Journal, Vol. XXI, No. 1, 1-5, 2013 Additional note

A search algorithm for finding valid ping-pong subsets in $\mathbb{RP}^1$[1]

Samuel Perales, Jordan Grant, Jeremy Krill, Abhay Katyal, Teddy Weisman, Jeff Danciger *Department of Mathematics, The University of Texas at Austin*
*__Corresponding author__:

Keyword1 — Keyword2 — Keyword3

...

February 14, 2023

Contents

Introduction

The Ping-Pong Lemma is a well known statement in group theory which lets us prove a group Γ is free by finding subsets of a space X which Γ acts on that meet certain conditions. The following is a statement of the lemma from Clay and Margalit (for more background and a proof, see [CM17]):

Lemma 0.1. *Ping-Pong for n-generators Suppose* $\{a_1, \dots, a_n\}$ *generate a group* G *which acts on space* X*. If*

1. X *has pairwise disjoint subsets* $\{X_1, \dots, X_n\}$*, and*
2. $a_i^k(X_j) \subset X_i$ *for all nonzero powers* k *and* $i \neq j$*,*

then G is a free group, with free generating set $\{a_1, \ldots, a_n\}$.

This paper focuses on a more recent generalization of the ping-pong lemma in the context of group actions on projective space, due to Avila, Bochi, and Yoccoz. Using this result (stated as Theorem 1.7 below), we describe an algorithm which can calculate explicit bounds on the size of the kernel of a representation of certain finitely generated groups into $SL(2, \mathbb{R})$. [Okay now say one sentence about finite-state automata and subsets of $\mathbb{RP}^1$.]

[Teddy write a paragraph about Anosov representations]

1 Generalized Ping Pong

1.1 Finite State Automata

In order to state a generalized version of the classical ping-pong lemma, we will need the machinery of finite state automata. What follows is a summary of the subject's basic definitions. For more background, see for example [Ree22].

Definition 1.1. For a finitely presented group G with finite generating set $X = \{a_1, \ldots, a_n\}$, a *word in G* is a finite string of k symbols which represents an element in X. We say the *length* of a word is $|w| = k$.

When working with groups it is often convenient to treat the word and the group element it represents as equivalent; however a single group element can be represented by an infinite number of words, which necessitates that we choose a particular representative. Typically the shortest representative, or *geodesic word*, is chosen.

Definition 1.2. If $g \in G$ we define the length of g, $|g|$ as the length of the shortest word over $X^{\pm}$ that represents g. Words of this form are also called *geodesic words*.

Strings are not the only way of representing group elements, and for this paper we will be primarily concerned with representing group elements using finite state automata.

Definition 1.3. A finite state automaton F_A where A is the alphabet, or symbol set, is a finite directed graph with edges labeled by one or more elements of A. One vertex, which for our purposes will always be the identity, is identified as the start.

Suppose that we have a finitely presented group G and a finite state automaton F_X where X is the generating set of G. Now with the labeling associated to F_X, consider a group element $g = a_{i_1} a_{i_2} \cdots a_{i_n}$. If we start at the vertex associated to the identity, then by following edges labeled with the generators comprising g as a word, we construct a path on F. If this path exists on F then I say that the word g is *accepted* by the finite state automaton F. This path then becomes a new, geometric representative of g. Unlike strings, for a given group G there is no guarantee that there exists a finite state automaton F that accepts every element of g. If such a graph exists, then we call G an *automatic group.*

Definition 1.4. For a finitely presented group G, if there exists a finite state automaton F which accepts every $g \in G$ then we call G an *automatic group.*

Our primary objects of study in this paper will be automatic groups whose path representatives on their associated finite state automata correspond to geodesic words. In yet another caveat, such a finite state automaton isn't guaranteed to exist for an automatic group G, and when it does then G is said to be *strongly geodesically automatic.*

1.2 Generalization

In the case of classical ping-pong, for a strongly geodesically automatic group G with 2 generators which acts on a space X, the existence of valid subsets of X yields information about the size of the kernel of the group action, namely that it is trivial. In the case of a representation $\rho : G \to SL(2,\mathbb{R})$, we apply a similar idea, where valid subsets of $\mathbb{RP}^1$ give us data about relations in $\rho(G) \subset SL(2,\mathbb{R})$, which in turn determines an explicit bound on the size of $\ker(\rho)$. To prove that this bound exists, we need to take advantage of some of the structure of group actions on $\mathbb{RP}^1$.

