

ON COMPUTING FINITE INDEX SUBGROUPS OF $\mathrm{PSL}_2(\mathbb{Z})$

NICOLÁS MAYORGA URUBURU, ARIEL PACETTI, AND LEANDRO VENDRAMIN

ABSTRACT. We present a method to compute finite index subgroups of $\mathrm{PSL}_2(\mathbb{Z})$. Our strategy follows Kulkarni's ideas, the main contribution being a recursive method to compute bivalent trees as well as their automorphism group. As a concrete application, we compute all subgroups of index up to 20. We then use this database to produce tables with several arithmetical properties.

1. INTRODUCTION

Let $\mathrm{PSL}_2(\mathbb{Z})$ be the modular group, obtained as the quotient of the set of all two by two matrices with integral entries of determinant 1 modulo the subgroup $\{\pm \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\}$. It is a classical problem that of determining all subgroups of $\mathrm{PSL}_2(\mathbb{Z})$ of a given finite index. In a remarkable article Newman ([13]) computed the number of subgroups of index up to a hundred (including the asymptotic behavior of the counting function). The group $\mathrm{SL}_2(\mathbb{Z})$ acts on the set of subgroups by conjugation, and it is also a natural problem that of determining the subgroups (or their number) up to $\mathrm{SL}_2(\mathbb{Z})$ -equivalence. Similarly, the group $\mathrm{GL}_2(\mathbb{Z})$ acts on $\mathrm{PSL}_2(\mathbb{Z})$ so the same sort of questions for $\mathrm{GL}_2(\mathbb{Z})$ -equivalence classes can be considered. In [16] the author gave a method to compute the number of $\mathrm{SL}_2(\mathbb{Z})$ -equivalence classes, whose sequence corresponds to the integer sequence *A121350* (see <https://oeis.org/A121350>).

From a computational point of view, it is challenging to actually compute tables of subgroups of $\mathrm{PSL}_2(\mathbb{Z})$ (modulo conjugation). To our knowledge, it was Kulkarni (in [9]) the first one who actually proposed an algorithm to compute them. In the aforementioned article (in Appendix 1) he even gives a list of subgroups of index up to 6 (for these small indexes, there are only a few of them). Let us give some details on Kulkarni's approach, which also shows the relevance of the problem.

Let $\mathcal{H}^* = \{z \in \mathbb{C} \mid \mathrm{Im} z > 0\} \cup \{\infty\} \cup \mathbb{Q}$ denote the extended complex upper half plane. The group $\mathrm{SL}_2(\mathbb{Z})$ acts on \mathcal{H}^* by Möbius transformations. Since the matrix $\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$ acts trivially, we get a well defined action of the modular group $\mathrm{PSL}_2(\mathbb{Z})$. The typical fundamental domain \mathcal{T} for the action of $\mathrm{PSL}_2(\mathbb{Z})$ on \mathcal{H}^* is the hyperbolic triangle with vertices $\rho = \exp(2\pi i/6)$, ρ^2 and ∞ . The half-plane \mathcal{H} has an extra transformation ι (as a real variety) given by $\iota(z) = -\bar{z}$. This allows to extend the action of $\mathrm{PSL}_2(\mathbb{Z})$ on \mathcal{H}^* to an action of $\mathrm{PGL}_2(\mathbb{Z})$ on \mathcal{H}^* given by

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot z = \begin{cases} \frac{az+b}{cz+d} & \text{if } ad - bc = 1, \\ \frac{a\bar{z}+b}{c\bar{z}+d} & \text{if } ad - bc = -1. \end{cases}$$

The hyperbolic triangle \mathcal{F} with vertices ρ , i and ∞ (see Figure 1) is a fundamental domain for the action of $\mathrm{PGL}_2(\mathbb{Z})$ on \mathcal{H}^* . Clearly $\mathcal{T} = \mathcal{F} \cup \iota(\mathcal{F})$. Following [9] in the present article we use the following terminology.

