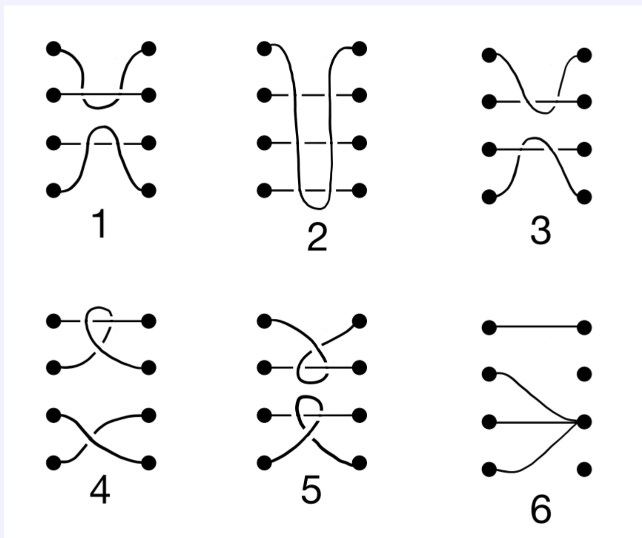
Grade 9/10 Math Circles

An Introduction to Group Theory Part 3 - Solutions

Exercise Solutions

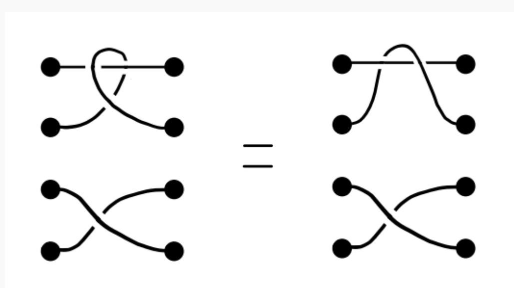
Exercise 1

Identify which of the below diagrams are 4-braids.



Exercise 1 Solution

Diagrams 1,2,3, and 4 are all 4-braids. At a quick glance, it looks like diagram 4 has a knot, however, if you pull the loop part of the second string towards the right, we get that:



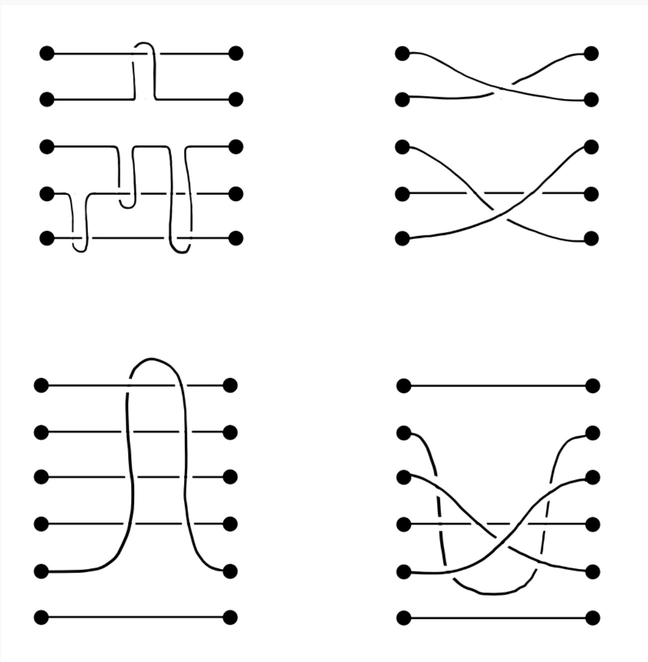
So, diagram 4 is indeed a 4-braid. Diagram 5 is not a 4-braid because it has a knot (in fact, two knots). Diagram 6 is not a 4-braid because not every dot has a string attached to it.

Exercise 2

Choose a small n, say $1 \leq n \leq 6$, and make n-braids. Feel free to use a few different n values.