Definition 1.5. If I is an interval in $\mathbb{RP}^1$ with endpoints a, b, for any $x, y \in I$, we define the *Hilbert distance* $d_I(x,y)$ between x and y by

$$d_I(x,y) = |\log[a,x,y,b]|$$

Here $[a,x,y,b]$ is the cross-ratio $\frac{y-a}{x-a} \cdot \frac{x-b}{y-b}$, where the differences are measured in any affine identification of $\mathbb{RP}^1$ with $\mathbb{R} \cup \{\infty\}$.

Definition 1.6. For a matrix $A \in GL(2,\mathbb{R})$, the matrix norm of A is given by:

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}, x \in \mathbb{R}^2$$

Theorem 1.7 (see [ABY10], Theorem 2.2)**.** *Suppose Γ is a strongly geodesically automatic group with finite generating set $\{a_1, a_2, \cdots, a_n\}$ and $\rho : \Gamma \to SL(2,\mathbb{R})$ is a representation of Γ. Suppose that there exist non-empty open subsets $(M_v) \subset \mathbb{RP}^1$ with $\overline{M_v} \neq \mathbb{RP}^1$ for each vertex v such that if $v \xrightarrow{\alpha} u$ is a path accepted by the finite state automaton attached to Γ, then*

$$\overline{\rho(\alpha)(M_u)} \subset M_v.$$

Then there exist constants $C, \lambda > 1 \in \mathbb{R}^+$ such that if $\gamma \in \ker(\rho)$, then $|\gamma| \leq \left\lfloor \frac{\log(C)}{\log(\lambda)} \right\rfloor$.

We provide a proof of this result, because we will need some of the ideas to describe our algorithm. The proof below is taken directly from [ABY10] (where the theorem statement is given in slightly different language).

Definition 1.8. For a finite state automaton F, we define the *set of recurrent vertices* of F as those vertex's which have both an incoming and an outgoing arrow.

Remark. Ping-Pong sets need only be constructed for recurrent vertices on a finite state automaton. If a vertex has only incoming arrows then the finite state automaton will fail to define a regular language for the underlying group. If a vertex only has incoming arrows then we can simply have its associated valid subset be the union of the implied inclusions. Therefore for each of the valid subsets M_α in the proof below, we can safely assume that every point in M_α lies in the image of another subset M_β.

Proof. For each generator α, let d_α be the Riemannian metric on M_α which coincides with the Hilbert metric in each of its finite number of components. Let K_α be the closure of the union of the sets $\rho(\alpha)M_\beta$ where $\beta \to \alpha$. We can assume that K_α intersects each connected component of M_α, because otherwise we can take a smaller M_α. Let $\overline{L_\alpha} \subset M_\alpha$ be an open set containing K_α with the same number of connected components as M_α. Then each component $M_{\alpha,i}$ of M_α contains a unique component $L_{\alpha,i}$ of L_α. Let $\lambda = \min_{\alpha,i}\lambda(M_{\alpha,i}, L_{\alpha,i})$ where

$$\lambda(M_{\alpha,i}, L_{\alpha,i}) = \min_{x,y} \frac{d^L_{\alpha,1}(x,y)}{d^M_{\alpha,1}(x,y)}$$

and $d^M_{\alpha,1}$,$d^L_{\alpha,1}$ are the Hilbert metrics on the subsets $M_{\alpha,i}, L_{\alpha,i}$ respectively. Take an accepted path on our finite state automaton $v \xrightarrow{a_{I_1}} \cdots \xrightarrow{a_{I_k}} u$ and let $A = \rho(a_{I_1}) \cdots \rho(a_{I_k})$. If u, v belong to the same component of M_{I_0} then

$$d_{a_{I_k}}(Au, Av) \leq \lambda^{-k} d_{I_0}(u, v)$$

The metrics $d_\alpha|L_\alpha$ are comparable to the Euclidean metric on $\mathbb{RP}^1$. So if u, v belong to the same component of L_{I_0} we get $d(Au, Av) \leq C\lambda^{-k} d(u, v)$, where $C > 0$ is the greatest ratio between the Euclidean and Circle metrics. This in turn implies that

$$||A|| \geq C^{-1/2}\lambda^{k/2}.$$

□

The explicit construction of the constants C, λ are given in sections 2.3.1 and 2.3.2 respectively.

Definition 1.9. Suppose Γ is a strongly geodesically automatic group with finite state automaton F_Γ. A collection of open subsets $(M_v) \subset \mathbb{RP}^1$, for all vertices v in the F_Γ, is called *valid subsets* if they satisfy the conditions of Theorem 1.7.