Definition 1.1. The points of \mathcal{H} in the orbit of i (resp. in the orbit of ρ) by the action of $\mathrm{GL}_2(\mathbb{Z})$ are called *even or red vertices* (resp. *odd or blue vertices*). The points in the orbit of $\mathrm{SL}_2(\mathbb{Z}) \cdot \infty$ are called *cusps*.

The $\mathrm{PGL}_2(\mathbb{Z})$ -translates of the triangle \mathcal{F} provide a tessellation of \mathcal{H} . The hyperbolic geodesic of \mathcal{F} joining i with ∞ (resp. joining ρ with ∞) is called an *even edge* (resp. *odd edge*) and so is all of its translates by $\mathrm{GL}_2(\mathbb{Z})$. The hyperbolic geodesic of \mathcal{F} joining i with ρ is of finite length. Its $\mathrm{GL}_2(\mathbb{Z})$ -translates are called *f-edges*.

2010 *Mathematics Subject Classification.* 11F06, 05C85.

Key words and phrases. Bi-valent graphs, Subgroups of the modular group.

N.M.U. was partially supported by a Conicet grant, AP was partially supported by the Portuguese Foundation for Science and Technology (FCT) within project UIDB/04106/2020 (CIDMA), L.V. was supported by project OZR3762 of Vrije Universiteit Brussel.

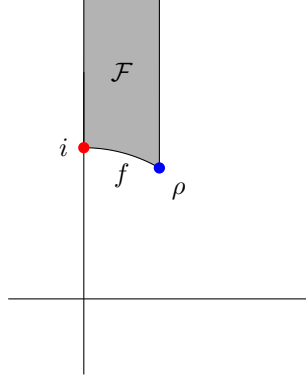


FIGURE 1. Fundamental domain for $\mathrm{PGL}_2(\mathbb{Z})$.

Let Δ be a finite index subgroup of $\mathrm{PSL}_2(\mathbb{Z})$. Then a fundamental domain for its action is given by the union of translates of \mathcal{T} . A key idea introduced by Kulkarni in [9] was to replace the fundamental domain \mathcal{T} by

$$(1) \quad \mathcal{D} = \mathcal{F} \cup \iota(S \cdot \mathcal{F}),$$

where $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$. In this way, any fundamental domain obtained as the union of translates of \mathcal{D} will have no f -edges on its border. Furthermore, if the fundamental domain has nice properties, the set of f -edges inside the fundamental domain form a graph with an orientation on it. It is expected that the subgroup Δ could be recovered (up to conjugation) from the graph. This allowed Kulkarni to give three different ways to describe a finite index subgroup:

- (1) Via the graph of f -edges on the quotient $\Delta \backslash \mathcal{H}$, what Kulkarni called a bipartite cuboid graph.
- (2) Via removing from the whole graph of f -edges those vertices having valence 2 and making some cuts to the graph in order to get a tree, what Kulkarni called a tree diagram.
- (3) Via listing the cusps appearing on a “special” fundamental domain, and listing how the paths going through them are glued together. This was called a Farey symbol by Kulkarni.

For theoretical purposes, the first approach is the best one, as on the one hand it contains all the information needed to recover the subgroup and on the other one it is in bijection with the set conjugacy classes of subgroups. However, the second one is better for computational purposes (which is the goal of the present article) since the tree has a smaller number of edges.