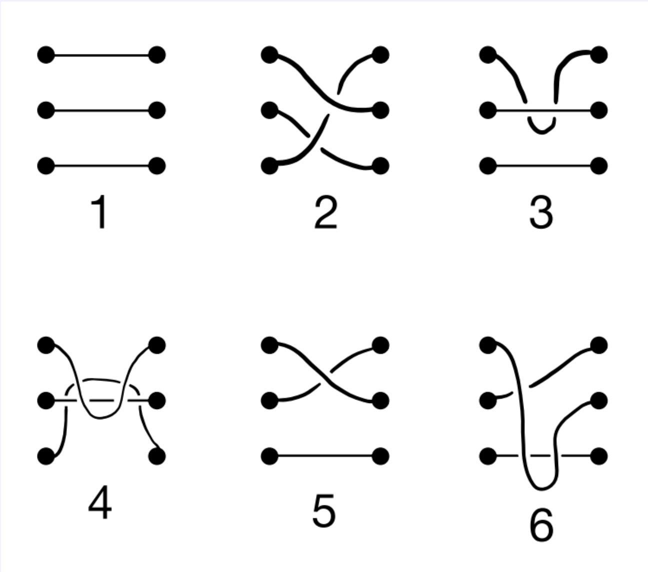
Exercise 2 Solution

Here are some examples of braids that look a bit more complicated:



Exercise 3

Consider the six 3-braids below. How many different 3-braids are there?



Exercise 3 Solution

There are 3 different 3-braids. Diagram 3 becomes diagram 1 by pulling the top string upwards. Diagram 4 becomes diagram 1 by pulling the top and bottom string upwards and downwards, respectively. Similarly, diagram 6 becomes diagram 5 by pulling the top string upwards. Since diagram 5 and 6 have one cross, they are not the same as diagram 1. Diagram 2 is not the same as any other diagram as it has two crosses. Given this, we can take diagram 1,2, and 5 to be the 3 different 3-braids.

Exercise 4

Try to define a binary operation $\bullet$ on B_n so that $(B_n, \bullet)$ is a group. Here are some suggestions and comments to help you get started:

- Asking for a binary operation on B_n is the same as asking "how can we combine two n-braids to make another n-braid?".
- Feel free to use a small n, say $n = 3$ or $n = 4$, to figure out the binary operation.
- Don't worry too much about trying to formalize the binary operation. Playing around and trying to combine explicit n-braids in a natural way is a great way to get you on the right track!
- Keep in mind that we want this binary operation and B_n to form a group. Once you have a guess of what the binary operation on B_n is, try to see if the group axioms hold for B_n and your binary operation.

Exercise 4 Solution

See pages 11-12 of lesson 3 for an explanation of the binary operation on B_n.

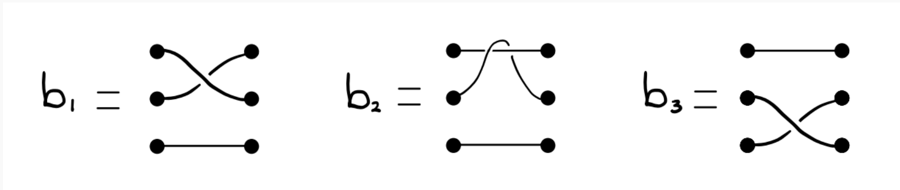
Exercise 5

Convince yourself that $(B_3, *)$ is a group.