1.3 Stability Properties

Discussion of stability properties of these ping-pong sets, namely that they imply

2 Computational Approach

To find a collection of valid subsets for a particular strongly geodesically automatic group Γ, finite state automate F_Γ, and representation $\rho : \Gamma \to SL(2, \mathbb{R})$, we start with an initial guess at these subsets and repeatedly refine the guess, checking for validity at each step. If a solution is found, it is not unique, and if no solution exists, the algorithm will run forever. The general outline for this refined guessing procedure is as follows:

1. Initialize a small subset around a point in $\mathbb{RP}^1$ for some vertex v of F_Γ which we are confident must be contained in the valid subset M_v.

2. For each edge of the automaton, check if containment required by the edge is satisfied with the current configuration of intervals. If all containments are satisfied, we have found valid subsets and we can move to step 4. Otherwise, continue to step 3.

3. For each image that failed containment, create a small open interval containing the image and merge it with the subset which should have contained it. Then, go back to step 2.

4. Calculate an explicit bound on the kernel of the representation based on the valid subsets found.

5. Check all words of length less than N, where N is the bound on the kernel given by the constants C and λ in Theorem 1.7. If none of the words are the identity, we know the representation is faithful.

2.1 Initializing Subsets

In order to begin a search for a collection of valid subsets (M_v) with v a vertex in F_Γ, it helps to first identify points which each M_v must necessarily contain. Once we find these, we can then build our subsets around them. First, we introduce some important terminology to be used throughout the description of the algorithm.

Definition 2.1. For $\epsilon > 0$ and a point $x \in \mathbb{RP}^1$, let the *ϵ-neighborhood* of x be

$$N_\epsilon(x) = \{z \in \mathbb{RP}^1 : d(x,z) < \epsilon\}$$

Definition 2.2. The *singular value decomposition* of a matrix $M \in \mathbb{R}^{2x2}$ is the factorization $M = U\Sigma V^T$ where U is an orthogonal matrix with columns called *singular directions*, Σ is a diagonal matrix of *singular values*, and V^T is the transpose of another orthogonal matrix. We denote the greatest singular value of M as $\sigma_{max}(M)$ and its associated singular direction $v_{max}(M)$.

Lemma 2.3. *Suppose (A_i) is a sequence of $SL(2,\mathbb{R})$ matrices, with singular value decomposition $A_i = U_i \begin{pmatrix} \lambda_i & 0 \\ 0 & \lambda_i^{-1} \end{pmatrix} V_i^*$ and suppose also that the sequences of left and right, singular vectors $\sigma^i_{R,1}, \sigma^i_{R,2}, \sigma^i_{L,1}, \sigma^i_{L,2}$ converge to $\sigma_{R,1}, \sigma_{R,2}, \sigma_{L,1}, \sigma_{L,2}$. Suppose also that $\lim_{i\to 0} \lambda_i = \infty$. Then for any $x \in \mathbb{RP}^1/\{\sigma_{R,1}\}$*

$$\lim_{i\to 0} A_i(x) = \sigma_{R,1}$$

Proof. Since the singular vectors converge

$$\lim_{i\to 0} A_i(x) = \lim_{i\to 0} \begin{pmatrix} \sigma^i_{R,1} & \sigma^i_{R,2} \end{pmatrix} \begin{pmatrix} \lambda_i & 0 \\ 0 & \lambda_i^{-1} \end{pmatrix} \begin{pmatrix} \sigma^i_{L,1} & \sigma^i_{L,2} \end{pmatrix} = \lim_{i\to 0} \begin{pmatrix} \sigma_{R,1} & \sigma_{R,2} \end{pmatrix} \begin{pmatrix} \lambda_i & 0 \\ 0 & \lambda_i^{-1} \end{pmatrix} \begin{pmatrix} \sigma_{L,1} & \sigma_{L,2} \end{pmatrix}$$

Since each $A_i \in SL(2,\mathbb{R})$ our U_i, V_i are rotations of the form

$$\begin{pmatrix} cos(\theta_i) & -sin(\theta_i) \\ sin(\theta_i) & cos(\theta_i) \end{pmatrix} = U_i, \begin{pmatrix} cos(\phi_i) & -sin(\phi_i) \\ sin(\phi_i) & cos(\phi_i) \end{pmatrix} = V_i$$

Therefore the limits of our right hand singular vectors, $\sigma_{R,1}, \sigma_{R,2}$, are $\begin{pmatrix} cos(\theta) \\ sin(\theta) \end{pmatrix} \begin{pmatrix} -sin(\phi) \\ cos(\phi) \end{pmatrix}$ with $\theta, \phi \in [0, 2\pi)$ respectively. Now suppose $x = a_1e_1 + a_2e_2$, and consider $\lim_{i\to\infty} A_i(x)$.