In [1, §3.1] the authors gave an alternative method to describe a finite-index subgroup of $\mathrm{PSL}_2(\mathbb{Z})$, following the ideas introduced by Millington in [12]. Roughly speaking, if Δ is a subgroup of $\mathrm{PSL}_2(\mathbb{Z})$ of index d , then $\mathrm{PSL}_2(\mathbb{Z})$ acts by left multiplication on right coset representatives of $\Delta \backslash \mathrm{PSL}_2(\mathbb{Z})$ (in particular the action of each element of $\mathrm{PSL}_2(\mathbb{Z})$ is given by a permutation of \mathbb{S}_d). To describe the action completely it is enough to determine the permutations attached to a set of generators of $\mathrm{PSL}_2(\mathbb{Z})$, for example the permutation attached to the matrix S introduced before and the one attached to the matrix $T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ (the usual translation matrix). This way of representing a subgroup is also called a “passport” in the literature. The method was extended in [15], where a table containing the permutations attached to all subgroups (up to $\mathrm{GL}_2(\mathbb{Z})$ -equivalence) of index up to 12 (and up to index 18 in electronic format) is given. Stromberg’s computations were extended in different ways (including computing tables of modular forms) in [3] and [4].

The purpose of the present article is to provide an algorithm to compute finite index subgroups (up to $\mathrm{SL}_2(\mathbb{Z})$ and $\mathrm{GL}_2(\mathbb{Z})$ equivalence) via computing tree diagrams. A *bi-valent* tree is a tree whose set of valences has two elements. Our main contribution (of interest on its own) is to provide a recursive algorithm to construct bi-valent trees. Along the way we prove interesting properties of the automorphism group of a bi-valent tree (see Theorem 3.8) which is crucial to speed up the algorithms used to compute tree diagrams.

Since our final goal is to provide tables of finite index subgroups of $\mathrm{PSL}_2(\mathbb{Z})$ (up to both $\mathrm{SL}_2(\mathbb{Z})$ and $\mathrm{GL}_2(\mathbb{Z})$ equivalence) we also provide the needed algorithms to construct the subgroup attached to a tree diagram (as well as its passport representation). Most of these algorithms are already part of the literature. Our humble contribution is to include missing details (that appeared while writing the code) regarding

“boundary issues” as well as correcting a few mistakes. Of particular interest is the algorithm to construct the generalized Farey symbol attached to a tree diagram (see Theorem 4.7). Although such an algorithm plays a crucial role in [9], somehow there is no description of such an algorithm in the article, so we take the opportunity to present it here, and prove its correctness.

Our algorithms are implemented in the computer software GAP [6]. The code can be downloaded from <https://github.com/vendramin/subgroups> with DOI 10.5281/zenodo.8113635. The GitHub repository contains some precomputed data as well as a tutorial (with examples) on how to use the code. The package depends on the packages: **Yags** for graph theory, [5] for some calculations related to generalized Farey symbols, and **HomAlgTools** for speeding up recursive functions.

The article is organized as follows: Section 2 contains a quick review of Kulkarni’s main results used in the present article. It includes some definitions (used during the article) as well as the correspondence between subgroups and some particular fundamental domains (called special polygons). This result justifies our computation of tree diagrams. Section 3 contains the main recursive algorithm to compute bi-valent trees as well as their automorphism group. Although our trees have valence $\{1, 3\}$, the proven results hold for any tree with valence set $\{1, n\}$, n being arbitrary. Of particular interest is Theorem 3.8, which describes the automorphism group of a bi-valent tree as a semi-direct product of two other ones. The section includes algorithms used to add an orientation and a coloring on the set of external vertices to a bi-valent tree.

Section 4 contains the definition of a generalized Farey symbol (g.F.s. for short), which consists of the set of cusps of a special polygon. As previously mentioned, a given g.F.s. together with the gluing of the sides allow to give an alternative description of a finite index subgroup of $\mathrm{PSL}_2(\mathbb{Z})$. This section contains an algorithm to compute a g.F.s. attached to a tree diagram (Theorem 4.7), providing a way to go from the second description to the third one. A tree diagram together with a g.F.s. is called a *Kulkarni diagram*. The same section contains different algorithms to, given a Kulkarni diagram corresponding to a subgroup Δ , compute arithmetic information of the quotient curve $\Delta \backslash \mathcal{H}^*$ (like cusp representatives and width of the cusps, genus, ramification points, etc) as well as information of the subgroup Δ itself (including a reduction algorithm needed to solve the word problem, and finding a set of coset representatives).

