University of North Carolina Asheville
Journal of Undergraduate Research
Asheville, North Carolina
Spring 2025

The Garside Normal Form of The Braid Groups

Noah Haley

Department of Mathematics
University of North Carolina Asheville
One University Heights
Asheville, North Carolina 28804 USA

Faculty Advisor: Dr. David Peifer

Abstract

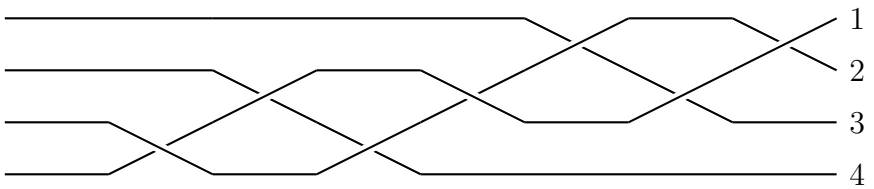
A mathematical braid is a collection of crossings between n strings which flow continuously from right to left with fixed starting and end points. These crossing must be done in a specific manner in order to maintain structure. Specifically, after crossing that continue horizontally to the left. These will form a group, B_n, under concatenation. As a braid need not have a particular length, or number of crossings, this group is infinite in size and quite difficult to analyze. By turning our attention to the work done by William Thurston in [4], we see that the subset of positive braids, B_n^+, the problem is reduced. Specifically, using the Garside braid to generate a unique "factorization" of braids in order to create a structuring of B_n.

1 Introduction

Imagine you have n parallel strings laid out on a table flowing right to left. You then intertwine those strings by crossing one string over the one directly below it, making sure that afterwards the strings once again flow right to left but in their reordered positions. Once you've intertwined them to a desired amount, by doing as many of these crossings as needed, you have created what is called braid on those n strings. An example of a braid on 4 strands is shown in Figure 1 below. As seen in Figure 1, the braid must flow in a continuous path from right to left, i.e. if you were to put it in $\mathbb{R}^2$ the end points must lie in the same y-positions as the starting points, possibly in a different order. Furthermore, we label the strings 1 to n starting from top to bottom.

This choice of direction of flow is an arbitrary decision, the strands could flow left to right. In fact, many people these days will construct the braids such that they flow vertically. However, once a direction is set, there must be consistency. The reason for left to right, is that it is convenient for the operation as it behaves similarly to function composition. From Figure 1, we can identify a convenient way to describe the crossings making up the braid. When a crossing is done by crossing two strands such that the top goes over the bottom, creating a positive slope in the crossing, we call it a positive crossing. Furthermore, we can denote that crossing by which

strand crossed over another. Thus, the first crossing in Figure 1 can be denoted x_1, as strand 1 crosses over strand 2. Now, there can also be a crossing where a strand crosses over the strand above, denoted as a negative crossing. It's important to note that negative crossings are the inverse of positive crossings, as they do exactly the opposite move. So mathematically we can represent them as x_i^{-1}. Lastly, we see in Figure 1, as we continue to follow the strings we can read off each crossing. Thus, giving the word representing the braid as seen below the braid.



$$x_1x_2x_1x_2^{-1}x_3x_2x_3^{-1}$$

Figure 1: Example Braid on 4 Strands

2 Mathematical Background

Let's quickly remind ourselves of some concepts from group theory before diving into the specifics of B_n.

Definition: *Group*
Let G be a set paired with a binary operation (represented in multiplicative notation). We say G is a group under the operation if the following properties are satisfied:

$0)$ ***Closure***. The operation is closed; that is, for all $a, b \in G$, $(ab) \in G$.

$1)$ ***Associativity***. The operation is associative; that is, $(ab)c = a(bc)$ for all $a, b, c \in G$.

$2)$ ***Identity***. There is an element of G, denoted as e, called the identity, such that $ae = ea = a$ for all $a \in G$.

$3)$ ***Inverses***. For all $a \in G$, there is a $b \in G$, called the inverse of a, such that $ab = ba = e$

In a later section we will discuss why B_n satisfies each one of the groups properties.

Definition: *Permutation, Symmetric Group*
Let A be a set. A permutation of A is a bijective mapping of A to A. The symmetric group, S_n, is the set of all permutations of $\{1, .., n\}$ under function composition.