$$\begin{aligned} \lim_{i\to\infty} A_i(x) &= \lim_{i\to\infty} a_1A(e_1) + a_2A(e_2) \\ &= \lim_{i\to\infty} (\lambda_i(a_1cos(\phi_i) + a_2sin(\phi_i))\sigma^i_{R,1} + \lambda_i^{-1}(a_1sin(\phi_i) + a_2cos(\phi_i))\sigma^i_{R,2}) \\ &= \lim_{i\to\infty} \lambda_i(a_1cos(\phi) + a_2sin(\phi))\sigma_{R,1} + \lim_{i\to\infty} \lambda_i^{-1}(a_1sin(\phi) + a_2cos(\phi))\sigma_{R,2} \\ &= \lim_{i\to\infty} \lambda_i(a_1cos(\phi) + a_2sin(\phi))\sigma_{R,1} \end{aligned}$$

Since $\sigma_{R,1}, \sigma_{R,2}$ form an orthonormal basis for $\mathbb{R}^2$, we can write this limit in this coordinate scheme as $\lim_{i\to\infty}\begin{pmatrix}\lambda_i(a_1 cos(\phi) + a_2 sin(\phi))\\ 0\end{pmatrix} = \lim_{i\to\infty}\begin{pmatrix}\lambda_i\\ 0\end{pmatrix} = \begin{pmatrix}\infty\\ 0\end{pmatrix}$. Projectivizing both sides of the above equation to $\mathbb{RP}^1$ then yields

$$\lim_{i\to\infty}[A_i(x)] = \left[\begin{pmatrix}\infty\\ 0\end{pmatrix}\right]$$

which is identified with the basis vector $\sigma_{R,1}$. □

Theorem 2.4. *Suppose that we have a representation $\rho : G \to SL(2,\mathbb{R})$ where G is a finitely presented automatic group with finite state automaton F_G, generating set $(A_1, \cdots, A_n)$, and a set of valid subsets $(M_v) \subsetneq \mathbb{RP}^1$. Then for any vertex v on F, if $v \xrightarrow{A_{i_1}} \cdots \xrightarrow{A_{i_m}} u$ is an accepted path on F_G and (V^m) is the sequence of maximal singular directions defined by the product sequence $V^m = v_{max}(B^m) = v_{max}(\rho(A_{i_1})\cdots\rho(A_{i_m})$, then every accumulation point of (V^m) lies in M_v.*

Proof. If we consider a vertex v on F, then due to the direction of the inclusions described in theorem 1.7, a path $v \xrightarrow{A_{i_1}} \cdots \xrightarrow{A_{i_m}} u$ defines an inclusion $\rho(A_{i_1})\cdots\rho(A_{i_m})(M_u) \Subset M_v$ where $M_u, M_v \subsetneq \mathbb{RP}^1$ are valid subsets. Now suppose that the limit of maximal singular directions $\lim_{m\to\infty} V^m = v_0$ lies outside of M_v. Then by 2.3, $\lim_{m\to 0} B^m(M_u) = s_0 \notin M_v$, implying that for every $x \in M_u$ there exists some $n_0 \in \mathbb{N}$ such that $B^{n_0}(x) \notin M_u$, contradicting our assumption that M_u and M_v are valid subsets.

□

By taking paths starting at a vertex v through F_Γ and considering the maximal singular directions of matrices represented by these paths, we can get approximations of a point that we know must be inside the valid subset M_v.

To start the algorithm, we choose a vertex v_0 and a random path $v_0 \xrightarrow{A_1} \cdots \xrightarrow{A_n} v_n$ of our finite state automaton with some large length n (explicitly, our code uses $n = 100$). We take the singular direction with largest singular value of the matrix $A_n \cdots A_1$ associated to this path, $v_{max}(A_1 \cdots A_n)$, and set our initial guess at subset M_v^0 to be $N_\epsilon(v_{max})$. We set all other subsets to initially be the empty set.

2.2 Permeating Subsets

Once the subset M_v^0 is initialized, we can start to compute what the valid subsets must be. This is done by a method which we call *patching*. Suppose the finite state automaton associated to our automatic group requires $A_k M_v \subset M_u$. We can then update our guess for interval M_u to be:

$$M_u^{i+1} = M_u^i \cup N_\epsilon(A_k M_v^i)$$

In general, if the finite state automaton has multiple outward pointing edges at vertex u corresponding to multiple containment conditions required by Theorem 1.7, we can update our guess M_u^{i+1} to be the union of M_u^i and all $N_\epsilon(A_k M_v^i)$ given by the finite state automaton.

