occurring in $\pi_1, ..., \pi_{i-1}$ are greater than t and all of the indices occurring in $\pi_{i+1}, ..., \pi_k$ are less than t. The induction hypothesis implies that

$$C_1 \delta_{\pi_1} \cdots \delta_{\pi_{i-1}} = a_{(n-1)(n-2)} \cdots a_{(t+1)t}$$

$$\delta_{\pi'_i} \delta_{\pi_{i+1}} \cdots \delta_{\pi_k} C_2 = a_{t(t-1)} \cdots a_{21}$$

where $\pi'_i = (t, ..., s)$ and C_1, C_2 are positive words. Thus

$$C_1 A C_2 = a_{(n-1)(n-2)} \cdots a_{(t+1)t} a_{nt} a_{t(t-1)} \cdots a_{21}$$
$$= a_{n(n-1)} a_{(n-1)(n-2)} \cdots a_{21} = \delta.$$

Thus our condition is necessary.

Now assume that A is in $[0, 1]$. We prove sufficiency by induction on the word length of A. The theorem is true when $|A| = 1$. Suppose, then, that $|A| > 1$. Let $A = a_{ts} A'$. By the induction hypothesis $A' = \delta_{\pi_1} \delta_{\pi_2} \cdots \delta_{\pi_k}$ for some parallel, descending cycles $\pi_1, \pi_2, ..., \pi_k$ in Σ_n. Since A is in $[0, 1]$, we know, from Lemma 3.3, that A has no decomposition as $BaCbD$ with (a, b) an obstructing pair, so in particular there is no $a_{rq} \in A'$ such that (a_{ts}, a_{rq}) is an obstructing pair. Therefore, in particular, by cases (1) and (2) for obstructing pairs we must have $(t-r)(t-q)(s-r)(s-q) > 0$ for all $a_{rq} \leqslant A'$. Therefore, if neither t nor s appears among the indices in any of the π_i, then the descending cycle (t, s) is clearly parallel to each π_i and $A = a_{ts} \pi_1 \cdots \pi_k$ is in the desired form.

Suppose that t appears in some $\pi_i = (t_1, t_2, ..., t_m)$. Then, by cases (3) and (4) for obstructing pairs we must have $t = t_1$ and $s < t_m$. Suppose that s appears in some $\pi_j = (s_1, s_2, ..., s_l)$. Then case (5) in our list of obstructing pairs tells us that either $s = s_h$ and $t < s_{h-1}$ for $1 < h \leqslant l$ or $s = s_1$. Thus we have the following three possibilities:

(i) t appears in some $\pi_i = (t_1, t_2, ..., t_m)$ and s does not appear. Then

$$A = \delta_{\pi_1} \cdots \delta_{\pi_{i-1}} \delta_{\pi'_i} \delta_{\pi_{i+1}} \cdots \delta_{\pi_k}$$

is in the desired form, where $\pi'_i = (t_1, t_2, ..., t_m, s)$;

(ii) s appears in some $\pi_j = (s_1, s_2, ..., s_l)$ and t does not appear. Then

$$A = \delta_{\pi_1} \cdots \delta_{\pi_{j-1}} \delta_{\pi'_j} \delta_{\pi_{j+1}} \cdots \delta_{\pi_k}$$

is in the desired form, where $\pi'_j = (s_1, ..., s_{h-1}, t, s_h, ..., s_l)$ or $(t, s_1, ..., s_l)$;

(iii) t appears in some $\pi_i = (t_1, t_2, ..., t_m)$ and s appears in some $\pi_j = (s_1, s_2, ..., s_l)$. Then we may assume $i < j$ and

$$A = \delta_{\pi_1} \cdots \delta_{\pi_{i-1}} \delta_{\pi'_i} \delta_{\pi_{i+1}} \cdots \delta_{\pi_{j-1}} \delta_{\pi_{j+1}} \cdots \delta_{\pi_k}$$

is in the desired form, where $\pi'_i = (t_1, t_2, ..., t_m, s_1, s_2, ..., s_l)$. ∎

Definition. From now on we will refer to a braid which is in $[0, 1]$, and which can therefore be represented by a product of parallel descending cycles, as a *canonical factor*. For example, the 14 distinct canonical factors for $n=4$ are:

$$e,\ a_{21},\ a_{32},\ a_{31},\ a_{43},\ a_{42},\ a_{41},\ a_{32}a_{21},\ a_{43}a_{32},\ a_{43}a_{31},\ a_{43}a_{21},$$

$$a_{42}a_{21},\ a_{41}a_{32},\ a_{43}a_{32}a_{21}.$$

A somewhat simpler notation describes a descending cycle by its subscript array. In the example just given the 13 non-trivial canonical factors are:

$$(21), (32), (31), (43), (42), (41), (321), (432), (431), (421),$$

$$(4321), (43)(21), (41)(32).$$

The associated permutation is the cycle associated to the reverse of the subscript array, with all indices which are not listed explicitely fixed.