Since a permutation is bijective, you get all elements of A using all elements of A, effectively just reordering the elements. S_n helps to investigate many objects that are studied in group theory, including B_n. In a later section we will discuss the relation of S_n to B_n.

The final concept to discuss is the idea of generators, relations, and group presentations. As this a abstract mathematical concept, we wish only to give a conceptual overview. See [9] and [5] for concrete mathematical definitions. Given a set $A = \{a, b, c, ...\}$ of distinct symbols and an inverse set $A^{-1} = \{a^{-1}, b^{-1}, c^{-1}...\}$. Define, $W(A)$ to be the set of all finite strings of the symbols from $A \cup A^{-1}$. We call these finite strings words. Furthermore, let's define equivalence classes on $W(A)$ where words of the form $x_ix_i^{-1} = e$, where e is the empty word. For example, take

$abb^{-1}baa^{-1}b^{-1} \in W(A)$ and simplify it by removing all parts which are equivalent to the empty word. So it would break down into, $aebeb^{-1} = abb^{-1} = ae = a$. Now, if we define $W(A)$ to have the operation of concatenation, we see that $W(A)$ form a group.

Furthermore, we can create an algebraic group to have an isomorphism with any group by using the correct generators and relations. Given some group of symbols G, some subset $A = \{a_1, a_2, ...a_n\}$, and a set of relations $W = \{w_1, w_2, ...w_n\}$. Here is an example of a relation, $a^2bb^{-1} = e$. Notice this allows you to further simplify a given word if that combination of letters is there. This relation would commonly be written like, $a^2b = b$, to keep things orderly. Now, we say that G is generated by A if when we apply the relations in W, to A we obtain G. Furthermore, we say that G has the presentation, $G \equiv \langle a_1, a_2, ..., a_n | w_1, w_2, ...w_n \rangle$. Now, we can say that any group is isomorphic to an algebraic group with a given presentation, if it exhibits the same properties of the algebraic group. An easy example of this is, $\mathbb{Z}_n \equiv \langle a | a^n \rangle$. Say $n = 3$, so $\mathbb{Z}_3 = \{0, 1, 2\}$. Notice that the group obtained from the presentation, $\{e, a, a^2\}$, behaves exactly like $\mathbb{Z}_3$ under concatenation. Thus, we say that $\mathbb{Z}_3 \equiv \langle a | a^3 \rangle$.

3 Group Presentation for B_n

Before algebraically defining B_n with a group presentation, let's confirm that B_n actually forms a group under concatenation. Firstly, is B_n closed? If we concatenate two braids we create a longer braid on n strands, making B_n closed. Second, is concatenation associative? If we take three braids, say x, y, z, and concatenate like so $(xy)z$, where we do the parentheses first, it is exactly the same as $x(yz)$. The order of the crossings will remain the same no matter which two we concatenate first. Therefore, B_n is associative. Third, is there an identity element? Yes, it's the braid represented by the empty word e, i.e. the braid with no crossings, as seen in Figure 2. Yes, if we concatenate any braid with e we technically created a longer braid in B_n by extending the strands. However, we have added no new crossings, thus it's as if we have done nothing at all. Furthermore, when we concatenate the braid words, we are adding the empty word, or adding nothing at all. Therefore, e is the identity in B_n.

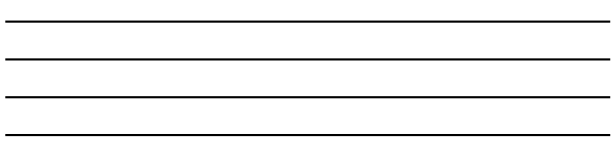


Figure 2: Identity in B_n

Lastly, are there inverse elements in B_n? Yes, given any braid in B_n just replace all positive crossings with negatives, and all negative crossings with positives, and do them in reverse order, i.e. undo each crossing sending it to e after concatenating. An example is given in Figure 3 below. Notice, that when we concatenate these two braids each strand ends up back in its original position, as we are "unwinding" the braid.