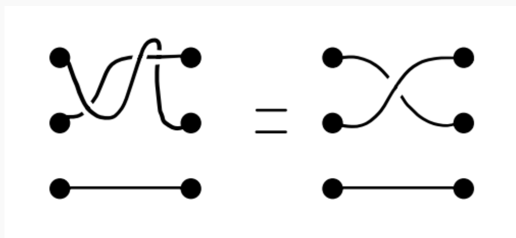
Exercise 5 Solution

We need to convince ourselves that the 3 group axioms hold for B_3 with concatenation. Let's go through each axiom:

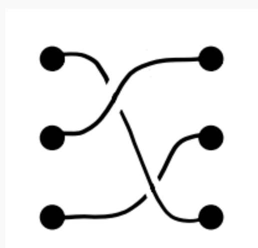
Axiom 1: For all $b_1, b_2, b_3 \in B_3$, we need $(b_1 * b_2) * b_3$ to be the same 3-braid as $b_1 * (b_2 * b_3)$. This is a bit tough to argue rigorously without introducing a lot of extra notation, so let's do a concrete example to see how it works. Consider the following 3-braids:



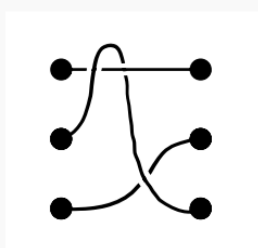
We compute $b_1 * b_2$ to be



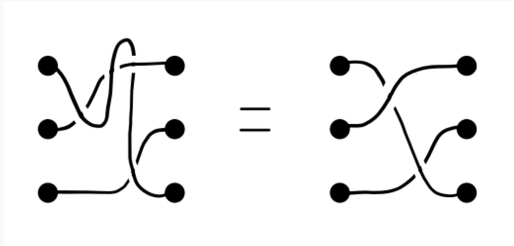
Concatenating this with b_3, we find that $(b_1 * b_2) * b_3$ is



Similarly, we find that $b_2 * b_3$ is

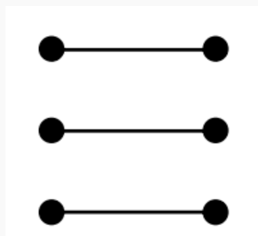


and hence $b_1 * (b_2 * b_3)$ is



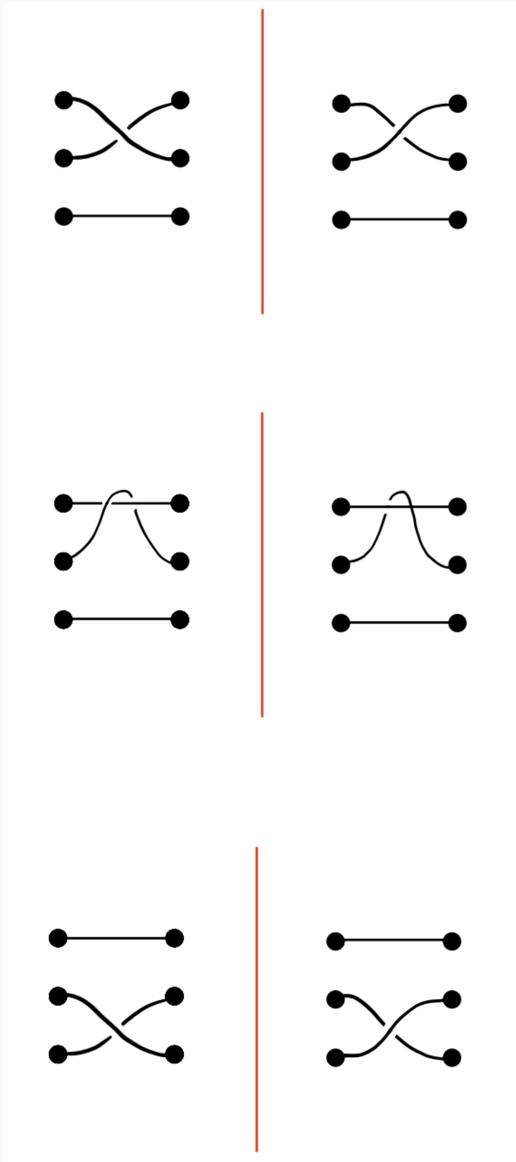
We see that $(b_1 * b_2) * b_3 = b_1 * (b_2 * b_3)$, as desired.

