SORTING AND GENERATING REDUCED WORDS

OLCAY COSKUNE AND MUGE TASIKIN

ABSTRACT. We introduce a partial order on the set of all reduced words of a given permutation \( \omega \), called directed-braid poset of \( \omega \). This poset enables us to produce two algorithms: One is a sorting algorithm applied on any reduced word of \( \omega \) and aims to obtain the natural word (lexicographically largest reduced word); the other one is a generation algorithm applied on the natural word and aims to obtain the set of all reduced words of \( \omega \).

1. Introduction

The symmetric group \( S_n \) on \([n] = \{1, 2, \ldots, n\}\) is generated by adjacent transpositions \( \{s_i : i = 1, 2, \ldots, n-1\}\), where \( s_i \) stands for the transposition \((i, i+1)\). Given any permutation \( \omega \in S_n \), and any expression \( s_{i_1}s_{i_2}\ldots s_{i_l} \) representing \( \omega \), we call the sequence \( i_1i_2\ldots i_l \) a word for \( \omega \). Such an expression with minimal \( l \) is called a reduced word for \( \omega \) and \( l \) is called as the length of \( \omega \), denoted by \( l(\ast) \). The index \( l \) is determined by \( \omega \) and, by a result of Tits, any two reduced words for \( \omega \) are related by the braid relations given as follows.

(1) (Short braid relation) \( s_is_j = s_js_i \) for any \( i, j \) with \(|i - j| \geq 2\).

(2) (Long braid relation) \( s_is_{i+1}s_i = s_{i+1}s_is_{i+1} \) for any \( i \).

For any permutation \( \omega \), we denote by \( \mathcal{R}(\omega) \) the set of all reduced words of \( \omega \). By the above result, any reduced word can be obtained from any other by applying a series of braid relations. However this procedure of applying a sequence of braid relations is non-trivial since one needs to go back and forth using the braid relations. For that reason, we introduce the directed-braid poset \( (\mathcal{R}(\omega), <_{\text{d-braid}}) \) where the underlying partial order is obtained by putting a direction on the applications of the short and long braid relations.

The partial order \(<_{\text{d-braid}}\) is weaker than the lexicographic order on \( \mathcal{R}(\omega) \). Moreover the unique maximal word in \( (\mathcal{R}(\omega), \leq_{\text{lex}}) \) remains as the unique maximal element in the directed-braid poset. This word is the one that we obtain when we apply the selection sort algorithm to the one line notation of the permutation. The algorithm aims to

Date: 10, 2012.

Both of the authors are supported by Boğaziçi - Bap - 6029.
convert the permutation to the identity, by moving, at each step, the smallest number in $\omega$ which is not in its identity position.

The maximal element can also be described by tower diagrams introduced in [2], where it is called the natural word of the permutation. We will use the notation of [2]. By [2 Proposition 5.1], the natural word can be characterized as a list of increasing sequences of consecutive integers with decreasing first terms.

With the direction provided by directed-braid poset, we then introduce a sorting algorithm on any reduced word of $\omega$, which at the end obtains the natural word of $\omega$ as chain in directed-braid poset. Thus we obtain a canonical route between any two reduced words of $\omega$ which goes through the natural word. This algorithm can be seen as the combination of selection sort and insertion sort algorithms.

Having the canonical route provided by the sorting algorithm, we then introduce a generation algorithm for all reduced words of given permutation, starting from its natural word. At the first step, we produce basic words of the given permutation. This step is basically the application of the long braid relation to the natural word. After determining the set of all basic words, it only remains to apply the short braid relation, which is done by the restricted shuffle on basic words. This step is a variation of the well-known shuffle operation on words.

There are combinatorial objects to determine the set of all reduced words of a given permutation, such as balanced labeling of the Rothe diagram [3] and RC graphs [11] of permutations, the plastification map [6] and the tower tableaux [2]. Generating reduced words from these objects is not efficient. Two examples of more efficient algorithms are the one that counts saturated chains in the weak order on permutations and the one which uses heaps of reduced words [8]. See [4] for more information on these algorithms. We leave the comparison of our algorithm, in terms of computational complexity, with these algorithms as a problem to the interested reader.

The paper is organized as follows. In Section 1, we introduce the directed-braid poset together with its basic properties. The sorting and the generation algorithms are introduced in 3 and 4 respectively. To illustrate our algorithm, we produce the reduced words of the longest permutation in $S_4$.

The following notations will be used throughout the paper. Let $b = \beta_1 \beta_2 \ldots \beta_l$ be a word with $l \geq 1$. We call $b$ a tower word if for any index $1 < i \leq l$, we have $\beta_i = \beta_{i-1} + 1$; that is, if $b$ is an increasing sequence of consecutive integers. It is clear, from the construction in the above discussion, that if $b$ is a tower word, then it is the natural
word of a tower diagram with a unique tower of positive length. For example, 4567 is a tower word.