By taking the union of our previous guess with all of these 'patches' around the necessary images, we guarantee that M_u^{i+1} contains all $A_k M_v^i$ and therefore satisfies all containment conditions required by Theorem 1.7. Note that although the ping-pong Lemma requires subsets to be connected, the generalized statement of ping-pong allows our subsets to be disconnected.

Since updating our guess for M_u causes it to grow, the containment conditions of other subsets may break after iterating. Each iteration of the algorithm, all subset guesses are sequentially patched. If none of the subsets grow during an iteration, we know that all containments conditions have been satisfied and that our collection (M_v) of subsets are a set of valid subsets for the representation.

2.3 Verifying Injectivity

Theorem 1.7 implies that if the algorithm described above terminates after finding valid subsets of $\mathbb{RP}^1$, then the maximum word length of any group element in the kernel of the representation $\rho : G \to SL(2, \mathbb{R})$ we are working with is bounded by an expression in terms of certain constants λ, C. [recall what λ and C are here.] To verify that the representation ρ is actually faithful, we compute λ and C explicitly, and then check that $\rho(g)$ is nontrivial for every $g \in G$ with $|g| < \log(C)/\log(\lambda)$.

2.3.1 Computing λ

Let us first expand the definition, supposing that the subsets $\overline{L_\alpha} \subset M_\alpha$ have endpoints $[-c, c] \subset (-1, 1) \subset \mathbb{R} \cup \{\infty\}$ under the projective chart P defined in section 1.2:

$$\begin{aligned}
\lambda &= \min_{\alpha,i} \lambda(M_{\alpha,i}, L_{\alpha,i}) \\
&= \min_{\alpha,i,x,y} \frac{d^M_{\alpha,1}(x,y)}{d^L_{\alpha,1}(x,y)} \\
&= \min_{\alpha,i,x,y} \frac{\log[-c,x,y,c]}{\log[-1,x,y,1]} \\
&= \min_{\alpha,i,x,y} \frac{\log(\frac{y+c}{x+c}\frac{x-c}{y-c})}{\log(\frac{y+1}{x+1}\frac{x-1}{y-1})}
\end{aligned}$$

To find this minimum we take derivatives of the resulting function which we call f. We get that $f_y(x,y) = 0$ and

$$f_x(x,y) = \frac{(\frac{1}{x-1} - \frac{1}{x+1})(\log\frac{x-c}{y-c} + \log\frac{y+c}{x+c})}{(\log\frac{x-1}{y-1} + \log\frac{y+1}{x+1})}$$

leaving us with critical points along the x-axis with $x = \pm 1, \pm c, 0$. However, since we only care about $x, y \in (-c, c)$, we can ignore all of these except for the origin, which must be the minimum. Therefore

$$\lambda = \min_{\alpha,i} \lim_{(x,y)\to(0,0)} \frac{\log(\frac{y+c}{x+c}\frac{x-c}{y-c})}{\log(\frac{y+1}{x+1}\frac{x-1}{y-1})} = \frac{1}{c}$$

Now we have an explicit value for λ given that the nested subsets are symmetric and centered about 0 under P. Since $[a, x, y, b] = [A(a), A(x), A(y), A(b)]$ for $A \in SL(2, \mathbb{R})$ and there is always such a transformation that will take four points of $\mathbb{RP}^1$ to four of the form $[-c, c] \Subset (-1, 1)$, we can always explicitly compute λ in this way.

2.3.2 Computing C

Let M_{a_i} be a component of some M_α with endpoints F, H. and we take a closed interval $[c, d] \subset [F, H] \subsetneq \mathbb{RP}^1$. We want to be able to compare this metric to the round metric d_s on S^1. For simplicity's sake, we first rotate $[F, H]$ until it is of the form $[-a, a]$ for some $a \in \mathbb{R}^+$. We then make a comparison of the Hilbert metric to the standard Euclidean metric on $\mathbb{R}$ pointwise on the interval $[c, d]$, which we do by using the ratio of the Hilbert distance

$$|\log(\frac{x+a}{x-a} \cdot \frac{x+h+-a}{x+h+a})|$$

to the euclidean distance h, as $h \to 0$, which is simply the derivative of the Hilbert distance.

$$\lim_{h \to 0} \frac{|\log(\frac{x+a}{x-a} \frac{x+h+-a}{x+h+a})|}{h} = \frac{2a}{a^2 - x^2}$$

We then make the same comparison between the round distance and euclidean distance, which is simply the derivative of $\arctan(x)$, $\dfrac{1}{1+x^2}$.