Section 5 contains the definition of a passport and algorithms to compute the passport attached to a Kulkarni diagram (to compare with other tables in the literature). At last, Section 6 contains some tables and interesting facts that can be deduced from the data we computed.

It should be clear to the reader how Kulkarni’s article crucially influenced our work, so we strongly recommend they to look at the article [9] which contains more details on the correspondences between tree diagrams, Farey symbols and bipartite cuboid graphs.

2. A QUICK GUIDE TO KULKARNI’S APPROACH

Recall the following definition from [9, §4.1].

Definition 2.1. A *bipartite cuboid graph* is a finite graph whose vertex set is divided into two disjoint subsets V_0 (the red ones) and V_1 (the blue ones) such that

- (1) every vertex in V_0 has valence 1 or 2,
- (2) every vertex in V_1 has valence 1 or 3,
- (3) there is a prescribed cyclic order on the edges incident at each vertex of valence 3 in V_1 ,
- (4) every edge joins a vertex in V_0 with a vertex in V_1 .

Let Δ be a finite index subgroup of $\mathrm{PSL}_2(\mathbb{Z})$. The quotient $\mathcal{H}^* \backslash \Delta$ is a compact oriented real surface, with a tessellation given by the translates of \mathcal{D} . The graph of f -edges of the surface is an example of a cuboid graph, where V_0 corresponds to the red vertices (i.e. the translates of i) while V_1 corresponds to the blue vertices (the translates of ρ).

We will only use bipartite cuboid graphs twice on the present article: once in Example 2.8, to prove that two non-isomorphic tree diagrams provide conjugated subgroups and also while computing the generalized Farey symbol attached to a tree diagram. In contrast to a bipartite cuboid graph, the notion of a tree diagram appears while working with a fundamental domain (a *special polygon*) for the quotient. The advantage of this approach is that we do not need to glue its edges together, it is enough to store which/how edges get identified. In this way we can work with a tree instead of a graph.

Definition 2.2. A *special polygon* is a convex hyperbolic polygon \mathcal{P} whose boundary $\partial\mathcal{P}$ is a union of even and odd edges, satisfying the following properties:

- (1) The even edges in $\partial\mathcal{P}$ come in pairs, each pair forming a complete hyperbolic geodesic called *even line*.
- (2) The odd edges in $\partial\mathcal{P}$ come in pairs. The edges in each pair meet at an odd vertex, making an internal angle of $\frac{2\pi}{3}$.
- (3) There exists an involution on the edges of $\partial\mathcal{P}$ so that no edge is carried into itself.
- (4) The involution sends an odd edge into another odd edge, making an internal angle of $\frac{2\pi}{3}$ between themselves.
- (5) Let e_1, e_2 be two even edges in $\partial\mathcal{P}$ forming a even line. Then either e_1 is paired to e_2 , or else $\{e_1, e_2\}$ form a *free side* of $\partial\mathcal{P}$ and this free side is paired to another such free side of $\partial\mathcal{P}$.
- (6) 0 and ∞ are two of the vertices of \mathcal{P} .

Remark 2.3. Note that any special polygon has a canonical orientation induced from the “counterclockwise” orientation of \mathcal{H} . The action of $\mathrm{SL}_2(\mathbb{Z})$ on a special polygon preserves the orientation (since it corresponds to a holomorphic function). However, the action of the matrix $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ has the effect of reversing the orientation of our special polygon. The choice of the opposite orientation corresponds to the action by ι on \mathcal{H} . The choice of the counterclockwise orientation or its inverse accounts for the difference between equivalence classes of subgroups up to $\mathrm{SL}_2(\mathbb{Z})$ conjugation or up to $\mathrm{GL}_2(\mathbb{Z})$ conjugation. This will prove crucial later (see Remark 2.6).