It is also clear that any word \( \beta = \beta_1 \beta_2 \ldots \beta_n \) can be written uniquely as the concatenation of tower words, say as \( \beta = b_1 b_2 \ldots b_s \) for some \( s \), where \( b_i \) is called \( i \)-th tower word in \( \beta \). We call \( b_1 b_2 \ldots b_s \) as the tower decomposition of \( \beta \). For example, if \( \beta = 78954534562 \), then the tower decomposition is given by

\[
\beta = \underbrace{789}_{b_1} \underbrace{5}_{b_2} \underbrace{45}_{b_3} \underbrace{3456}_{b_4} \underbrace{2}_{b_5}.
\]

2. The directed-braid poset of a permutation

In this section, we introduce the directed-braid poset of a permutation \( \omega \) as a main tool for constructing a sorting algorithm on the set \( R(\omega) \) of all reduced words of \( \omega \). This poset will be the combination of two relations defined as follows.

**Definition 2.1.** Let \( \alpha, \beta \in R(\omega) \) be two reduced words and \( l = l(\omega) \).

We write \( \alpha <_1 \beta \) if there exist \( 1 \leq i < l \) such that

\[
\begin{align*}
\alpha &= \alpha_1 \ldots \alpha_i \alpha_{i+1} \ldots \alpha_l, \\
\beta &= \alpha_1 \ldots \alpha_{i+1} \alpha_i \ldots \alpha_l \text{ and} \\
\alpha_{i+1} &\geq \alpha_i + 2.
\end{align*}
\]

Clearly this covering relation refers to the short braid relation and also puts a restriction on the direction that we can apply it. A key property of this relation is the following lemma.

**Lemma 2.2.** The relation \( <_1 \) implies the lexicographic order, that is, if \( \alpha <_1 \beta \), then we also have \( \alpha \leq_{\text{lex}} \beta \).

The proof follows easily from the definition. It is trivial that the converse is not true. For example \( 121 \leq_{\text{lex}} 212 \) but \( 121 \neq 121 \). As a corollary to this lemma, we get that the reflexive and transitive closure of the relation generated by \( <_1 \) is a partial order. Indeed, the only non-trivial property is that of anti-symmetry which is guaranteed by the above lemma.

In order to define the second relation, we will first introduce several notations: Let \( \mathbf{a} \) be a tower word. Then we denote by

i. \( \tilde{\mathbf{a}} \) the tower word obtained by increasing the numbers in the tower word \( \mathbf{a} \) by 1

ii. \( \text{in}(\mathbf{a}) \) the initial letter of \( \mathbf{a} \).

iii. \( \text{fin}(\mathbf{a}) \) the final letter of \( \mathbf{a} \).

The definition of the second relation which depends the tower decomposition of reduced words is as follows:
Definition 2.3. Let \( \alpha, \beta \in \mathcal{R}(\omega) \) and let \( \alpha = a_1 \cdots a_i a_{i+1} \cdots a_s \) be the tower decomposition of \( \alpha \). We say \( \alpha <_2 \beta \) if there exist \( 1 \leq i < s \) such that

\[
\text{in}(a_i) \leq \text{in}(a_{i+1}) < \text{fin}(a_{i+1}) < \text{fin}(a_i)
\]

and \( \beta = a_1 \cdots a_{i-1} \tilde{b}_1 a_i b_2 a_{i+2} \cdots a_s \)

where the tower words \( b_1 \) and (possibly empty) \( b_2 \), satisfy that

\[
b_1 b_2 = a_{i+1}.
\]

Remark. First observe that the representation of \( \beta \) in the above definition is not necessarily the tower decomposition of \( \beta \), since \( a_{i-1} \tilde{b}_1 \) might already be a tower word.

Secondly the condition that \( \text{in}(a_i) \leq \text{in}(a_{i+1}) < \text{fin}(a_{i+1}) \) necessarily implies \( \text{fin}(a_{i+1}) < \text{fin}(a_i) \), since the other case yields a contradiction to the fact that \( \alpha \) is a reduced word.

As an example, consider the following three reduced words

\[
\alpha = 23\ 5678\ 67, \ \beta = 23\ 78\ 5678 \text{ and } \gamma = 23\ 7\ 5678\ 7
\]

of the permutation \( w = 134268975 \) given by their tower decompositions. Then we have \( \alpha <_2 \beta \) and \( \alpha <_2 \gamma \) and also \( \gamma <_2 \beta \).

It is clear that via \( <_2 \), we are moving tower words from right to left and letter by letter. During these moves, the condition is basically given by a series of braid relations which always include a long one. As in the previous case, we have the following lemma.

Lemma 2.4. The relation \( <_2 \) implies the lexicographic order, that is, if \( \alpha <_2 \beta \), then we also have \( \alpha \leq_{\text{lex}} \beta \).

Again, the proof follows from the definition and clearly the converse is not true. For example \( 124 \leq_{\text{lex}} 142 \) but \( 124 \not<_2 142 \). Moreover, the reflexive and the transitive closure of the relation generated by \( <_2 \) is a partial order.

Now we define a partial order on the set of all reduced words.