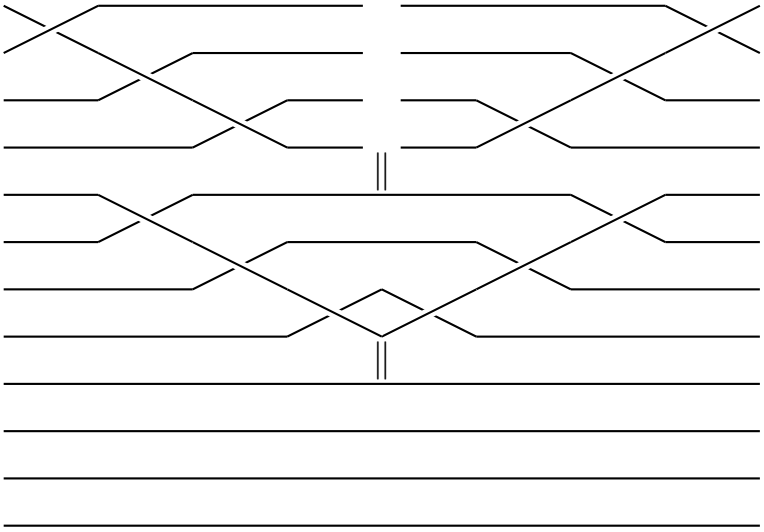


Figure 3: $(x_1x_2x_3)^{-1} = x_3^{-1}x_2^{-1}x_1^{-1}$

Now, in order to properly define B_n we will define the group presentation. First, notice that we can take the generating set to be the set of positive crossings, $\{x_1, x_2, ..., x_{n-1}\}$. Furthermore, we will use the set of negative crossings as inverses, $\{x_1^{-1}, x_2^{-1}, ..., x_{n-1}^{-1}\}$. From here, we will need to find the relations that will create all braids in B_n. Figure 4 below demonstrates the two relations with their algebraic representation.

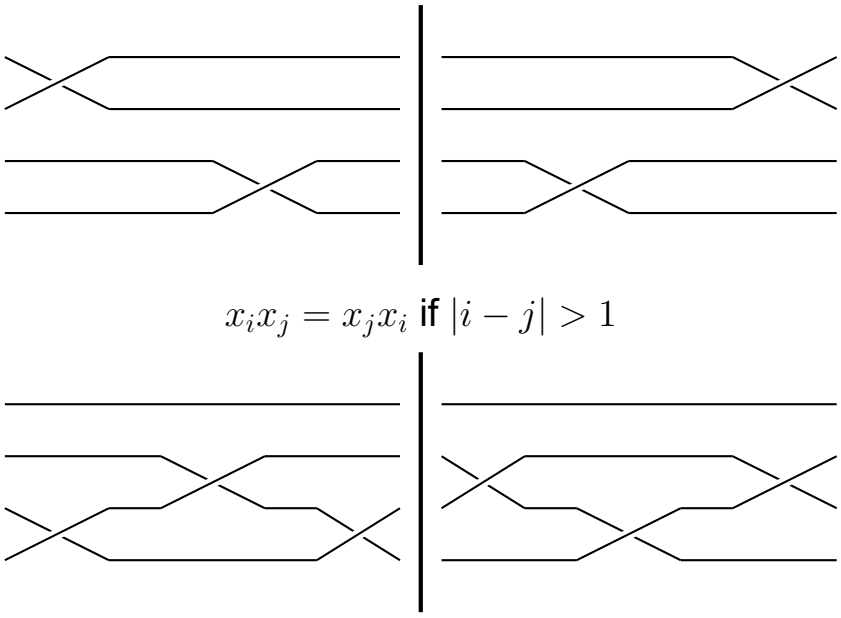


Figure 4: Braid Relations

If we were to read off the words representing the braids in the top portion of Figure 4, we get x_1x_3 for the right, and x_3x_1 for the left. Furthermore, if you follow the strands through each braid you'll notice that in both braids strand 1 crosses over strand 2, and strand 3 crosses over strand 4. This implies that the two braids are equal to each other as they have exactly the same crossings, and $x_1x_3 = x_3x_1$. Figure 4 also shows, two crossings will commute only when they are separated by a strand, i.e. the

