Building a Computer By Braiding Colorful Knots

Authors: Aishani Dutta, Suchetan Dontha, Yixiao Liu, Qingyu Meng

Graduate Mentor: William Linz

Faculty Advisor: Eric Samperton

Background

Groups

A ***group*** G is a set of elements, together with an associative binary operation that contains an inverse for each element and an identity element. Two fundamental groups we examined this semester are S_n and A_n:

- S_n = group of all permutations on *n* elements
- A_n = group of all *even* permutations on *n* elements
  - An even permutation requires an even number of swaps between elements

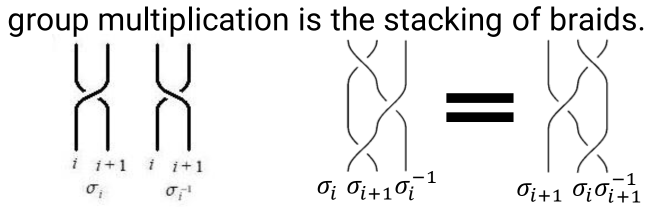
Group Actions

A ***group G acts on a set X*** **(G↷X)** if there is a homomorphism which maps the elements of G to permutations of the elements in *X*.

- The ***orbit*** of an element $x \in X$ is the set of elements $y \in X$ s.t. $g * x = y$ for some $g \in G$.
- Every group acts on itself by ***conjugation***: *G acts on G via the formula* $g * x = gxg^{-1}$
- A ***conjugacy class*** is an orbit of the group (as a set) under this action of conjugation

Braids

A ***braid*** is a sequence of crossings in which adjacent strands are passed over or under each other. The ***braid group*** *(denoted B_n)* is the group of all braids on *n* strands. The group multiplication is the stacking of braids.



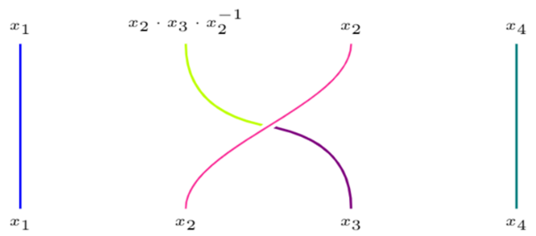
The braid relation and its inverse

Two equivalent braid words

The Hurwitz Action

Hurwitz action is the specific group action that we are using in our project.

$$\sigma_i \cdot (x_1, x_2, \ldots, x_n) = (x_1, x_2, \ldots, x_{i-1}, x_i \cdot x_{i+1} \cdot x_i^{-1}, x_i, x_{i+2}, \ldots, x_n).$$



An example of σ_2 is shown to the right.
The resulting strand 2 is $x_2 \cdot x_3 \cdot x_2^{-1}$.
Different colors represent different elements.

Constructions

We defined the set **C** to be the conjugacy class of the element (1 2 3 4 5) ∈ A_5 (the permutation that sends 1 to 2, 2 to 3, ..., 5 to 1). Then B_n/C^n is the quotient set of orbits under the Hurwitz action. We denote an orbit C_0^n if each c ∈ C_0^n satisfies the following two properties:

- A tuple c ∈ C^n has **trivial boundary monodromy** if its entries $c_1, \ldots, c_n$ satisfy $c_1 \cdots c_n$ = ( ), the identity permutation.
- A tuple c ∈ C^n **generates** A_5 if every element in A_5 can be realized as a finite length product of the entries in c (repeats are allowed).

In general, **Sym(X)** is the group of permutations amongst the elements of a set X. Then Sym(C_0^n) is the group of permutations amongst the elements of C_0^n. We define **$Sym_{A5}(C_0^n)$** to be the this group equivariant under A_5. Put simply, these are the elements σ ∈ Sym(C_0^n) such that ∀ α ∈ A_5, σα = ασ.

Our Goals

$$\rho : B_n \to Sym_{A_5}(C_0^n)$$

Define the ρ as shown on the left. We know that for n large enough, ρ(B_n) contains the **commutator subgroup** of $Sym_{A5}(C_0^n)$ which we will call the **Rubik subgroup** because a similar group appears in the group of Rubik's cube symmetries [1, 2]. That is, the group generated by the set $\{\sigma_1\sigma_2\sigma_1^{-1}\sigma_2^{-1} \mid \sigma_1, \sigma_2 \in Sym_{A5}(C_0^n)\}$. Our goal for this project was to computationally determine what value of n is sufficient. Once we found this n, our next goal was to determine if ρ is surjective, that is if ρ(B_n) = $Sym_{A5}(C_0^n)$.