Definition 2.5. For \( \alpha, \beta \in \mathcal{R}(\omega) \), we write

\[
\alpha \leq_{d\text{-braid}} \beta
\]

if either \( \alpha = \beta \) or there is a sequence \( \gamma^0, \gamma^1, \ldots, \gamma^m \) of reduced words in \( \mathcal{R}(\omega) \) such that \( \gamma^0 = \alpha, \gamma^m = \beta \) and for any \( i, 0 \leq i \leq m - 1 \), we have either \( \gamma^i <_1 \gamma^{i+1} \) or \( \gamma^i <_2 \gamma^{i+1} \).

We have the following result.

Proposition 2.6. The relation \( \leq_{d\text{-braid}} \) on \( \mathcal{R}(\omega) \) is a partial order.
Proof. Reflexivity and transitivity of the relation follows directly from the definition. We only prove that the relation is anti-symmetric. By Lemma 2.2 and Lemma 2.3 if \( \alpha \preceq_i \beta \), for \( i = 1, 2 \) then \( \alpha \preceq_{\text{lex}} \beta \). Thus \( \leq_{\text{d-braid}} \) is anti-symmetric as \( \leq_{\text{lex}} \) is.

The main result of this section is that the poset \( (\mathcal{R}(\omega), \leq_{\text{d-braid}}) \) has a unique maximal element. As we have described above, this result is the starting point of an algorithm to generate reduced words from the natural one.

Proposition 2.7. The natural word \( \eta_\omega \) of \( \omega \) is the unique maximal element in \( (\mathcal{R}(\omega), \leq_{\text{d-braid}}) \).

Proof. Let \( \alpha \) be a maximal element in \( (\mathcal{R}(\omega), \leq_{\text{d-braid}}) \) and let
\[
\alpha = a_1 \ldots a_ia_{i+1} \ldots a_k
\]
be the tower decomposition of \( \alpha \). Now since \( \alpha \) is maximal, there is no word \( \beta \) such that \( \alpha <_1 \beta \). But this is only possible if for any \( i, 1 \leq i < k \), we have
\[
\text{fin}(a_i) > \text{in}(a_{i+1}).
\]
Indeed, otherwise, \( \text{fin}(a_i) < \text{in}(a_{i+1}) \) yields \( \text{in}(a_{i+1}) - \text{fin}(a_i) > 2 \) since \( a_{i+1} \) and \( a_i \) are different tower words. Hence one can interchange \( \text{fin}(a_i) \) and \( \text{in}(a_{i+1}) \) to obtain a greater word in \( (\mathcal{R}(\omega), \leq_{\text{d-braid}}) \).

Now suppose that for some \( 1 \leq i < k \) we have \( \text{in}(a_i) \leq \text{in}(a_{i+1}) \). Then by the above discussion we have
\[
\text{in}(a_i) \leq \text{in}(a_{i+1}) < \text{fin}(a_i)
\]
but this forces that \( \text{fin}(a_{i+1}) \leq \text{fin}(a_i) \), since otherwise \( \alpha \) is not a reduced word. Hence by definition of \( <_2 \), one can obtained a reduced word \( \beta \) such that \( \alpha <_2 \beta \), but this contrads to the maximality of \( \alpha \). Therefore for each \( 1 \leq i < k \), we have
\[
\text{in}(a_i) > \text{in}(a_{i+1}).
\]

Thus we have proved that any maximal element \( \alpha \) in \( (\mathcal{R}(\omega), \leq_{\text{d-braid}}) \) has the property that in its tower decomposition, the sequence of initial letters of tower words is decreasing. But by [2, Proposition 5.1], there is a unique word with this property, namely the natural word.

Although there is a unique maximal in the braid poset, there might be many minimal elements. An example of a braid poset with two minimal elements is the poset of the word 432123, where the minimal are the words 124321 and 143213. The full poset in this case is given as follows.
Remark 1. The above result tells us that it is possible to generate all reduced words starting from the natural word. The problem is to determine a canonical path from the maximal element to the chosen one. The advantage we have here is that via the directed-braid poset, we insist a direction on the braid relations: The short braid relation is the equality $s_i s_j = s_j s_i$ if $|j - i| \geq 2$, but in the braid poset, to go from the natural word to an arbitrary word, we can only interchange $j$ and $i$ if $j$ is smaller than $i$. Similar comment is true for the long braid relation. Therefore, a maximal element in this poset is a word on which the directed-braid relations cannot be applied. In this sense, the natural word is the only braid-free word. Hence one would expect to have a canonical route from the natural word to any other reduced word and vice versa. For the rest of the paper, we explain such canonical routes.

Remark 2. In [5], the poset of commutation classes is introduced. Given a permutation $\pi$, two reduced words $\alpha$ and $\beta$ are in the same commutation class if they differ from each other by the short braid relation, hence are comparable with the partial order generated by $<_1$. This poset is used in [5] in relation with factorization of Schubert cells.