difference between the strands must be greater than 1. Otherwise, each respective braid will behave differently. Furthermore, if we read off the words representing the braids in the bottom portion of Figure 4, we get $x_2x_3x_2$ for the right, and $x_3x_2x_3$ for the left. Once again, if we follow the strands through right braid, 1 crosses over nothing, 2 crosses over 3 then 4, 3 crosses under 2 then over 4, and 4 crosses under 2 then 3. For the left braid, 1 crosses nothing, 2 crosses over 4 then 3, 3 crosses over 4 then under 2, and 4 crosses under 3 then 2. Implying that each braid has the same exact crossings, even though reversed, and thus are equal, i.e. $x_2x_3x_2 = x_3x_2x_3$. From here, justifying that all relations in B_n are a consequence of this small set of relations is difficult. This was first proved by Emil Artin in [1], where Artin showed that the presentation is,

$$B_n \equiv \langle x_1, x_2, ...x_{n-1} | x_i x_{i+1} x_i = x_{i+1} x_i x_{i+1} \text{ , } x_i x_j = x_j x_i \text{ if } |i-j| > 1 \rangle$$

4 Positive Braids in B_n

After understanding the presentation of B_n it is clear that B_n is infinite in size. There is no limit as to how many combinations of crossing you can put into a braid. This makes it rather difficult to dig further into B_n. However, if we turn our attention to special subsets of B_n that behave nicely, we can narrow in our focus by seeing how these subsets interact with the entire group. One of the more important subsets for our study is the set of positive braids, B_n^+. A positive braid is an element of B_n consisting of only positive crossings. Figure 1 is an example of a positive braid. Quickly note, this subset does not form a subgroup, as it does not contain negative crossings, or inverse elements. In fact, this set forms a *monoid* under concatenation. For further information on monoids, see [2], and [11].

This subset has a number of useful properties, primarily in how it lies in B_n. That is, two positive words in B_n represent the same braid if and only if they represented the same braid in B_n^+. This was first proved by Garside in [6], but later shown by Thurston in [4]. What makes this useful is that the options for a braid which is represented by a given positive word has a finite number of choices in B_n^+, unlike B_n. This is because in B_n we can leverage inverse crossings to simplify words into a word representing a positive braid. Furthermore, B_n^+ contains what is called the Garside braid, $\mathcal{G}_n$. This is a half twist of the strands, as seen in Figure 5. This is called a half twist as each strand only makes it halfway through the braid, i.e. it doesn't make it back to it's "original" position. Notice, if you follow each strand, $1 \to 4$, $4 \to 1$, $3 \to 2$, and $3 \to 2$.

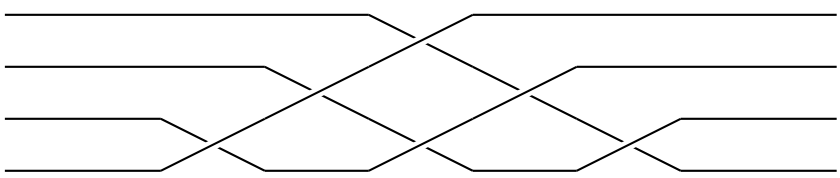


Figure 5: $\mathcal{G}_4 = (x_3x_2x_1)(x_3x_2)(x_3)$

This half twist has a number of important properties. First, given a positive crossing, x_i, there exists *positive braids*, L_i and R_i, where $\mathcal{G}_n = x_iR_i = L_ix_i$. We call L_i the left tail, and R_i the right tail of $\mathcal{G}_n$. Rearranging the equations yields, $x_i^{-1} = \mathcal{G}_n^{-1}R_i = L_i\mathcal{G}_n^{-1}$. Therefore, each negative crossing can be written as a product of $\mathcal{G}_n^{-1}R_i$, or $L_i\mathcal{G}_n^{-1}$. This means that every braid in B_n can be rewritten using only positive

generators by figuring out what power, denoted t, of $\mathcal{G}_n^{-1}$ is needed for each x_i^{-1} in the braid. In fact, you can take any braid, including positive braids, and break them down into these "factors" of $\mathcal{G}_n$. We call this factorization the Garside normal form for the braid groups. This is a similar idea to factorization in the integers. We wish to break down a "larger" braid into "factors" of $\mathcal{G}_n$ in order to understand the overall structure of B_n. As without a form of factorization, the braids are too complex to study, as there are infinitely many braid words which can represent any given braid. Below we will dive into an example of performing this factoring on a positive braid. The following example can also be found in [3], although our presentation will include more detailed comments and pictures. For those interested in further mathematical details on this factorization, see the following section.