We now want to find two constants D_1, D_2 for our interval $[c, d]$ which satisfy the inequality

$$D_1^{-1} d_I(u, v) \leq d_s(u, v) \leq D_2^{-1} d_I(u, v).$$

For D_1 we want to maximize the Hilbert distance while also minimizing the round distance, therefore we aim to maximize $\frac{2a}{a^2-x^2}$, and minimize $\frac{1}{1+x^2}$. For D_2 we do the inverse, minimizing $\frac{2a}{a^{;}-x^2}$ and maximizing $\frac{1}{1+x^2}$. Have $x_0 = \max\{|c|, |d|\}$ and $y_0 = \min\{|c|, |d|\}$. Then,

$$D_1 = \frac{2a(1+x_0^2)}{a^2 - x_0^2}$$

$$D_2 = \begin{cases} \frac{a}{2} & c < 0 < d \\ \frac{a^2-y_0^2}{2a} \cdot \frac{1}{1+y_0^2} & \text{otherwise} \end{cases}$$

Referring back to our notation from theorem 1.7, if one of our $M_{\alpha,i}$ has n connected $L_{\alpha,i}$, then we repeat this process for every $L_{\alpha,i}$ yielding a family of D_1^i, D_2^i's. We set $C_1 = \max\limits_{1 \leq k \leq n}(D_1^i)$, $C_2 = \min\limits_{1 \leq k \leq n}(D_2^i)$. Then referring back to our equation at the end of theorem 1.7,

$$C_1^{-1} d_s(Au, Av) \leq d_{a_{I_k}}(Au, Av) \leq \lambda^{-n} d_{a_{I_0}}(u, v) \leq C_2^{-1} \lambda^{-k} d_s(u, v)$$

$$d_s(Au, Av) \leq C_2^{-1} C_1 \lambda^{-k} d_s(u, v)$$

$$\boxed{C = C_2^{-1} C_1}$$

3 Results and Discussion

3.1 Valid Subsets

Our algorithm was able to compute explicit subsets of $\mathbb{RP}^1$ which meet the conditions of Theorem 1.7 and verify the faithfulness of the representations in a number of simple examples. The following subsections include the group presentation, $SL(2,\mathbb{R})$ representation, and valid subsets of $\mathbb{RP}^1$ mapped to $[0,\pi)$.

3.1.1 Free Product of Finite Cyclic Groups

As an initial test of the algorithm, we input free products of finite cyclic groups with presentations of the form:

$$\langle a, b | a^n = b^m = 1 \rangle$$

We created representations for these groups by mapping a to a rotation by $\frac{\pi}{n}$ and b to a conjugated rotation by $\frac{\pi}{m}$. In particular, this example shows valid subsets for $n = 2$ and $m = 3$. We note that there are often multiple finite state automata which can describe a group, but any choice will work with the algorithm. Below, you'll find examples of a couple finite state automata for a free product of cyclic groups with orders 2 an 3 (for computing our valid subsets, we used the left-most graph):

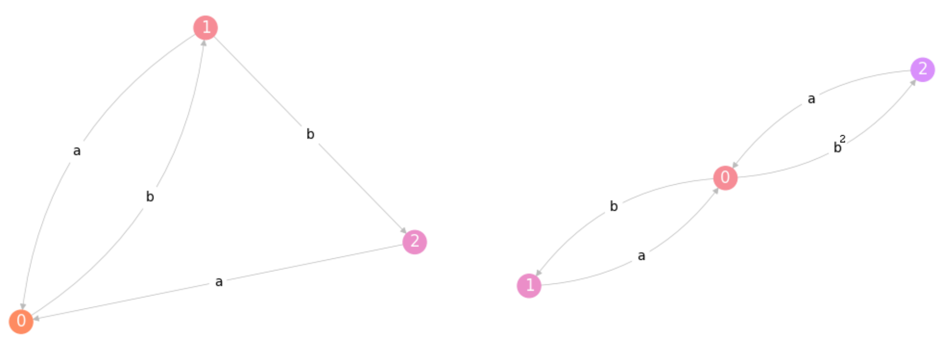

Figure 1: Several finite-state automata for free products of cyclic groups with $n = 2, m = 3$