3. Sorting algorithm: From a reduced word to the natural word of a permutation

By the Proposition 2.7, the natural word of a permutation $\omega$ is the unique maximum among all reduced words of $\omega$ in the directed-braid poset. In this section, we introduce two algorithms, by which one obtains the natural word $\eta_{\omega}$ from any reduced word $\alpha$ of $\omega$ as a chain in this poset.
3.1. **A selection sort algorithm on reduced words.** Let $\alpha$ be a word reduced word for a permutation $\omega$, given with its tower decomposition $\alpha = a_1 \ldots a_i a_{i+1} \ldots a_k$. We call $\alpha$ a *natural basic word* for $\omega$ if for each $1 \leq i < k$ we have

$$\text{fin}(a_i) > \text{in}(a_{i+1}).$$

It is easy to see the natural word of $\omega$ is the unique natural basic word which satisfies that the sequence of initial letters of its tower words is strictly decreasing.

The selection sort algorithm on any reduced word $\alpha$ of $\omega$ aims to obtain a natural basic word, say $\beta$ of $\omega$, as a chain $\alpha = \beta_0 \leq \beta_1 \leq \ldots \beta_k = \beta$ of reduced words in $(\mathcal{R}(\omega), \leq_{d\text{-braid}})$.

We describe the algorithm inductively. Suppose that $\beta_j$ is constructed. Then we obtain $\beta_{j+1}$ as follows: Write $\beta_j = b_1 \ldots b_r$ in its the tower decomposition and $a$ let be its smallest letter. Then $a$ is necessarily an initial letter of some tower words in $\beta_j$ and let $b_s$ be the right most tower word starting with $a$, for some $1 \leq s \leq r$.

i. If $s < r$ and if $\text{fin}(b_s) < \text{in}(b_{s+1})$ (necessarily $\text{in}(b_{s+1}) - \text{fin}(b_s) \geq 2$) then we let

$$\beta_{j+1} = b_1 \ldots b_s b_{s+1} \ldots b_r.$$

ii. If $s = r$ or if $\text{fin}(b_s) > \text{in}(b_{s+1})$ then we continue by applying the same algorithm on the right most tower word starting with $a$, which is on the left of $b_s$.

iii. Finally, if none of the tower words starting with $a$ in $\beta_j$ can move to the right subject to the above rule, then we continue to apply the same algorithm with the smallest initial letter bigger than $a$ in $\beta_j$.

Observe that the above algorithm may not produce a new word, $\beta_{j+1}$, and this happens if and only if $\beta_j$ is a natural basic word i.e., for any $1 \leq s < r$

$$\text{fin}(b_s) > \text{in}(b_{s+1})$$

In this case we let $\beta = \beta_j$. On the other hand if $\beta_{j+1}$ is produced as a result then we have $\beta_j <_1 \beta_{j+1}$.

**Example 3.1.** We consider the reduced word $\alpha = 134521321654321$ to produce a natural basic word. We use brackets to indicate the tower...
words which are to be moved according to the above algorithm.

\[ \alpha = \alpha^0 = 1 34567 2 1 3 2 [1] 6 45 4 3 2 1 \]
\[ <_1 \beta^1 = 1 34567 2 1 3 2 6 [1] 45 4 3 2 1 \]
\[ <_1 \beta^2 = 1 34567 2 1 3 2 6 45 [1] 4 3 2 1 \]
\[ <_1 \beta^3 = 1 34567 2 1 3 2 6 45 4 [1] 3 2 1 \]
\[ <_1 \beta^4 = 1 34567 2 [1] 3 2 6 45 4 3 12 1 \]
\[ <_1 \beta^5 = 1 34567 23 [12] 6 45 4 3 12 1 \]
\[ <_1 \beta^6 = 1 34567 23 6 [12] 45 4 3 12 1 \]
\[ <_1 \beta^7 = 1 34567 23 6 45 [12] 4 3 12 1 \]
\[ <_1 \beta^8 = [1] 34567 23 6 45 4 123 12 1 \]
\[ <_1 \beta^9 = 34567 [123] 6 45 4 123 12 1 \]
\[ <_1 \beta^{10} = 34567 6 12345 4 123 12 1 = \beta. \]

3.2. An insertion sort algorithm on natural basic words. Our next aim is to construct an algorithm which transforms a natural basic word of a permutation to its natural word. The algorithm is based on the relation \(<_2\).

Let \( \beta = b_1 \ldots b_k \) be a natural basic word for a permutation, given with its tower decomposition. Therefore

\[ \text{fin}(b_i) > \text{in}(b_{i+1}) \text{ for all } 1 \leq i < k. \]

Note that, in this case we have

either \( \text{in}(b_i) > \text{in}(b_{i+1}) \)

or \( \text{in}(b_i) \leq \text{in}(b_{i+1}) < \text{fin}(b_i) \).

Observe that the second case also forces that

\[ \text{in}(b_i) \leq \text{in}(b_{i+1}) \leq \text{fin}(b_{i+1}) < \text{fin}(b_i) \]

since otherwise \( \beta \) can not be a reduced word. Moreover in this case the two words

\[ \beta = b_1 \ldots b_i b_{i+1} \ldots b_k \text{ and } \beta' = b_1 \ldots \widetilde{b}_{i+1} b_i \ldots b_k \]

can be obtained from one another through a sequence of short and long braid relations and \( \beta <_2 \beta' \).