Computing n

We decided to begin with the simplest case when n = 4. Thus the first step was to construct a C_0^4. We used Sage, a derivative of Python that has implementations of several concepts in group theory.

Calculating C_0^4

We first generated C^4, the set of 4-tuples of C, and constructed the set of its orbits under the Hurwitz action (algorithm shown below). We then identified the orbits whose tuples exhibited trivial boundary monodromy and generated A_5. In this case, there was one such orbit, C_0^4, where | C_0^4 | = 600.

```
orbits = []
for i in C4:
    if check_in_list_of_list(orbits,i):
        continue
    x = i; queue = []; queue.append(x); orbit = []; orbit.append(x)
    while (len(queue)!=0):
            x = queue[0]; x1 = braid(x,1); x2 = braid(x,2);...;x6 = braid_inverse(x,3)
            if (check_in_list(orbit,x1)==false):
                orbit.append(x1)
                queue.append(x1)
            if (check_in_list(orbit,x2)==false):
                ......
            queue.pop(0)
    orbits.append(orbit)
```

Pop one element from queue as s. Use 6 braid actions to get 6 results. Check if each result is present in a list storing existing values of s. If a result is not a duplicate, add it to the queue. Loop until queue is empty and we get the orbit that contains the first element entered in the queue.

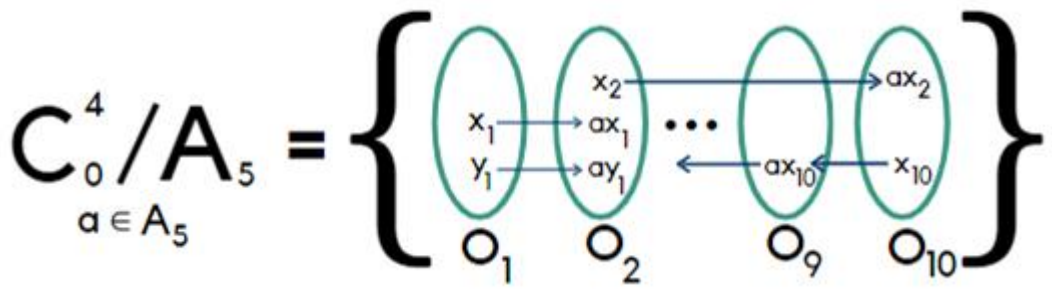
A Helpful Construction

We needed to prove whether ρ(B_4) contains the Rubik subgroup of $Sym_{A5}(C_0^4)$ which is the set of all permutations of the orbit C_0^4. To accomplish this, we defined some quantities that would help us prove a more approachable helpful construction:

- *Aut(A_5, C)* is the group of automorphisms of A_5 that fix C setwise; it is isomorphic to A_5
- *Sym(C_0^4 /A_5)* is the symmetric group of the orbits of A_5 acting on C_0^4
- $\bar{\rho}$ is the homomorphism defined to the right, which maps elements of B_4 to the set Sym(C_0^4/A_5)

$$\begin{tikzcd} B_4 \arrow[d, "\rho"] \arrow[dr, "\bar{\rho}"] & \\ Sym_{A_5}(C_0^4) \arrow[r, two heads, "\pi"] & Sym(C_0^4/A_5) \end{tikzcd}$$

Proving $\bar{\rho}$ is surjective is helpful for showing that ρ(B_4) contains the Rubik subgroup of $Sym_{A5}(C_0^4)$



- When we generated the elements of C_0^4/A_5, we found that it consisted of exactly 10 orbits with 60 elements each. Each orbit size must be 60 because |A_5| = 60 and A_5 acts freely on C_0^4
- Applying some automorphism α ∈ A_5 to every element in an orbit O_k will map each *o* ∈ O_k to some element in a different orbit, O_m. For example, every member of O_2 corresponds to some element in O_{10} when α is applied.
- As a result of this property, we can define an isomorphism $\bar{\Phi}$: Sym(C_0^4/A_5) → S_{10}. This allows us to prove that $\bar{\rho}$ is surjective as follows (cont'd on next column):
  - We know that b_1, b_2, and b_3 generate B_4 and that $\bar{\Phi} \circ \bar{\rho}$ is a homomorphism
  - This means that if ($\bar{\Phi} \circ \bar{\rho}$)($b_1$), ($\bar{\Phi} \circ \bar{\rho}$)($b_2$), and ($\bar{\Phi} \circ \bar{\rho}$)($b_3$) generate S_{10}, then $\bar{\rho}$ is surjective.