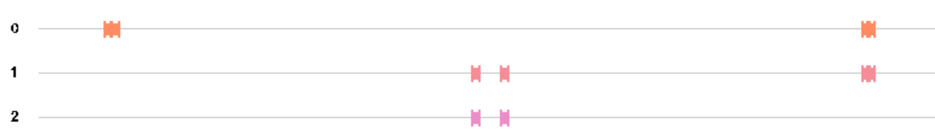

Figure 2: Valid subsets for a representation of a free product of finite cyclic groups with $n = 2, m = 3$

3.1.2 (3,3,4)-Triangle Group

We used the following presentation of the (3,3,4)-triangle group along with the usual geometric representation to try and find valid ping-pong subsets:

$$\langle a, b, c | a^2 = b^2 = c^2 = 1, (ab)^3 = (ac)^3 = (cb)^4 = 1 \rangle$$

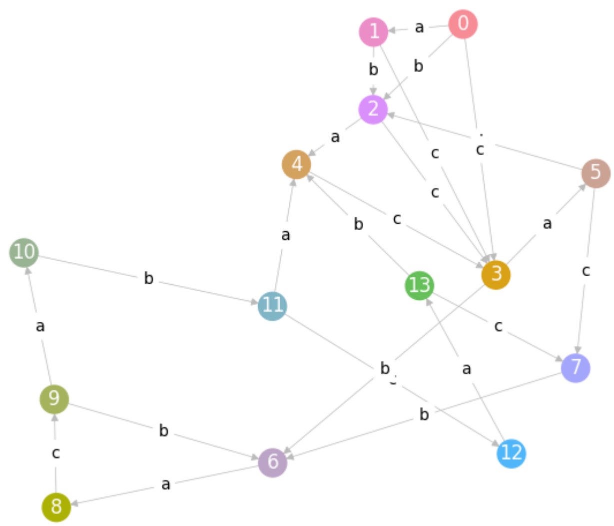

Figure 3: A finite-state automaton which describes the (3,3,4)-triangle group

3.1.3 Surface Group

Our third test case was the surface group:

$$\langle a, b, c, d | adc^{-1}ba^{-1}d^{-1}cb^{-1} = 1 \rangle$$

along with a the representation given by ... .

We note that with certain presentations of surface groups, small numerical errors tended to diverge forcing us to run the algorithm with interval expansion values of $\epsilon < 10^{-6}$. In these cases however, the trade-off was runtime. Without further optimized code, it is unlikely to find valid subsets for presentations such as the one below:

$$\langle a, b, c, d | [a, b][c, d] = 1 \rangle$$

which comes with a one-parameter family of representations given by ... .

3.2 Notes on the Implementation

There are several potential speed and memory improvements which could be made to broaden the range of representations which the algorithm is able to find valid subsets for.