Now we are ready to explain the algorithm which converts any natural basic word to the unique the natural word of the corresponding permutation. Let \( \beta \) be a natural basic word whose tower word decomposition is of the following form.

\[ \beta = \beta^0 = b_1 \ldots b_k. \]
If \( \text{in}(b_i) > \text{in}(b_{i+1}) \) for all \( 1 \leq i \leq k \) then \( \beta \) is the natural word and we do not proceed. Otherwise let \( j \) be the largest index such that \( \text{in}(b_i) \leq \text{in}(b_{i+1}) \). As it is discussed above, this forces that
\[
\text{in}(b_i) \leq \text{in}(b_{i+1}) \leq \text{fin}(b_{i+1}) < \text{fin}(b_i)
\]
and we consider the following word
\[
(1) \quad b_1 \ldots b_{i-1}\widetilde{b}_{i+1}b_i \ldots b_k
\]
which is clearly is braid equivalent to \( \beta^0 \). We have three cases to consider.

i. If \( \text{fin}(b_{i-1}) > \text{in}(\widetilde{b}_{i+1}) \) then \( b_1 \ldots b_{i-1}\widetilde{b}_{i+1}b_i \ldots b_k \) is the tower decomposition of a natural basic word and we let
\[
\beta^1 = b_1 \ldots b_{i-1}\widetilde{b}_{i+1}b_i \ldots b_k.
\]

ii. If \( \text{fin}(b_{i-1}) + 1 = \text{in}(\widetilde{b}_{i+1}) \) then we naturally concatenate \( b_{i-1} \) and \( \widetilde{b}_{i+1} \) to get the tower decomposition of this word. Since
\[
\text{fin}(b_{i-1}\widetilde{b}_{i+1}) > \text{in}(b_i),
\]
we let \( \beta^1 = b_1 \ldots b_{i-1}\widetilde{b}_{i+1}b_i \ldots b_k \) be the natural basic word in (1).

iii. If \( \text{fin}(b_{i-1}) + 1 < \text{in}(\widetilde{b}_{i+1}) \) then the word in (1) is not a natural basic word. But then one can move \( \widetilde{b}_{i+1} \) to the left of \( b_{i-1} \) by using short braid relations and continue in the similar manner, if necessary, until the resulting word is a natural basic word. In this case we let \( \beta^1 \) be this natural basic word.

The above algorithm yields that \( \beta^0 \prec_{\text{d-braid}} \beta^1 \). Now applying the above algorithm repeatedly we obtain a sequence of natural basic words
\[
\beta = \beta^0 < \beta^1 \ldots.
\]
This sequence terminates after finitely many steps at some word, say \( \beta^r \), in which the sequence of initial letters of each tower words is strictly decreasing, i.e \( \beta^r \) is the natural word of the corresponding permutation.

**Example 3.2.** Observe that \( \beta = 2345678 \ 234 \ 1234567 \ 56 \) is a natural basic word given by its tower decomposition. In the following we use brackets to indicate the tower words subject to move according to above algorithm.
\[ \beta = \beta^0 = 2345678 \ 234 \ 1234567 \ [56] \]
\[ <_{d\text{-braid}} \beta^1 = 2345678 \ [67] \ 234 \ 1234567 \]
\[ <_{d\text{-braid}} \beta^2 = 78 \ 2345678 \ [234] \ 1234567 \]
\[ <_{d\text{-braid}} \beta^3 = 78 \ 345 \ 2345678 \ 1234567 \]

For another example we consider \( \gamma = 3456 \ 5 \ 1234 \ 123 \ 12 \ 1 \) which is also a natural basic word given with its tower decomposition. Then

\[ \gamma = \gamma^0 = 3456 \ 5 \ 1234 \ 123 \ 12 \ [1] \]
\[ <_{d\text{-braid}} \gamma^1 = 3456 \ 5 \ 1234 \ 123 \ [2] \ 12 \]
\[ <_{d\text{-braid}} \gamma^2 = 3456 \ 5 \ 1234 \ [3] \ 123 \ 12 \]
\[ <_{d\text{-braid}} \gamma^3 = 3456 \ 5 \ 4 \ 1234 \ 123 \ [12] \]
\[ <_{d\text{-braid}} \gamma^4 = 3456 \ 5 \ 4 \ 1234 \ [23] \ 123 \]
\[ <_{d\text{-braid}} \gamma^5 = 3456 \ 5 \ 4 \ 34 \ 1234 \ [123] \]
\[ <_{d\text{-braid}} \gamma^6 = 3456 \ [5] \ 4 \ 34 \ 234 \ 1234 \]
\[ <_{d\text{-braid}} \gamma^7 = 6 \ 3456 \ [4] \ 34 \ 234 \ 1234 \]
\[ <_{d\text{-braid}} \gamma^8 = 6 \ 5 \ 3456 \ [34] \ 234 \ 1234 \]
\[ <_{d\text{-braid}} \gamma^9 = 6 \ 5 \ 45 \ 3456 \ 234 \ 1234 = \eta \]

4. Generation algorithm: from the natural word to a reduced word

The sorting algorithm introduced in the previous section shows that there is a canonical route from an arbitrary reduced word to the natural word. In this section, we will try to reverse this algorithm to get a generation theorem.