Our goal is to find the Garside normal form of the following positive braid,

$$P = x_1x_3x_2^2x_1x_3^2x_2x_3x_2$$

First lets draw out this braid.

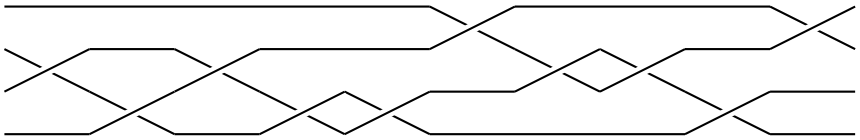


Figure 6: Example Positive Braid

Now, you want to algebraically break the braid up wherever you see a repeated crossing. This makes it easier to view any braid moves which can be done. Furthermore, in the following braid pictures, there will be separating lines at each place the braid is algebraically split.

$$P = (x_1x_3x_2)(x_2x_1x_3)(x_3x_2x_3)(x_2)$$

As we work from right to left in the braid word, notice that we can preform a braid move on $x_3x_2x_3$.

$$P = (x_1x_3x_2)(x_2x_1x_3)(x_2x_3x_2)(x_2)$$

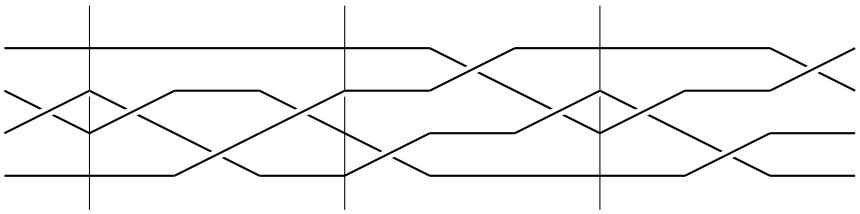


Figure 7: $x_3x_2x_3 = x_2x_3x_2$

Notice, that we can now push x_2x_3 from the third piece into the second, as these strands will no longer cross twice there.

$$P = (x_1x_3x_2)(x_2x_1x_3x_2x_3)(x_2)(x_2)$$

Back to the braids moves.

$$P = (x_1x_3x_2)(x_2x_1x_2x_3x_2)(x_2)(x_2)$$

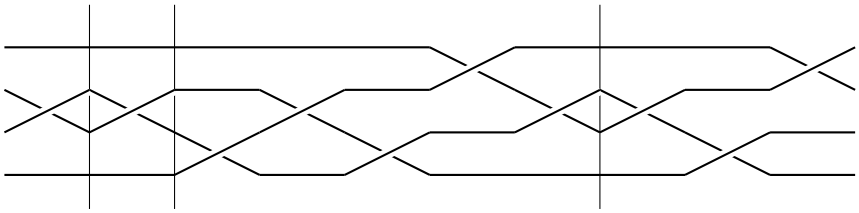


Figure 8: Moving x_2x_3

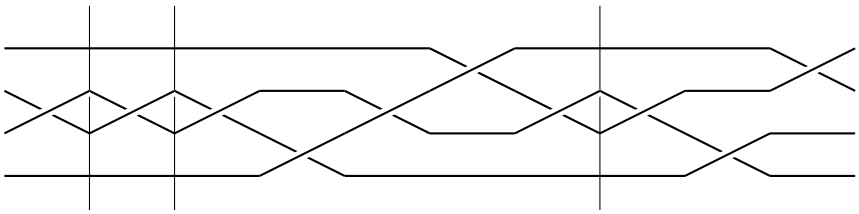


Figure 9: $x_3x_2x_3 = x_2x_3x_2$

One more time on that piece.

$$P = (x_1x_3x_2)(x_1x_2x_1x_3x_2)(x_2)(x_2)$$

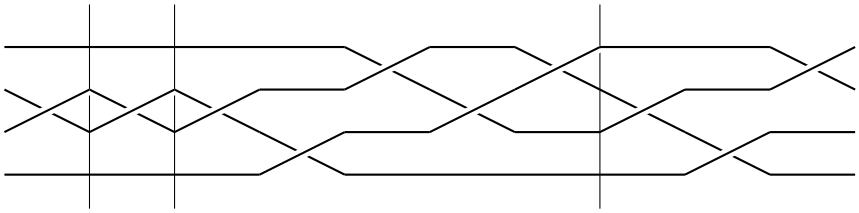


Figure 10: $x_1x_2x_1 = x_2x_1x_2$

Now we can push x_1 from the second into the first. We cannot push the x_2 as strands 3 and 2 would now cross twice.