Axiom 2: The identity element in B_3 is the 3-braid



If we concatenate this braid with any other 3-braid, say $b \in B_3$, then the strings of b just get longer. Since the length of strings does not matter, the result is just b. So, this braid is the identity element in B_3 and we call it id_{B_3}.

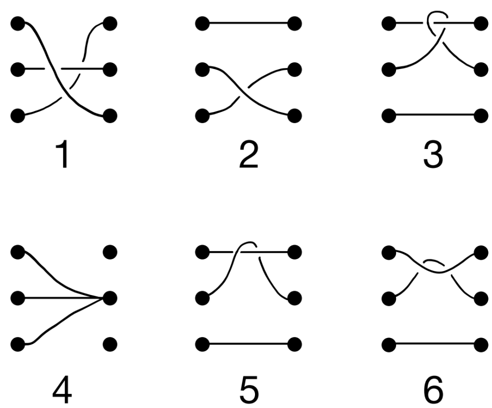
Axiom 3: Lastly, we want to show that each braid in B_3 has an inverse. There is a systematic way to obtain the inverse of any braid in B_3, which we illustrate through concrete examples. Basically, given a braid in B_3, we obtain it's inverse by flipping the braid over horizontally. To see this, consider the following photo:



We explain how to obtain the inverses of the 3 braids to the left of the red vertical lines. Flipping each of these braids over horizontally is the same as reflecting them in the red vertical line next to them. The result of these reflections are the braids to the right of the vertical lines. The braids on the right are the inverses of the respective braids on the left. We can do this with any braid to obtain it's inverse!

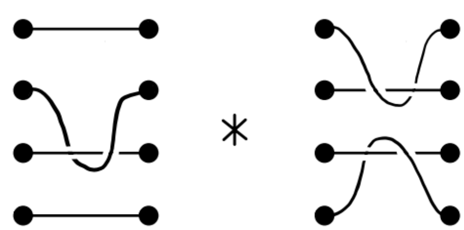
Problem Set Solutions

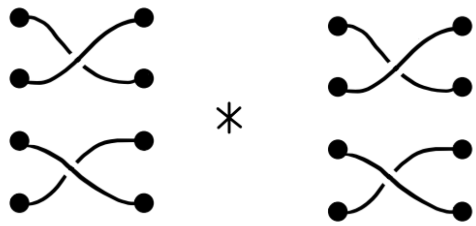
1. Identify which of the below diagrams are 3-braids.



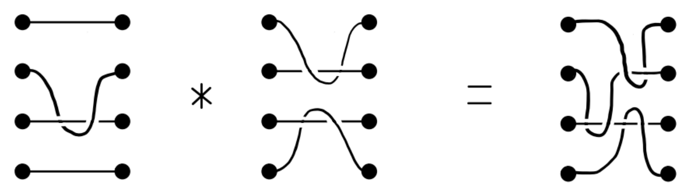
> *Solution*: Diagrams 1,2,5, and 6 are all 3-braids. Diagram 3 is not a 3-braid because it has a knot. Diagram 4 is not a 3-braid because not every dot has a string attached to it.