The generation algorithm of the reduced words of any permutation \( \omega \), as the sorting algorithm suggests, consists of two parts. In the first part, we only allow the tower words of the natural word of \( \omega \) to pass each other by the passage operation, whose definition arises from the insertion sort algorithm. We call each word obtained in this way a basic word, in fact some of the words are natural basic words. In the next step, by taking the selection sort algorithm into account, we apply restricted shuffle operation on each basic words to obtain all reduced words of \( \omega \).

4.1. Basic words of a tower diagram. We begin with preliminary definitions.
Let \( b \in \mathbb{Z}^+ \) and let \( \alpha = a_1, \ldots, a_r \) be a reduced word of a permutation given by its tower decomposition. If \( b_0 a_1 a_2 \ldots a_r \) is a reduced word then the track sequence of \( b \) through \( \alpha \) is the largest finite sequence of terms 
\[ b_0, b_1, \ldots, \]
such that \( b_0 = b \) and for \( i \geq 1 \)
\[ b_i := \begin{cases} 
(b_{i-1}) - 1 & \text{if } \text{in}(a_i) < b_{i-1} \leq \text{fin}(a_i) \\
b_{i-1} & \text{if either } b_{i-1} \leq \text{in}(a_i) - 2 \text{ or } b_{i-1} \geq \text{fin}(a_i) + 2, \\
\text{undefined} & \text{otherwise.} 
\end{cases} \]

It is clear that if \( b_i \) is not defined, for some \( 1 \leq i \leq r \), then \( b_{i+1} \) is not defined. Hence the maximum possible number of elements in this sequence is \( r + 1 \).

**Definition 4.1.** Let \( b \in \mathbb{Z}^+ \) and let \( \alpha = a_1, \ldots, a_r \) be a reduced word of a permutation given by its tower decomposition. If \( b_0 a_1 a_2 \ldots a_r \) is not a reduced word then we set passwords\((b, \alpha) = \emptyset\). Otherwise we set
\[
\text{passwords}(b, \alpha) := \{\alpha^0 = b_0 a_1 a_2 \ldots a_r\} \\
\cup \{\alpha^i = a_1 \ldots a_i b_i a_{i+1} \ldots a_r \mid 1 \leq i \leq s\}
\]
where \( b_0, b_1, \ldots, b_s \) is the track sequence of \( b \) through \( \alpha \).

**Example 4.2.** Let \( \alpha = 3456 \ 789 \ 12345 \), and \( b = b_0 = 6 \). Then passwords\((b, \alpha)\) consist of the following reduced words
\[
\alpha^0 = [6] \ 3456 \ 789 \ 12345 \\
\alpha^1 = 3456 [5] \ 789 \ 12345 \\
\alpha^2 = 3456 \ 789 \ [5] \ 12345 \\
\alpha^3 = 3456 \ 789 \ 12345 \ [4]
\]
where the numbers in the brackets, gives the track sequence of \( b = 6 \) through \( \alpha \). Namely, \( b_0 = 6, b_1 = 5, b_2 = 5, b_3 = 4 \). One can easily check the following examples.
\[
\text{passwords}(10, \alpha) = \{10 \ 3456 \ 789 \ 12345, \ 3456 \ 10 \ 789 \ 12345\} \\
\text{passwords}(2, \alpha) = \{23456 \ 789 \ 12345\} \\
\text{passwords}(1, \alpha) = \emptyset
\]

**Definition 4.3.** Let \( \alpha \) and \( \beta \) be two reduced words such that the concatenation \( \beta \alpha \) is also reduced. We define the set of all passage
words of $\beta$ through $\alpha$ as follows: Write $\beta = \beta_1 \beta_2 \ldots \beta_n$, then

$$\text{passwords}(\beta, \alpha) = \text{passwords}(\beta_1 \beta_2 \ldots \beta_n, \alpha)$$

$$:= \bigcup_{\tilde{\alpha} \in \text{passwords}(\beta_n, \alpha)} \text{passwords}(\beta_1 \beta_2 \ldots \beta_{n-1}, \tilde{\alpha}).$$

$$:= \bigcup_{\tilde{\alpha} \in \text{passwords}(\beta_2 \ldots \beta_n, \alpha)} \text{passwords}(\beta_1, \tilde{\alpha}).$$

Moreover we let that $\text{passwords}(\beta, \alpha) = \emptyset$ if $\beta \alpha$ is not a reduced word and that $\text{passwords}(\beta, \alpha) = \{\alpha\}$ if $\beta$ is the empty word. Finally, for any two sets $A$ and $B$ of words, we define

$$[B, A] := \bigcup_{\alpha \in A, \beta \in B} \text{passwords}(\beta, \alpha).$$