Computing n (cont'd)

```
sigma1A5 = phi_bar_rho_bar([1])
sigma2A5 = phi_bar_rho_bar([2])
sigma3A5 = phi_bar_rho_bar([3])
generatesGroup((sigma1A5, sigma2A5, sigma3A5), S10)

True
```

We were able to programmatically show that ($\bar{\Phi} \circ \bar{\rho}$)($b_1$), ($\bar{\Phi} \circ \bar{\rho}$)($b_2$), ($\bar{\Phi} \circ \bar{\rho}$)($b_3$) generate S_{10}. This proves that $\bar{\rho}$ is a surjective homomorphism when n = 4.

Surjectivity of ρ

Now that we can focus on determining if ρ is surjective. To do this, we must introduce a few more definitions:

Wreath Product

The **direct product** of two groups K and G (K × G) is a group over the set K × G (the cartesian product) and binary operation ⋅ defined by (k_1, g_1) ⋅ (k_2, g_2) = ($k_1 \cdot k_2$, $g_1 \cdot g_1$). Note that the two groups need not have the same binary operation.

The **semidirect product** of two groups K and G (K $\rtimes_\varphi$G), where φ: G -> Aut(K), is a direct product of K and G but binary operation ⋅ defined by (k_1, g_1) ⋅ (k_2, g_2) = ($k_1 \cdot \varphi(g_1) \cdot k_2$, $g_1 \cdot g_1$).

The **wreath product** of two groups K and G (K $\wr_\varphi$ G) is a semidirect product of the groups $K^{|G|} \rtimes_\varphi$ G, where $K^{|G|}$ is the direct product of K |G| times. G acts by permuting the factors of $K^{|G|}$

The **universal embedding theorem** states that any group extension of a group H by a group A is isomorphic to a subgroup of the regular wreath product A $\wr_\varphi$ H.

Equivalent Condition

$$\begin{tikzcd} ker(\pi) \arrow[r] \arrow[d, "\Psi"] & Sym_{A_5}(C_0^4) \arrow[r, two heads, "\pi"] \arrow[d, "\Phi"] & Sym(C_0^4/A_5) \arrow[d, "\bar{\Phi}"] \\ A_5^{10} \arrow[r, hook] & A_5 \wr S_{10} \arrow[r, two heads, "\bar{\pi}"] & S_{10} \end{tikzcd}$$

Using our map structure and the universal embedding theorem, we know that $Sym_{A5}(C_0^4)$ ~ $A_5^{10} \wr S_{10}$. We can define ψ as follows: each entry $\alpha_{i,\sigma}$ of a ∈ A_5^{10} satisfies $\alpha_{i,\sigma}x_i = \sigma x_j$, where σ is the permutation resulting from the braid and σx_j is in orbit i. For example, in the diagram, $\alpha_{10,\sigma}$ is the element of A_5 such that $\alpha_{10,\sigma}x_{10} = \sigma x_2$.

We can use the above definitions to restrict Φ to be (ψ ∘ ρ)(b), ($\bar{\Phi} \circ \bar{\rho}$)(b)). It can now be shown that ρ is surjective iff:

- ∀ σ ∈ S_{10}, ∃ b ∈ B_4 so that ($\bar{\Phi} \circ \bar{\rho}$)(b) = σ and (ψ ∘ ρ)(b) = (( ), . . . , ( ))
- ∀ a ∈ A_5^{10}, ∃ b ∈ B_4 so that (ψ ∘ ρ)(b) = a and ($\bar{\Phi} \circ \bar{\rho}$)(b) = ( )

Future Work

Future work is to verify that the two above conditions are true. The function phi_bar_rho_bar(b) shown before and generate_alphas(b) in our sage library return elements of $A_5^{10} \wr S_{10}$ given a b ∈ B_4. The next step is to systematically generate particular elements as described in the equivalent condition.

References

1. Ellenberg, Venkatesh, & Westerland. (2013). Homological stability for Hurwitz spaces and the Cohen-Lenstra conjecture over function fields, II. Retrieved from https://arxiv.org/abs/1212.0923

2. Kuperberg & Samperton. (2019). Coloring invariants of knots and links are often intractable. Retrieved from https://arxiv.org/abs/1907.05981

Support for this project was provided by the Illinois Geometry Lab and the Department of Mathematics at the University of Illinois at Urbana-Champaign.