COROLLARY 3.5. *For each fixed positive integer n the number of distinct canonical factors is the nth Catalan number $\mathscr{C}_n=(2n)!/n!\,(n+1)!$.*

Proof. We associate to each product $\pi=\pi_1\pi_2\cdots\pi_k$ of parallel descending cycles a set of n disjoint arcs in the upper half-plane whose $2n$ endpoints are on the real axis. Mark the numbers 1, 2, ..., n on the real axis. Join i to $\pi(i)$ by an arc, to obtain n arcs, some of which may be loops. Our arcs have disjoint interiors because the cycles in π are parallel. By construction there are exactly two arcs meeting at each integer point on the real axis. Now split the ith endpoint, $i=1, 2, ..., n$, into two points, i', i'', to obtain n disjoint arcs with $2n$ endpoints. The pattern so obtained will be called an [n]-configuration. To recover the product of disjoint cycles, contract each interval $[i', i'']$ to a single point i. In this way we see that there is a one-to-one correspondence between canonical factors and [n]-configurations. But the number of [n]-configurations is the nth Catalan number (see [11] for a proof). ∎

Note that $\mathscr{C}_n/\mathscr{C}_{n-1}=4-6/(n+1)\leqslant 4$ and so $\mathscr{C}_n\leqslant 4^n$. In the Artin presentation of B_n, the number of permutation braids is $n!$ which is much greater than $\mathscr{C}_n$. This is one of the reasons why our presentation gives a faster algorithm than the algorithm in [8].

It is very easy to recognize canonical factors when they are given as products of parallel descending cycles. If, however, such a representative is modified in some way by the defining relations, we will also need to be able to recognize it. For computational purposes the following alternative characterization of canonical factors will be extremely useful. It rests on Lemma 3.3:

COROLLARY 3.6. *A positive word A is a canonical factor if and only if A contains no obstructing pairs.*

Proof. We established necessity in Lemma 3.3. We leave it to the reader to check that the proof of Theorem 3.4 is essentially a proof of sufficiency. ∎

The braid δ can be written in many different ways as a product of the a_{ts} and by Corollary 3.6 each such product contains no obstructing pair. Any descending cycle δ_π also has this property. If an element in B_n is represented by a word which contains no obstructing pairs, then it is a canonical factor and so it can be written as a product of parallel descending cycles. It follows that there is no obstructing pair in any word representing it.

To get more detailed information about canonical factors $\delta_{\pi_1} \delta_{\pi_2} \cdots \delta_{\pi_k}$, we begin to investigate some of their very nice properties. We proceed as in the foundational paper of Garside [9] and define the *starting* set $S(P)$ and the *finishing* set $F(P)$:

$$S(P) = \{a \mid P = aP',\ P' \geqslant e,\ a \text{ is a generator}\},$$

$$F(P) = \{a \mid P = P'a,\ P' \geqslant e,\ a \text{ is a generator}\}.$$

Note that $S(\tau(P)) = \tau(S(P))$ and $F(\tau(P)) = \tau(F(P))$.

Starting sets play a fundamental role in the solutions to the word and conjugacy problems in [7]. Our canonical form allows us to determine them by inspection.

COROLLARY 3.7. *The starting sets of canonical factors satisfy the following properties*:

(I) *If $\pi = (t_m, t_{m-1}, ..., t_1)$ is a descending cycle, then the starting set (and also the finishing set) of δ_π is $\{a_{t_j t_i};\ m \geqslant j > i \geqslant 1\}$.*

(II) *If $\pi_1, ..., \pi_k$ are parallel descending cycles, then $S(\delta_{\pi_1} \cdots \delta_{\pi_k}) = S(\delta_{\pi_1}) \cup \cdots \cup S(\delta_{\pi_k})$.*

(III) *If A is a canonical factor, then $S(A) = F(A)$.*

(IV) *If A and B are canonical factors, and if $S(A) = S(B)$ then $A = B$.*

(V) *Let P be a given positive word. Then there exists a canonical factor A such that $S(P) = S(A)$.*

(VI) *If $S(A) \subset S(P)$ for some canonical factor A, then $P = AP'$ for some $P' \geqslant e$.*

(VII) *For any $P \geqslant e$, there is a unique canonical factor A such that $P = AP'$ for some $P' \geqslant e$ and $S(P) = S(A)$.*

Proof. To prove (I), observe that the defining relations (7) and (8) preserve the set of distinct subscripts which occur in a positive word, so if a_{qp} is in the starting set (resp. finishing set) of δ_π then $q = t_j$ and $p = t_i$ for some j, i with $m \geqslant j > i \geqslant 1$. Since it is proved in part (II) of Lemma 2.3 that every $a_{t_j t_i}$ occurs in both the starting set and the finishing set, the assertion follows.

To prove (II) one need only notice that δ_{π_r} commutes with δ_{π_s} when the cycles π_r, π_s are parallel.

Clearly (III) is a consequence of (I) and (II).

As for (IV), by Theorem 3.4 a canonical word is uniquely determined by a set of parallel descending cycles. If two distinct descending cycles π, μ are parallel, then δ_π, δ_μ have distinct starting sets, so if A and B are canonical factors, with $S(A) = S(B)$, the only possibility is that $A = B$.

To prove (V), we induct on the braid index n. The claim is clear for $n = 2$. Let $P \in B_n^+$ have starting set $S(P)$. If all generators of the form a_{nt} for $n-1 \geqslant t \geqslant 1$ are deleted from $S(P)$ we obtain a set $S'(P)$ which, by the induction hypothesis, is the starting set of a braid $A' = \delta_{\pi_1} \cdots \delta_{\pi_k}$, where $\pi_1, ..., \pi_k$ are parallel, descending cycles in Σ_{n-1}. It is now enough to check the following properties of $S(P)$:

(i) If $a_{ns}, a_{tr} \in S(P)$, with $t > s$, then $a_{ts} \in S(P)$;

(ii) If $a_{ns} \in S(P)$ and if s happens to be in one of the descending cycles $\pi_i = (t_m, ..., t_1)$ associated to $S'(P)$, then $a_{nt_j} \in S(P)$ for every j with $m \geqslant j \geqslant 1$;

(iii) If $a_{ns} \in S(P)$, where s is not in any of the descending cycles $\pi'_1, ..., \pi'_r$ associated to $S'(P)$, then there is no $a_{tr} \in S(P)$ such that $t > s > r$.