**Example 4.4.** Consider $\alpha = 345678$, and $\beta = 96$. Then

$$\text{passwords}(6, 345678) = \{6\ 345678, 3456578, 3456785\}$$

$$\text{passwords}(9\ 6, 345678) = \{9\ 6\ 345678, 9\ 3456578, 9\ 3456785\}$$

$$\quad 6\ 9\ 345678, 6\ 3456978,$$

$$\quad 9\ 3456578, 34569\ 5\ 78, 34565\ 9\ 78,$$

$$\quad 9\ 3456785, 34569\ 78\ 5\}$$

We are now ready to define basic words of a permutation.

**Definition 4.5.** Let $\eta_w = \eta_1 \ldots \eta_k$ be the tower decomposition of the natural word of $\omega$ and let $N_i = \{\eta_i\}$ for each $1 \leq i \leq k$. Then the set of basic words for $\omega$ is the set given by

$$\text{Basic}(\omega) := [[\ldots [[N_1, N_2], N_3], \ldots, N_{k-1}], N_k].$$

The following result follows directly from the above definition. We leave the justification to the reader.

**Lemma 4.6.** Let $\omega$ be a permutation. Then any basic word in $\text{Basic}(\omega)$ is reduced and is braid related to the natural word of $\omega$.

4.2. **Restricted shuffle.** The second step of the generation algorithm is the restricted shuffle. This operation is a restriction of the well-known shuffle operation on words. Recall that, given two words $\alpha$ and $\beta$, a shuffle of $\beta$ over $\alpha$ is obtained by placing the letters of $\beta$ arbitrarily between the letters of $\alpha$ without changing the order of letters of $\beta$. The set of all shuffles of $\beta$ over $\alpha$, denoted by $\text{Sh}(\alpha, \beta)$ can be obtained by first concatenating $\beta$ to the right of $\alpha$ to obtain a new word $\alpha \beta$ and
then moving the letters of $\beta$ to the left, without changing their orders, until the word $\beta \alpha$ is obtained.

On the other hand, the restricted shuffle employs the same idea by adding a restriction: A letter $\beta_i$ of $\beta$ can pass a letter $\alpha_j$ of $\alpha$ if and only if $\alpha_j \not\in \{\beta_i - 1, \beta_i, \beta_i + 1\}$. With this definition, it is clear that we are referring to the short braid relation. More precise definition is as follows.

**Definition 4.7.** Suppose that the letters in $\alpha = \alpha_1 \ldots \alpha_n$ and $\beta = \beta_1 \beta_2 \cdots \beta_m$ are colored by red and blue, respectively. A restricted shuffle of $\alpha$ with $\beta$ is a word $w = w_1 \ldots w_{n+m}$ of $n$ red and $m$ blue letters satisfying the following conditions.

1. The restrictions of $w$ on the red and blue letters give the words $\alpha$ and $\beta$ respectively.
2. If, for some $1 \leq k \leq m$ and $1 \leq i \leq n$, the letter $\beta_k$ lies to the left of $\alpha_i$ in $w$ then none of $\{\beta_k - 1, \beta_k, \beta_k + 1\}$ lies in $\alpha_i \ldots \alpha_n$.

We denote by $\text{ResSh}(\alpha, \beta)$ the set of all restricted shuffles of $\alpha$ with $\beta$.

**Example 4.8.** Let $\alpha = 13425$ and $\beta = 37$ and color the word $\beta$ by boldface. Then

$$\text{ResSh}(\alpha, \beta) = \{13425\overline{37}, 1342\overline{3}5\overline{7}, 134\overline{2}3\overline{7}5, 134\overline{3}2\overline{5}7, 134\overline{3}5\overline{2}7\}.$$ 

For $\alpha = 13465$ and $\beta = 37$ we have

$$\text{ResSh}(\alpha, \beta) = \{13465\overline{37}, 1346\overline{3}5\overline{7}, 1346\overline{3}57, 134\overline{3}6\overline{5}7, 134\overline{3}6\overline{7}5\}.$$ 

The following result follows easily from the definition of the restriction shuffle and the definitions of the braid relations. We leave the straightforward proof to the reader.

**Lemma 4.9.** Let $\alpha, \beta$ and the concatenation $\alpha \beta$ be reduced words of some permutations. Then

i. any restricted shuffle of $\alpha$ with $\beta$ is also reduced and

ii. the restricted shuffles of $\alpha$ with $\beta$ are pairwise distinct.