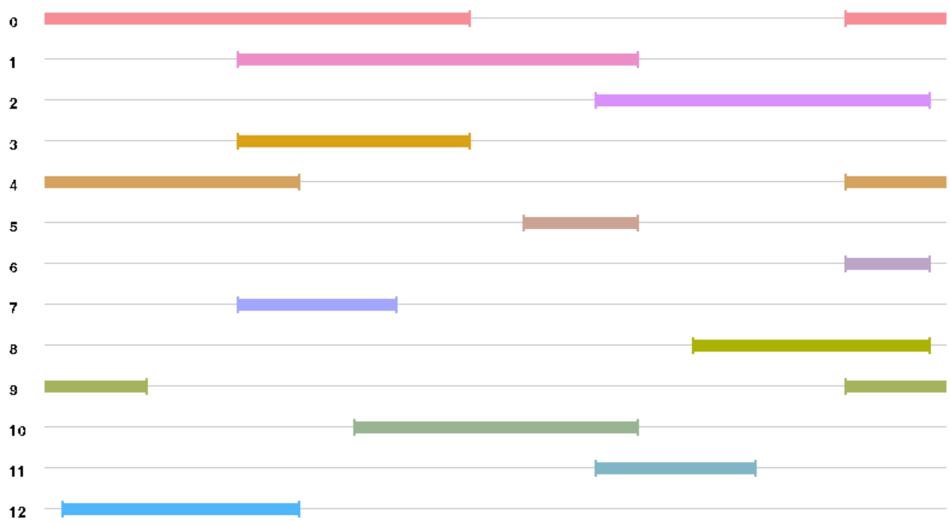

Figure 4: Valid subsets for a triangle group representation

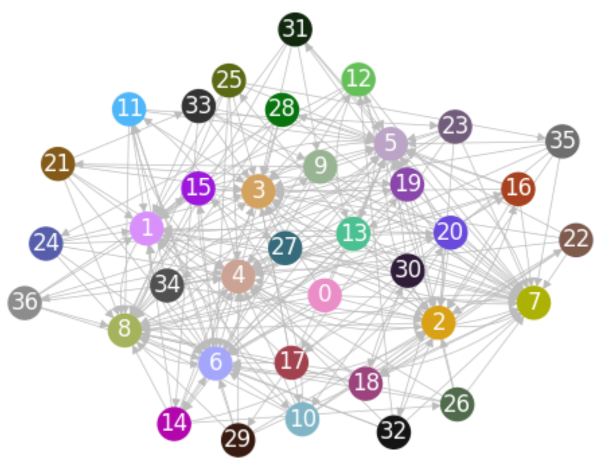

Figure 5: A finite-state automaton which describes the surface group (edge labels omitted)

3.3 Extension to $\mathbf{RP}^n$

Not all groups can be represented in $\mathbb{RP}^1$. Instead, to develop and apply an extension of this algorithm to these groups would mean searching for our subsets in $\mathbb{RP}^n$. Many of the underlying mechanics would remain in place, the primary change being the way subsets are stored and permeated.

In $\mathbb{RP}^1$, the subsets we are tracking are simply intervals of the form (a, b). As a first step, moving to $\mathbb{RP}^2$ requires a choice; should subsets be calculated as balls, cubes, tetrahedrons,

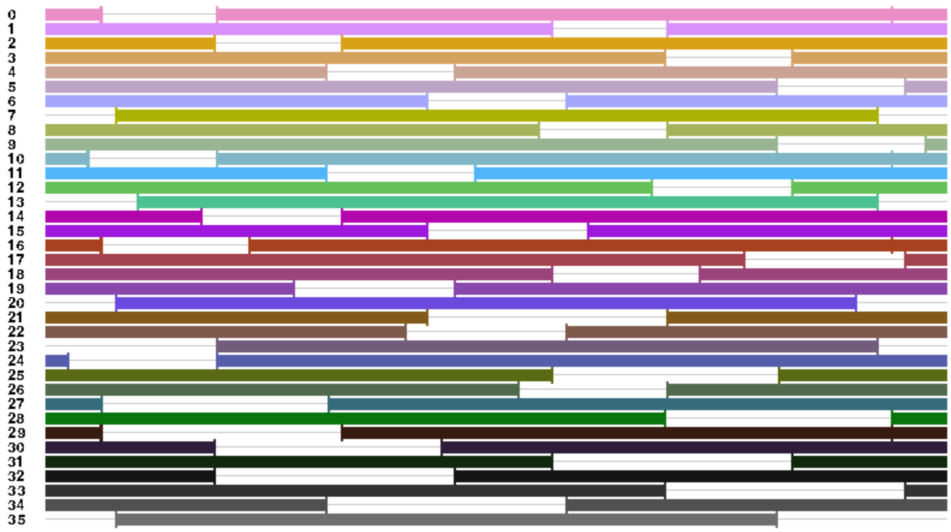

Figure 6: Valid subsets for a surface group representation

perhaps some other shape? The simplest approach that we would recommend is initializing subsets as tetrahedrons centered about maximal singular directions and patching intervals by taking the convex hull of the points which make up a subset and open subset over the images it must contain.

This extension to higher dimensions of real-projective space is necessary to expand our input to other groups, but it would also be interesting to see if representations of groups we've already demonstrated could be moved to $SL(n, \mathbb{R})$ for faster convergences or tighter bounds on the kernel.

Acknowledgments

Thanks Teddy
Thanks Jeff
Thanks NSF

References

[ABY10] Artur Avila, Jairo Bochi, and Jean-Christophe Yoccoz. "Uniformly hyperbolic finite-valued SL(2,$\mathbb{R}$)-cocycles". In: *Commentarii Mathematici Helvetici* (2010), pp. 813–884. DOI: 10.4171/cmh/212. URL: https://arxiv.org/pdf/0808.0133.pdf.

[CM17] Matt Clay and Dan Margalit. *Office Hours with a Geometric Group Theorist*. Princeton University Press, 2017. ISBN: 9781400885398. DOI: doi:10.1515/9781400885398. URL: https://doi.org/10.1515/9781400885398.

[Ree22] Sarah Rees. *The development of the theory of automatic groups*. 2022. DOI: 10.48550/ARXIV.2205.14911. URL: https://arxiv.org/abs/2205.14911.