To establish (i), note that since $a_{ns}, a_{tr} \in S(P)$, we have $P = a_{ns} X = a_{tr} Y$ with $n > t > s > r$. But then the assertion follows from Theorem 2.4, part (IV), case (ii).

To establish (ii), set $\mu = (n, t_m, ..., t_1)$. Then y is a descending cycle for A, so by (I) and (II) of this lemma we conclude that $a_{nt_j} \in S(P)$ for $m \geqslant j \geqslant 1$.

Property (iii) can be verified by observing that if $a_{ns} \in S(P)$, then $\mu = (n, ..., s...)$ (where μ could be (n, s)) must be a descending cycle belonging to a canonical factor δ_μ for A. But if so, and if $a_{tr} \in S(P)$ with $r < s < t$, then by (i) a_{ts} is also in $S(P)$, so that in fact $\mu = (n, ..., t, ..., s, ..., r, ...)$. But then the cycle survives after deleting n, contradicting the hypothesis that the subscript s does not appear in any descending cycle associated to $S'(P)$. Thus we have proved (V).

To prove (VI), induct on the word length of P. The assertion is clear if $|P| = 1$. Assume $|P| > 1$. We may assume that a_{nt} is in $S(A)$ for some $1 \leqslant t < n$, otherwise we apply the index-shifting automorphism τ. We make this assumption to reduce the number of the cases that we have to consider.

By Theorem 3.4, we may write $A = \delta_{\pi_1} \delta_{\pi_2} \cdots \delta_{\pi_k}$ for some parallel, descending cycles $\pi_1, \pi_2, ..., \pi_k$ in Σ_n and we may assume $\pi_1 = (n, t_1, ..., t_j)$. Let $A = a_{nt_1} B$ and $P = a_{nt_1} Q$. We are done by induction if we can show that $S(B) \subset S(Q)$. Let a_{sr} be any member of $S(B)$. We have three possible cases after considering the properties of the words A and B:

(i) $s = t_1$ and $r = t_i$ for some $2 \leqslant i \leqslant j$;

(ii) $s < t_1$;

(iii) $t_1 < r < s < n$.

When (i) is the case, then $A = a_{nt_1} a_{t_1 t_i} C = a_{nt_i} a_{t_1 t_i} C$ for some canonical factor C and so $a_{nt_i} \in S(P)$. Since both a_{nt_1} and a_{nt_i} are in $S(P)$, Theorem 2.4(II)(ii) implies that $a_{t_1 t_i} \in S(Q)$. For the other two cases we can show, in a similar way, that a_{sr} is in $S(Q)$, using Theorem 2.4.

Assertion (VII) is an immediate consequence of (V) and (VI). ∎

Theorem 3.4 has given us an excellent description of the canonical factors. What remains is to translate it into a solution to the word problem. For that purpose we need to consider *products* $A_1 A_2 \cdots A_k$, where each A_i is a canonical factor. The argument we shall use is very similar to that in [7], even though our δ and our canonical factors are very different from their Δ and their permutation braids. See [10], where a similar argument is used.

A decomposition $Q = AP$, where A is a canonical factor and $P \geqslant e$, is said to be *left-weighted* if $|A|$ is maximal for all such decompositions. Notice that AP is *not* left-weighted of there exists $p \in S(P)$ such that AP is a canonical factor, for if so then $|A|$ is not maximal. We call A the *maximal head* of Q when $Q = AP$ is left-weighted. The symbol $A \ulcorner P$ means that AP is left-weighted. The following corollary gives an easy way to check whether a given decomposition is left-weighted.

Corollary 3.8. *Let A, P be positive words, with A representing a canonical factor. Then $A \ulcorner P$ if and only if for each $b \in S(P)$ there exists $a \leqslant A$ such that (a, b) is an obstructing pair.*

Proof. By the definition of left-weightedness, $A \ulcorner P$ if and only if, for each $b \in S(P)$, Ab is not a canonical factor. By Corollary 3.6 Ab is not a canonical factor if and only if Ab contains an obstructing pair (a, q). We cannot have both $a \leqslant A$ and $q \leqslant A$ because by hypothesis A is a canonical factor so that by Corollary 3.6 no word which represents it contains an obstructing pair. Thus $q = b$. ∎

Define the *right complementary set* $R(A)$ and the *left complementary set* $L(A)$ of a canonical factor A as follows:

$$R(A) = \{a \mid Aa \leqslant \delta\}$$

$$R(A) = \{a \mid aA \leqslant \delta\},$$

where a is a generator. Define the *right complement* of a canonical factor A to be the word A^* such that $AA^* = \delta$. Since $W\delta = \delta\tau(W)$, we have $L(\tau(A)) = \tau(L(A))$ and $R(\tau(A)) = \tau(R(A))$.

Note that $(A^*)^* = \tau(A)$, $R(A) = S(A^*) = F(A^*)$, $L(A^*) = F(A) = S(A)$. Also $R(A^*)$ $S(\tau(A)) = F(\tau(A))$, because $\tau^{-1}(A^*)A = \delta = A^*\tau(A)$ and $L(A) = F(\tau^{-1}(A^*)) = S(\tau^{-1}(A^*))$.

The next proposition shows us equivalent ways to recognize when a decomposition of a positive word is left-weighted.

PROPOSITION 3.9. *For any $Q \geqslant e$, let $Q = AP$ be a decomposition, where A is a canonical factor and $P \geqslant e$. Then the following are equivalent*:

(I) $A \ulcorner P$.