As in the case of passage words, we can generalize the above operation to a restricted shuffle of several words by induction. Let $u_1, u_2, \ldots, u_n$ be some words. Then we define

$$\text{ResSh}(u_1, u_2, \ldots, u_n) := \bigcup_{\alpha \in \text{ResSh}(u_1, u_2, \ldots, u_{n-1})} \text{ResSh}(\alpha, u_n).$$

4.3. **Generation Theorem.** We are now ready to state the generation theorem
Theorem 4.10 (Generation Theorem). Let $\omega$ be a permutation. Then

$$\text{Red}(\omega) = \bigcup_{\alpha \in \text{Basic}(\omega)} \text{ResSh}(a_1, a_2, \ldots, a_r)$$

where $\alpha \in \text{Basic}(\omega)$ is given by its tower decomposition $\alpha = a_1a_2 \ldots a_r$.

Proof. It is clear from Lemma 4.9 and Lemma 4.6 that the right hand side is contained in the left hand side. To prove the converse inclusion, it is sufficient to show that the sorting algorithm is inverse to the generation algorithm. But clearly, any step in the selection sort algorithm is a restricted shuffle. Moreover, these steps are consistent with the order of the parenthesis in the restricted shuffle of the generation algorithm. Similarly, any natural basic word is a basic word. Indeed the reverse of each step in the insertion sort algorithm is a passage word construction. \qed

5. Example: Longest words

As our first example, we produce all reduced expressions for the longest word in $S_4$. The general case, for an arbitrary $S_n$, can be treated in the same way. Recall that the longest word $\omega_0$ is the reverse of the identity permutation and its natural word is given by $\eta = 3 \ 23 \ 123$ in its tower decomposition. Then

$$\text{Basic}(\omega_0) = \{[3], \{23\}, \{123\}\}.$$  

Here $[3], \{23\} =$ passwords($3, 23$) = $\{3 \ 23, 23 \ 2\}$ and hence

$$\text{Basic}(\omega_0) = \text{passwords}(3 \ 23, 123) \cup \text{passwords}(23 \ 2, 123)$$

where

passwords($3 \ 23, 123$) = $\{3 \ 23 \ 123, 3 \ 2 \ 123 \ 2, 3 \ 123 \ 12, 123 \ 2 \ 12, 123 \ 12 \ 1\}$

passwords($23 \ 2, 123$) = $\{23 \ 2 \ 123, 23 \ 123 \ 1, 2 \ 123 \ 2 \ 1, 123 \ 12 \ 1\}$.

Note that the last elements of the above sets coincide and we omit one of them. To obtain the set of all reduced words, it remains to apply restricted shuffle to each of the basic words which are listed below.

$$\text{ResSh}(3, 23, 123) = \{3 \ 23 \ 123, 3 \ 2 \ 1 \ 3 \ 23\}$$

$$\text{ResSh}(3, 2, 123, 2) = \{3 \ 2 \ 123 \ 2\}$$

$$\text{ResSh}(3, 123, 12) = \{3 \ 123 \ 12, 1 \ 3 \ 23 \ 12, 3 \ 12 \ 1 \ 3 \ 2, 1 \ 3 \ 2 \ 1 \ 3 \ 2\}$$
ResSh(123, 2, 12) = \{123 2 12\}
ResSh(23, 2, 123) = \{23 2 123\}
ResSh(23, 12, 1) = \{23 12 23 12 2 3 2 1 3 2 3 2 1 3\}
ResSh(123, 2, 21) = \{2 123 21\}
ResSh(123, 12, 1) = \{123 12 1\}

Finally, the union of all these restricted shuffles gives us the full set of reduced words for the longest permutation of $S_4$. Note that there are 16 reduced words in the union and the number coincides with the one that Stanley’s formula \cite{Stanley} gives.

\textbf{References}

[1] N. Bergeron, S. Billey, \textit{RC-graphs and Schubert polynomials}, Experiment Math. \textbf{2}, (1993), 257-269.
[2] O. Coşkun, M. Taşkın, \textit{Tower Tableaux}, J. Combin. Theory Series A, \textbf{120} (2013), 843-871.
[3] S. Fomin, C. Greene, V. Reiner, M. Shimozono, \textit{Balanced labellings and Schubert polynomials}, European J. Combinatorics, \textbf{18} (1997), 379-389.
[4] A. Garsia, \textit{The saga of reduced factorizations of elements of the symmetric group}, Publications du LaCIM, Université du Québec à Montréal, Canada, Vol. 29, 2002.
[5] C.. Kassel, A. Lascoux, C. Reutenauer, \textit{Factorization in Schubert cells}, Adv. Math. \textbf{150} (2000), 1-35.
[6] V. Reiner, M. Shimozono, \textit{Plactification}, J. Alg. Comb. \textbf{4} (1995), 331-351.
[7] R. Stanley, \textit{On the number of reduced decompositions of elements of Coxeter groups}, European J. Combinatorics, \textbf{5} (1984), 359-372.
[8] J.R. Stembridge, \textit{Some combinatorial aspects of reduced words in finite Coxeter groups}, Trans. Amer. Math. Soc. \textbf{349} (1997), 1285-1332.

Boğaziçi Üniversitesi, Matematik Bölümü, Bebek, İstanbul, Turkey

\textit{E-mail address}: olicay.coskun@boun.edu.tr, muge.taskin@boun.edu.tr