$$P = (x_1x_3x_2x_1)(x_2x_1x_3x_2)(x_2)(x_2)$$

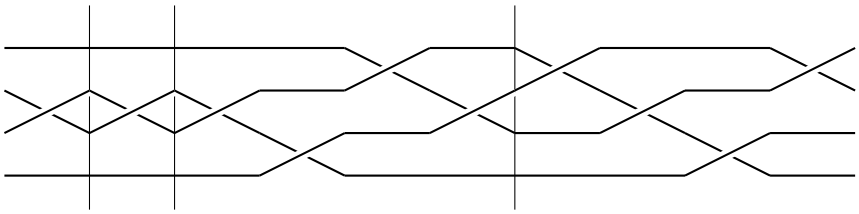


Figure 11: Moving x_1

Notice that in each piece, any two strands will cross at most once. Furthermore, the only braid moves which can be perform are $x_1x_3 = x_3x_1$ in both the first and second sections of the braid. However, either one of these moves would result in crossings getting entangled in each section. Therefore, we can conclude that this braid is now in Garside normal form.

5 The Mathematics Behind the Garside Normal Form

The following discussion elaborates on an example first examined by Thurston in [4], and later by Birman in [3]. Starting with the implied automorphism, $f : B_n \to B_n$ where $f(X) = \mathcal{G}_n^{-t} X \mathcal{G}_n^t$. We can rearrange the equation, $\mathcal{G}_n^t f(X) = X\mathcal{G}_n^t$, by multiplying by $\mathcal{G}_n^t$ on the left. This demonstrates that $\mathcal{G}_n^t$ can be factored to either side of a braid. Furthermore, this implies that given an arbitrary braid X, it can be represented by a word of the form, $\mathcal{G}_n^t Z$, where Z is a positive word. On the surface, we accomplished the goal, an arbitrary braid is factored! However, the power t is arbitrary. Therefore, the representation $\mathcal{G}_n^t Z$ is not unique. In order to correct this, we'll have discover if there exists a word, $\mathcal{G}_n^i Z$, such that i is maximal. Simply put, higher powers would begin to represent the wrong braids. Now, we still do not know what factoring into Z looks like, as Z is non unique. However, notice that if we take all words which represent Z to be, $Z_0, Z_1, ...$, we can choose Z_0 such that it's composition of crossings is minimal. Resulting in factorization, $\mathcal{G}_n^i Z_0$. So we are clearly interested in the composition of Z_0.

This can be rather difficult as there are a great number of words which can represent braids of this form. However, notice that Z_0 can contain arbitrary powers of sub words of $\mathcal{G}_n$. So we'll start by factoring, $\mathcal{G}_n = lr$, where we call l the left divisor, and r the right divisor. Notice that there are many words which can represent either l or r. So we'll put all left divisors into a set called P, and all right divisors into a set P'. As $\mathcal{G}_n$ is a half twist, the factors will have a $1-1$ correspondence with the permutations of the end points. So, there will be $n!$ elements, and in fact this set is usually referred to as the set of permutation braids. This connection comes with a very important property for factoring $\mathcal{G}_n$. Each element in P, and P' consists of braids such that any two strands can only cross once! This is a crucial property as it significantly narrows down how many words can represent our factoring. Below is a figure of all of the permutation braids in B_4. Where each step along the lattice, we are removing a crossing systematically such that the resulting braid will follow the above requirement. Quickly note, the figure below was constructed with the braids flowing slightly differently. They are numbered the same, but they flow left to right.