(II) $R(A) \cap S(P) = \varnothing$.

(III) $S(Q) = S(A)$.

(IV) *If $WQ \geqslant \delta$ for some $W \geqslant e$, then $WA \geqslant \delta$.*

(V) *For any $V \geqslant e$, $S(VQ) = S(VA)$.*

(VI) *If $Q = A_1 P_1$ is another decomposition with A_1 a canonical factor and $P_1 \geqslant e$, then $A = A_1 A'$ for some canonical factor A' (where A' could be e).*

Proof. See [7] and [10]. ∎

We can now give the promised normal form, wich solves the word problem for our new presentation for B_n:

THEOREM 3.10. *Any n-braid $\mathscr{W}$ has a unique representative W in left-canonical form*:

$$W = \delta^u A_l A_2 \cdots A_k,$$

where each adjacent pair $A_i A_{i+1}$ is left weighted and each A_i is a canonical factor. In this representation $\inf(\mathscr{W}) = u$ *and* $\sup(\mathscr{W}) = u + k$.

Proof. For any W representing $\mathscr{W}$ we first write $W = \delta^v P$ for some positive word P and a possibly negative integer v. For any $P \geqslant e$, we then iterate the left-weighted decomposition $P = A_1 P_1$, $P_1 = A_2 P_2$, ... to obtain $W = \delta^u A_1 A_2 \cdots A_k$, where $e < A_i < \delta$ and $R(A_i) \cap S(A_{i+1}) = \varnothing$. This decomposition is unique, by Corollary 3.7, because $S(A_i A_{i+1} \cdots A_k) = S(A_i)$ for $1 \leqslant i \leqslant k$. ∎

The decomposition of Theorem 3.10 will be called the *left-canonical form* of $\mathscr{W}$. For future use, we note one of its symmetries:

PROPOSITION 3.11. (I) *Let A, B be canonical factors. Then $A \ulcorner B$ if and only if $B^* \ulcorner \tau(A^*)$.*

(II) *The left-canonical forms of $\mathscr{W}$ and $\mathscr{W}^{-1}$ are related by:*

$$W = \delta^u A_l A_2 \cdots A_k, \qquad W^{-1} = \delta^{-(u+k)} \tau^{-(u+k)}(A_k^*) \cdots \tau^{-(u+1)}(A_1^*).$$

Proof. It is easy to show that the following identities hold for $A \in [0, 1]$.

$$S(\tau(A)) = \tau(S(A)),\ R(\tau(A)) = \tau(R(A)),$$
$$S(A^*) = R(A),\ R(A^*) = S(\tau(A)).$$

Then (I) is clear because

$$R(B^*) \cap S(\tau(A^*)) = S(\tau(B)) \cap R(\tau(A)) = \tau(R(A) \cap S(B))$$

As for (II), it is easy to see that the equation for W^{-1} holds. And it is the canonical form by (I). ∎

We end this section with two technical lemmas and a corollary which will play a role in the implementation of Theorem 3.10 as an algorithm. They relate to the steps to be followed in the passage from an arbitrary representative of a braid of the form $\delta^u A_1 A_2 \cdots A_r$, where each A_i is in $[0,1]$, to one in which every adjacent pair $A_i A_{i+1}$ satisfies the conditions for left-weightedness. The question we address is this: suppose that $A_1 A_2 \cdots A_i$ is left-weighted, that A_{i+1} is a new canonical factor, and that $A_i A_{i+1}$ is not left-weighted. Change to left-weighted form $A'_i A'_{i+1}$. Now $A_{i-1} A'_1$ may not be left-weighted. We change it to left-weighted form $A'_{i-1} A''_i$. The question which we address is whether it is possible that after both changes $A''_i A'_{i+1}$ is not left-weighted? The next two lemmas will be used to show that the answer is "no".

LEMMA 3.12. *Let AB, BC be canonical factors. Then $A \ulcorner C$ if and only if $(AB) \ulcorner C$.*

Proof. By Corollary 3.8, $(AB) \ulcorner C$ iff for each $c \in S(C)$ there exists $a \leqslant (AB)$ such that (a, c) is an obstructing pair. Since BC is a canonical factor, we know from Corollary 3.6 that we cannot have $a \leqslant B$. Therefore the only possibility is that $a \leqslant A$. ∎

LEMMA 3.13. *Suppose that A, B, C, D, B', C' are canonical factors and that $ABCD = AC'B'D$. Suppose also that BC, CD, AC', $C'B'$ are canonical factors, and that $A \ulcorner B$ and $B \ulcorner D$. Then $B' \ulcorner D$.*

Proof. By Proposition 3.11 it suffices to show that $D^* \lceil \tau((B')^*)$.

$$\begin{aligned} S(D^*) &\subseteq S(D^*\tau((B')^*)) \\ &\subseteq S(D^*\tau((B')^*)\, \tau^2((AC')^*)) \\ &= S(((AC')\, B'D)^{-1}\, \delta^3) \\ &= S((AB(CD))^{-1}\, \delta^3) \\ &= S((CD)^*\, \tau(B^*)\, \tau^2(A^*)) \\ &= S((CD)^*\, \tau(B^*)) \\ &= S(D^*\tau((BC)^*)) \\ &= S(D^*) \end{aligned}$$

Here the fourth equality which follows the first two inclusions is a consequence of the fact that $A \lceil B$. The sixth equality follows from $B \lceil D$, which (by Lemma 3.12) implies that $(BC) \lceil D$. But then, every inclusion must be an equality, so that $S(D^*) = S(D^*\tau(B'^*))$. But then, by Proposition 3.11, it follows that $B' \lceil D$. ∎

We now apply the two lemmas to prove what we will need about left-weightedness.