2. Compute the following two concatenations:

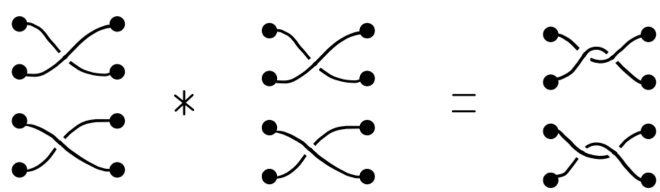




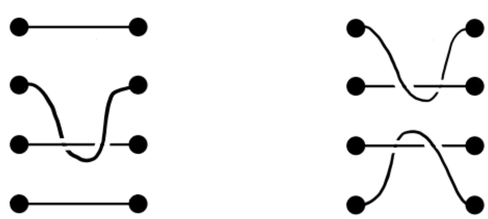
Solution: The first concatenation is



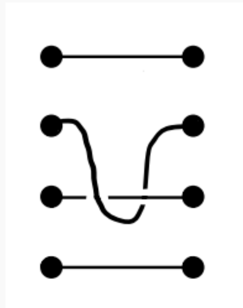
and the second concatenation is



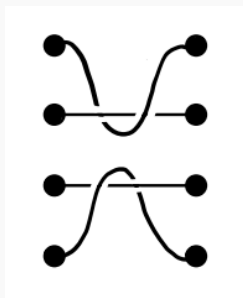
3. Compute the inverses of the following two braids:



Solution: The inverse of the first braid is



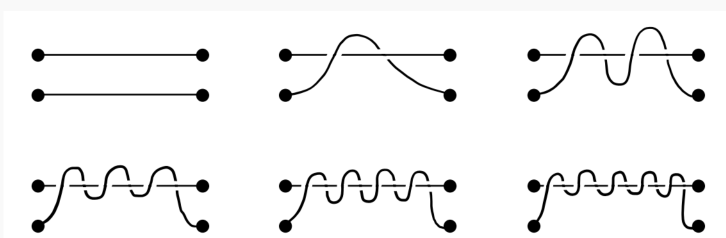
and the inverse of the second braid is



4. Consider the set B_n of all n-braids, where $n \in \{1, 2, 3, \dots\}$. Is B_n finite or infinite?

Solution: The set B_n is finite when $n = 1$ and infinite when $n \geq 2$. Let's see why this is true. First consider the case when $n = 1$. There is only one 1-braid, and it is

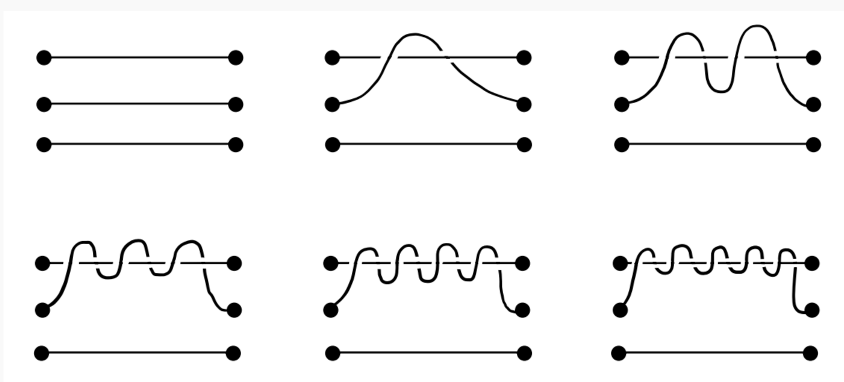
This means that there is only one element in B_1, and so B_1 is finite. Next, consider the case when $n = 2$. To see why B_2 is infinite, consider the following 2-braids:



In all of these 2-braids, we see that the second string wraps around the first string a various number of times. From left to right and top to bottom, we see that the number of wraps increases. From left to right, the 2-braids in the top row have 0 wraps, 1 wrap, and 2 wraps. From left to right, the 2-braid in the bottom row have 3 wraps, 4 wraps, and 5 wraps. We can keep increasing the number of these wraps forever, and so there are an infinite number of 2-braids.

Lastly, consider the case when $n \geq 3$. We will use B_2 to show that there are an infinite number of n-braids. We construct n-braids as follows. Take the last $n-2$ strings and dots to be

Then, we can think of the the remaining string connections (aka top 2 strings and pairs of dots) as a 2-braid. In other words, we have an "embedding" of B_2 into B_n. By varying the top two string connections through the elements of B_2, we generate an infinite number of n-braids (since B_2 is infinite). So, B_n is infinite. Here are some concrete examples of 3-braids constructed in this way:



We see that the top two string connections are just the 2-braids from above.