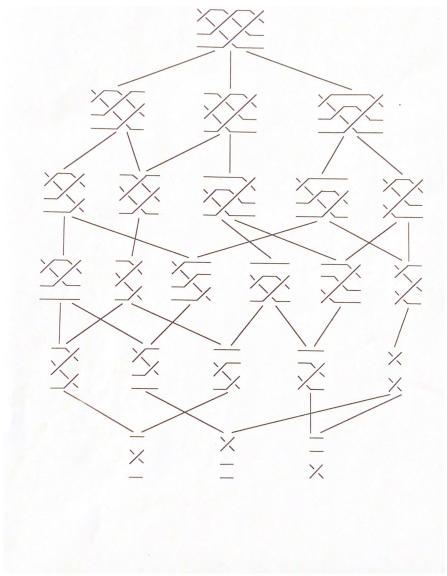


Figure 12: Permutation Braids in B_4

Going back to the factorization from above, we can say that $\mathcal{G}_n^i Z_0 = \mathcal{G}_n^i L$, where $L \in B_n^+$. Furthermore, if L consists of crossing where any two strands cross only once, then $L \in P$, and we can take a positive braid word, l_1, which has the same permutations as L to be it's representative. If not, set $L = l_1 l_1^*$ where l_1 is a maximal braid in L where any two strands cross at most once, and l_1^* is the rest of the braid. If any two strands cross at most once in l_1^*, then we say $l_1^* = l_2$ and stop. If not, we let $l_1^* = l_2 l_2^*$ where l_2 is maximal in l_1^*, and l_2^* is the rest of l_1^*. If any two braids in l_2^* cross at most once, we say $l_2^* = l_3$ and stop. If not we say, $l_2^* = l_3 l_3^*$, where l_3 is maximal in l_2^*. We repeat this process until we successfully factor $L = l_1 l_2 l_3 ... l_t$, where t is the number of times this process is done. So we have now successfully factored an arbitrary braid into $\mathcal{G}_n^i l_1 l_2 l_3 ... l_t$. This factoring is referred to as the Garside normal form, or even the "left greedy normal form", as we are pulling $\mathcal{G}_n^i$ towards the left of the word. Further details about this normal form can be found in [10].

6 Concluding Thoughts

The next logical step to take in this project is creating an algorithm which takes in any braid and outputs the normal form. Further reading on this algorithm can be found in [4]. The basic premise is to create a finite state machine which uses set theoretical logic to walk through the process as shown in Example 1. This machine not only holds importance in solving group theoretic problems, but also allows us to use B_n to solve other problems. For example, we can use the machine to encrypt information. Instead of forcing computers to run through simpler algorithms to factor primes, we force them to run through the more complicated algorithm demonstrated in Example 1. This idea can be seen in [8] and in [3].

Braids can also model certain physical situations in an interesting manner. As the braids have high levels of symmetries, they can model the symmetries of elementary particles. See [7] for further exploration of this application.

References

[1] Emil Artin. Theory of braids. *Annals of Mathematics*, 48, 1947.

[2] David Bessis. The dual braid monoid. *Annales scientifiques de l'École Normale Supérieure*, 36, 2003.

[3] Joan S. Birman and Tara E. Brendle. Braids: A survey. *arXiv:math/0409205v2 [math.GT]*, 2, 2004.

[4] David B.A. Epstein. *Word Processing in Groups*, volume 1. Jones and Bartlett Publishers, 1992.

[5] Joseph A. Gallian. *Contemporary Abstract Algebra*, volume 7. Brooks-Cole/Cengage Learning, Belmont, CA, 2010.

[6] F.A Garside. The braid group and other groups. *The Quarterly Journal of Mathematics*, 20, 1969.

[7] Jacak Janusz, Lucjan Jacak, and Ryszard Gonczarek. *Application of Braid Groups in 2D Hall System Physics*, volume 1. World Scientific, 2012.

[8] KiHyoungKo, SangJinLee, JungHeeCheon, JaeWooHan, Ju-sungKang, and ChoonsikPark. New public-key cryptosystem using braid groups. *Letcure Notes in Computer Science*, 1880, 2000.

[9] Wilhelm Magnus, Abraham Karrass, and Donald Solitar. *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations*, volume 2. Dover Publications, 2004.

[10] Simon-Philipp Merz and Christophe Petit. Factoring products of braids via garside normal form. *Lecture Notes in Computer Science*, 11443, 2019.

[11] Arthur Ogus. *lectures on Logorithmic Algebraic Geometry*, volume 1. Cambridge University Press, 2018.