COROLLARY 3.14. *Suppose that A_{i-1}, A_i, A_{i+1} are canonical factors, with $A_{i-1} \lceil A_i$. Let A'_i, A'_{i+1} be canonical factors with $A_iA_{i+1} = A'_iA'_{i+1}$ and $A'_i \lceil A'_{i+1}$. Let A'_{i-1}, A''_i be canonical factors with $A_{i-1}A'_i = A'_{i-1}A''_i$ and $A'_{i-1} \lceil A''_i$. Then $A''_i \lceil A'_{i+1}$.*

Proof. The conversion of $(A_i)(A_{i+1})$ to left-weighted form $A'_i \lceil A'_{i+1}$ implies the existence of $U \geqslant e$ with

$$(A_i)(A_{i+1}) = (A_i)(UA'_{i+1}) = (A_iU)(A'_{i+1}) = (A'_i)(A'_{i+1}).$$

The subsequent conversion of $(A_{i-1})(A'_i)$ to left-weighted form $A'_{i-1} \lceil A''_i$ implies the existence of $V \geqslant e$ with

$$(A_{i-1})(A'_i) = (A_{i-1})(VA''_i) = (A_{i-1}V)(A''_i) = (A'_{i-1})(A''_i).$$

Set $A_{i-1} = A$, $A_i = B$, $U = C$, $A'_{i+1} = D$, $V = B'$, $A''_i = C'$ and apply Lemma 3.13 to conclude that $A''_i \lceil A'_{i+1}$. ∎

4. ALGORITHM FOR THE WORD PROBLEM, AND ITS COMPLEXITY

In this section we describe our algorithm for putting an arbitrary $\mathscr{W} \in B_n$ into left-canonical form and analyze the complexity of each step in the algorithm. The *complexity* of a computation is said to be $\mathcal{O}(f(n))$ if the number of steps taken by a Turing machine (TM) to do the computation is at most $kf(n)$ for some positive real number k. Our calculations will be based upon the use of a random access memory machine (RAM), which is in general faster than a TM model (see Chapter 1 of [2]). An RAM machine has two models: in the first (which we use) a single input (which we interpret to be the braid index) takes one memory unit of time. Unless the integer n is so large that it cannot be described by a single computer word, this "uniform cost criterion" applies. We assume that to be the case, i.e. that the braid index n can be stored by one memory unit of the machine.

We recall that each canonical factor decomposes into a product of parallel descending cycles $A = \delta_{\pi_1} \cdots \delta_{\pi_k}$, and that A is uniquely determined by the permutation $\pi_1 \cdots \pi_k$. So we identify a canonical factor with the permutation of its image under the projection $B_n \to \Sigma_n$. We denote each cycle π_i by its ordered sequence of subscripts. For example, we write $(5, 4, 3, 1)$ for $a_{54} a_{43} a_{31}$.

We use two different ways to denote a permutation π which is the image of a canonical factor: the first is by the n-tuple $(\pi(1), ..., \pi(n))$ and the second is by its decomposition as a product of parallel, descending cycles $\pi_1 \cdots \pi_k$. The two notations can be transformed to one another in linear time. The advantage of the notation $\pi = (\pi(1), ..., \pi(n))$ is that the group operations of multiplication and inversion can be perfomed in linear time.

If $A, B \in [0, 1]$, the *meet* of A and B, denoted $A \wedge B$, is defined to be the maximal canonical factor C such that $C \leqslant A$ and $C \leqslant B$. Our definition is analogous to that in [8, page 185]. Note that C can be characterized by the property that $S(C) = S(A) \cap S(B)$.

LEMMA 4.1. *If $A, B \in [0, 1]$, then $A \wedge B$ can be computed in linear time as a function of n.*

Proof. Let $A = \pi_1 \cdots \pi_k$ and $B = \tau_1 \cdots \tau_\ell$, where the ordering of the factors is arbitrary, but once we have made the choice we shall regard it as fixed. Let $\coprod$ denote disjoint union. Then $A \wedge B = \prod_{i,j} \pi_i \wedge \tau_j$ since

$$S(A \wedge B) = S(A) \cap S(B) = \left(\coprod_i S(\pi_i) \right) \cap \left(\coprod_j S(\tau_j) \right)$$

$$= \coprod_{i,j} (S(\pi_i) \cap S(\tau_j)) = \coprod_{i,j} (S(\pi_i \wedge \tau_j)).$$

For two descending cycles $\pi_i = (t_1, ..., t_p)$ and $\tau_j = (s_1, ..., s_q)$, we have:

$$S(\pi_i) \cap S(\tau_j) = \{a_{ts} \mid t > s \text{ and } t, s \in \{t_1, ..., t_p\} \cap \{s_1, ..., s_q\}\}$$

Thus

$$\pi_i \wedge \tau_j = \begin{cases} (u_1, ..., u_r) & \text{if } \{t_1, ..., t_p\} \cap \{s_1, ..., s_q\} = \{u_1, ..., u_r\} \quad \text{where } r \geqslant 2 \\ e & \text{if } |\{t_1, ..., t_p\} \cap \{s_1, ..., s_q\} \leqslant 1 \end{cases}$$

If we treat a decreasing cycle as a subset of $\{1, ..., n\}$ and a canonical factor as a disjoint union of the corresponding subsets, we may write $A \wedge B$ as

$$A \wedge B = \coprod_{i,j} (\pi_i \cap \tau_j).$$

We will find this disjoint union $A \wedge B$ of subsets of $\{1, ..., n\}$ in linear time by the following four steps:

1. Make a list of triples $\{(i, j, m)$ such that $m = 1, ..., n$ appears in π_i and $\tau_j\}$. We do this by scanning $A = \pi_i \cdots \pi_k$ first and writing $(i, \_, m)$ if π_i contains m and then scanning $B = \tau_1 \cdots \tau_\ell$ and filling in the middle entry of the triple $(i, \_, m)$ with j if τ_j contains m. We throw away all triples with a missing entry. The list contains at most n triples. For example, if $A = (5, 4, 1)(3, 2)$ and $B = (4, 2, 1)$, our list contains three triples, $(1, 1, 4)$, $(1, 1, 1)$, $(2, 1, 2)$. This operation is clearly in $\mathcal{O}(n)$.

2. Sort the list of triples lexicographically. In the above example, $(2, 1, 2)$, $(1, 1, 4)$, $(1, 1, 1)$ are the entries in the sorted list. There is an algorithm to do this in time $\mathcal{O}(n)$. See [2, Theorem 3.1].

3. Partition the sorted list by collecting triples with the same first two entries and then throw away any collection with less than one element. In the above example, $\{(2, 1, 2)\}$, $\{(1, 1, 4), (1, 1, 1)\}$ forms the partitioned list and we need to throw away the collection $\{(2, 1, 2)\}$. This can be done by scanning the sorted list once. Its complexity is $\mathcal{O}(n)$.

4. From each collection, write down the third entry to form a descending cycle. Note that the third entries are already in descending order. In the above example, $(4, 1)$ becomes $A \wedge B$. This step again takes $\mathcal{O}(n)$.

Since the above steps are all in $\mathcal{O}(n)$, we are done. ∎

Remark 4.2. We remark that the key step in both our computation and that in [8] is in the computation of $A \wedge B$, where A and B are permutation braids in [8] and canonical factors in our work. Our computation is described in Lemma 4.1. We now examine theirs. The set R_σ which is used in [8] is defined on pages 184–5 of [8] and characterized inductively on page 185. The fact that R_σ is defined inductively means that one cannot use a standard merge-sort algorithm. To get around this, the merge-sort approach is modified, as explained on lines 2–5 of page 206: one sorts $1, 2, ..., n$ by the rule $i < j$ if $C(i) < C(j)$, where $C(i)$ is the image of i under the permutation $C = A \wedge B$. This ordering between two integers cannot be done in constant time. The running time is $\mathcal{O}(n \log n)$, where $\log n$ is the depth of recursion and n is the time needed to assign integers as above and to merge sets at each depth.

Lemma 4.3. *Let A, B be canonical factors, i, e. $e \leqslant A$, $B \leqslant \delta$. There is an algorithm of complexity $\mathcal{O}(n)$ that converts AB into the left weighted decompostion, i.e. $A \ulcorner B$.*

Proof. Let A^* be the right complement of A, i.e. $A^* \in [0, 1]$ and $AA^* = \delta$. Let $C = A^* \wedge B$ and $B = CB'$ for some $B' \in [0, 1]$. Then $(AC) \ulcorner B'$, for if there is $a_{st} \leqslant B'$ such that $ACa_{st} \in [0, 1]$, then $a_{st} \leqslant B' \leqslant B$ and $a_{st} \leqslant (AC)^* \leqslant A^*$, which is impossible by the definition of meet. Thus the algorithm to obtain the left weighted decomposition consists in the following four steps:

(I) Compute the right complement A^* of A.

(II) Compute $C = A^* \wedge B$.

(III) Compute B' such that $B = CB'$.

(IV) Compute AC.

Step (II) is in $\mathcal{O}(n)$ by the Lemma 4.1. Steps (I), (III) and (IV) are in $\mathcal{O}(n)$ since they involve inversions and multiplications of permutations like $A^* = A^{-1}\delta$ and $B' = C^{-1}B$. ∎

Now the algorithm for the left canonical decomposition of arbitrary words is given by the following four processes.

The algorithm. We are given an element $\mathcal{W} \in B_n$ and a word W in the band generators which represents it.

1. If W is not a positive word, then the first step is to eliminate each generator which has a negative exponent, replacing it with $\delta^{-1}A$ for some positive word $A \in [0, 1]$. The replacement formulas for the negative letters in W is:

$$a_{ts}^{-1} = \delta^{-1}(n, n-1, ..., t+1, t, s-1, s-2, ..., 2, 1)(t-1, t-2, ..., s+1, s)$$

The complexity of this substitution process is at most $\mathcal{O}(n\,|W|)$. Notice that $|P|$ can be as long as $\mathcal{O}(n\,|W|)$, because each time we eliminate a negative letter we replace it by a canonical factor of length $n-2$.

2. Use the formulas:

$$A_i \delta^k = \delta^k \tau^k(A_i) \qquad \text{and} \qquad \delta^{-1} \delta^k = \delta^{k-1}$$

to move the δ^{-1}'s to the extreme left, to achieve a representative of $\mathscr{W}$ of the form

$$W = \delta^u A_1 A_2 \cdots A_k, \qquad A_i \in [0, 1] \tag{12}$$

and $|A_i| = 1$ or $n-2$, according as A_i came from a positive or a negative letter in W. Since we can do this process by scanning the word just once, the complexity of this rewriting process depends on the length of $A_1 A_2 \cdots A_k$ and so it is at most $\mathcal{O}(n\,|W|)$.

3. Now we need to change the above decomposition (12) to left canonical form. In the process we will find that u is maximized, k is minimized and $A_i \ulcorner A_{i+1}$ for every i with $1 \leqslant i \leqslant k$. This can be achieved by repeated uses of the subroutine that is described in the proof of Lemma 4.3.

In order to make the part $A_1 A_2 \cdots A_k$ left-weighted, we may work either forward or backward. Assume inductively that $A_1 A_2 \cdots A_i$ is already in its left canonical form. Apply the subroutine on $A_i A_{i+1}$ to achieve $A_i \ulcorner A_{i+1}$ and then to $A_{i-1} A_i$ to achieve $A_{i-1} \ulcorner A_i$. Corollary 3.14 guarantees that we still have $A_i \ulcorner A_{i+1}$, i.e. we do not need to go back to maintain the left-weightedness. In this manner we apply the subroutine at most i-times to make $A_1 A_2 \cdots A_i A_{i+1}$ left-weighted. Thus we need at most $k(k+1)/2$ applications of the subroutine to complete the left canonical form of $A_1 A_2 \cdots A_k$ and the complexity is $\mathcal{O}(|W|^2\, n)$ since k is proportional to $|W|$.

We may also work backward to obtain the same left canonical form by assuming inductively that $A_i A_{i+1} \cdots A_k$ is already in its canonical form and trying to make $A_{i-1} A_i A_{i+1} \cdots A_k$ left-weighted.

4. Some of canonical factors at the beginning of $A_1 A_2 \cdots A_k$ can be δ and some of canonical factors at the end of $A_1 A_2 \cdots A_k$ can be e. These should be absorbed in the power of δ or deleted. Note that a canonical factor A is δ if and only if $|A| = n-1$ and A is e if and only if $|A| = 0$. Thus we can decide whether A is δ of e in $\mathcal{O}(n)$ and so the complexity of this process is at most $\mathcal{O}(kn) = \mathcal{O}(|W|\, n)$.

THEOREM 4.4. *There is an algorithmic solution to the word problem that is $\mathcal{O}(|W|^2\, n)$ where $|W|$ is the length of the longer word among two words in B_n that are being compared.*

Proof. When we put two given words into their canonical forms, each step has complexity at most $\mathcal{O}(|W|^2\, n)$. ∎

5. THE CONJUGACY PROBLEM

Let $W=\delta^u A_1 A_2 \cdots A_k$, be the left-canonical form of $W \in B_n$. The result of a *cycling* (resp. *decycling*) of $W=\delta^u A_l A_2 \cdots A_k$, denoted by $\mathbf{c}(W)$ (resp. $d(W)$), is the braid $\delta^u A_2 \cdots A_k \tau^{-u}(A_1)$ (resp. $\delta^u \tau^u(A_k)\, A_1 \cdots A_{k-1}$). Iterated cyclings are defined recursively by $\mathbf{c}^i(W)=\mathbf{c}(\mathbf{c}^{i-1}(W)$, and similarly for iterated decyclings. It is easy to see that the cycling (resp. decycling) does not decrease (resp. increase) the inf (resp. sup).

With essentially no new work, we are able to show that the solution to the conjugacy problem of [9] and [7] can be adapted to our new presentation of B_n. This approach was taken in [10] for $n=4$. But there are no new difficulties encountered when one goes to arbitrary n. The following two theorems are the keys to the solution to the conjugacy problem.

THEOREM 5.1 ([7], [10]). *Suppose that W is conjugate to V.*

(I) *If* $\inf(V)>\inf(W)$, *then repeated cyclings will produce* $c^l(W)$ *with* $\inf(\mathbf{c}^l(W))>\inf(W)$.

(II) *If* $\sup(V)<\sup(W)$, *then repeated decyclings will produce* $\mathbf{d}^l(W)$ *with* $\sup(\mathbf{d}^l(W))<\sup(W)$.

(III) *In every conjugacy class, the maximum value of* $\inf(W)$ *and the minimnum value of* $\sup(W)$ *can be achieved simultaneously.*

THEOREM 5.2 ([7], [10]). *Suppose that two n-braids $V, W \in [u, v]$ are in the same conjugacy class. Then there is a sequence of n-braids $V=V_0$, $V_1, ..., V_k=W$, all in $[u, v]$, such that each V_{i+1} is the conjugate of V_i by some element of* $[0, 1]$.

AN ALGORITHM FOR THE SOLUTION TO THE CONJUGACY PROBLEM. We can now describe our solution to the conjugacy problem. Suppose that two words V, W represent conjugate elements $\mathscr{V}, \mathscr{W}$ of B_n. Recall the definitions of $\inf(V)$ and $\sup(V)$ which were given after Proposition 3.2. By Theorem 5.1, $\inf(V)\leqslant\sup(W)$ and $\inf(W)\leqslant\sup(V)$. Let $u=\min\{\inf(V), \inf(W)\}$ and let $v=\max\{\sup(V), \sup(W)\}$. Then $V, W\in[u, v]$. The canonical lengths $\sup(V)-\inf(V)$ and $\sup(W)-\inf(W)$ are proportional to the word lengths $|V|$ and $|W|$, respectively. Thus $v-u$ is at most $\mathcal{O}(|V|+|W|)$. The cardinality $|[u, v]|$ is given by $|[0, 1]|^{v-u}$. Since $|[0, 1]|\leqslant 4^n$, it follows that $|[u, v]|$ is at most $\mathcal{O}(\exp(n(|V|+|W|)))$. By Theorem 5.2, there is a sequence $V=V_0, V_1, ..., V_k=W$ of words in $[u, v]$ such that each element is conjugate to the next one by an element of $[0, 1]$. The length k of this sequence can be $|[u, v]|$ (in the worst case) and so we must have $U^{-1}VU=W$ for some positive word U of canonical length $\leqslant|[u, v]|$. Since there are $|[0, 1]|^{|[u, v]|}$ many positive words of canonical length $\leqslant |[u, v]|$, the number of all possible U is at most $\mathcal{O}(\exp(\exp(n(|V| + |W|)))$. This certainly gives a finite algorithm for the conjugacy problem.

A more practical algorithm can be given as follows: Given a n-braid W, the collection of conjugates of W that has both the maximal infimum and the minimal supremum is called the *super summit set* of W after [9], [7]. Clearly the super summit set of a word is an invariant of its conjugacy class. If we iterate the cycling operati on a word W, then the fact that the number of positive words of fixed length is finite insures that we eventually obtain an integer K such that $\mathbf{c}^N(W) = \mathbf{c}^{N+K}(W)$. In view of Theorem 5.1 we conclude that $\inf(\mathbf{c}^N(W))$ is the maximum value of infimum among all conjugates of W, Similarly, by interated decycling on $\mathbf{c}^N(W)$, we have $\mathbf{d}^M\mathbf{c}^N(W) = \mathbf{d}^{M+L}\mathbf{c}^{N+K}(W)$ and so we conclude that $\sup(\mathbf{d}^M\mathbf{c}^N(W))$ is the minimum value of supremum among all all conjugates of W. Therefore $\mathbf{d}^M\mathbf{c}^N(W)$ belongs the super summit set of W.

In order to decide whether two words V, W in B_n are conjugate, we proceed as follows:

1. Iterate cycling and decycling on V and W until we have V' and W' in the super summit sets, respectively. If V, W have distinct maximal inf or minimal sup, we conclude that they are not conjugate.

2. If V, W have the same maximal inf and minimal sup, the entire super summit set of V must be generated by using Theorem 5.2 and the finiteness of the super summit set.

3. If any one element in the super summit set of W is also in the super summit set of V, then W and V are conjugate. Otherwise, they are not.

It is hard to give complexity estimates for steps 1 and 2 above. We have many conjectures and lots of data which gives evidence of structure, but we have not been able to solve the problem.

EXAMPLE. The conjugacy classes of the following two 4-braids X, Y have the same "numerical class invariants", i.e. the same inf, sup and cardinality of the super summit set. Also, the super summit set splits into orbits under cycling and decycling, and the numbers and lengths of these orbits coincide. But the braids are not conjugate because their super summit sets are disjoint:

$$X = a_{43}^{-2} a_{32} a_{43}^{-1} a_{32} a_{21}^{3} a_{32}^{-1} a_{21} a_{32}^{-2},$$

$$Y = a_{43}^{2} a_{32}^{-1} a_{21}^{3} a_{32} a_{43}^{-1} a_{21}^{-1} a_{32}^{-2}.$$

ACKNOWLEDGMENTS

We thank Marta Rampichini for her careful reading of earlier versions of this manuscript, and her thoughtful questions. We thank Hessam Hamidi-Tehrani for pointing out to us the need to clarify our calculations of computational complexity.

REFERENCES

1. S. I. Adjan, Defining relations and algorithmic problems for groups and semigroups, *Proc. Steklov Inst. Math.* **85** (1966).
2. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, The design and analysis of computer algorithms, Addison–Wesley, Reading, MA (1974).
3. E. Artin, Theorie der Zopfe, *Hamburg Abh.* **4** (1925), 47–72.
4. J. S. Birman and W. Menasco, Studying links via closed braids III: Classifying links which are closed 3-braids, *Pacific J. Math.* **161** (1993), 25–113.
5. A. H. Clifford and G. B. Preston, *in* "Algebraic Theory of Semi-groups," Vol. I, Am. Math. Soc., Survey 7, Providence, Am. Math. Soc., 1961.
6. P. Dehornoy, A fast method for computing braids, *Adv. Math.* **125** (1997), 200–235.
7. E. A. Elrifai and H. R. Morton, Algorithms for positive braids, *Quart. J. Math. Oxford* **45**, No. 2 (1994), 479–497.
8. D. B. A. Epstein (with Cannon, Holt, Levey, Patterson and Thurston), "Word Processing in Groups," Jones and Bartlett, Boston, MA, (1992).
9. F. A. Garside, The braid group and other groups, *Quart. J. Math. Oxford* **20**, No. 78 (1969), 235–254.
10. E. S. Kang, K. H. Ko, and S. J. Lee, Band-generator presentation for the 4-braid group, *Topology Appl.* **78** (1997), 39–60.
11. A. Errera, Une problème d'énumération, *Mém. Acad. Roy. Belgique Coll.* **8**(2), v. 11 (1931).
12. K. H. Ko *et al.*, *in* "Positive Presentation of the Braid Groups and the Embedding Problem," manuscript in preparation.
13. M. S. Paterson and A. A. Rasborov, The set of minimal braids is co-NP-complete, *J. Algorithms.* **12** (1991), 393–408.
14. J. H. Remmers, On the geometry of semigroup presentations, *Adv. Math.* **36** (1980), 283–296.
15. L. Rudolph, Braided surfaces and Seifert ribbons for closed braids, *Comm. Math. Helv.* **58** (1983), 1–37.
16. V. Sergiescu, Graphes planaires et présentations des groupes de tresses, *Math. Z.* **214** (1993), 477–490.
17. P. J. Xu, The genus of closed 3-braids, *J. Knot Theory Ramifications* **1**, No. 3 (1992), 303–326.