Theorem 2.4 (Kulkarni). *A special polygon is a fundamental domain for the subgroup Δ generated by the side-pairing transformations and these transformations form an independent set of generators for Δ . Conversely every subgroup Δ of finite index in $\mathrm{PSL}_2(\mathbb{Z})$ admits a special polygon as a fundamental domain.*

Proof. See page 1055 of [9]. □

A special polygon is the union of translates of \mathcal{D} , so as explained before, we can look at its tree of f -edges. Properties (4) and (5) of a special polygon imply that the end points of the f -edges tree will be points that are one of:

- the intersection of two odd edges, or
- the intersection of two even edges paired together, or
- the intersection of two even edges forming a free side.

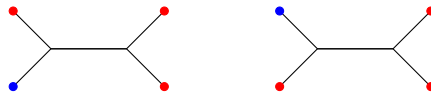
Any vertex of the tree has valence 1 (corresponding to an end point), 2 or 3. To avoid redundant information on the tree, one can remove the vertices having valence 2, obtaining the so called *tree diagrams*. A great advantage of the tree diagram is that its number of vertices tends to be much smaller than that of the original bipartite cuboid graph. Remark 2.3 implies that the vertices with valence 3 have a natural orientation.

Definition 2.5. A *tree diagram* is a tree with at least one edge such that

- (1) all internal vertices are of valence 3,
- (2) there is a prescribed cyclic order on the edges incident at each internal vertex (orientation),
- (3) the terminal vertices are partitioned into two possibly empty subsets R and B (red and blue vertices),
- (4) there is an involution ι on R .

By $\mathrm{Bv}(n, 3)$ we will denote the set of tree diagrams with n internal vertices.

Example 2.6. It is not true in general that if \mathfrak{T} is a tree diagram, and we consider the tree diagram where the orientation at all internal vertices are inverted (by considering the inverse of the permutation) we get isomorphic tree diagrams. For example, the following planar (with the anti-clockwise orientation) tree diagrams in $\mathrm{Bv}(2, 3)$ are not isomorphic. They correspond to two subgroups of $\mathrm{PSL}_2(\mathbb{Z})$ which are not isomorphic via conjugation under $\mathrm{SL}_2(\mathbb{Z})$, but are isomorphic under conjugation by $\mathrm{GL}_2(\mathbb{Z})$.

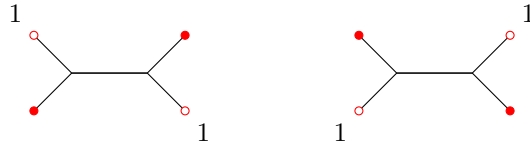


The way to relate a tree diagram with a bipartite cuboid graph as follows: remove from the bipartite cuboid graphs all vertices of valence 2 (connecting the remaining endings) and make some cuts to the graph so that it becomes a tree (see [9, §4.4] for details). Conversely, given a tree diagram, color blue all of its internal vertices (i.e. they are odd vertices) and keep the coloring of the tree diagram on the external vertices. If two consecutive vertices of the tree are blue, enlarge the tree by adding one extra vertex and paint it with red color. Finally, we need to glue together the identified external vertices: identify two external vertices v and w in R if $\iota(v) = w$, where ι is the involution of R . It is not hard to verify that the resulting graph is a bipartite cuboid graph.

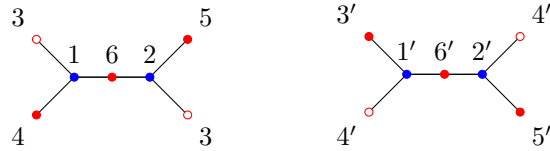
Remark 2.7. Following Kulkarni's notation, the way we encode the involution ι is as follows: if v is a vertex fixed by ι then we just save its color. Otherwise, there exists $w \neq v$ such that $\iota(v) = w$. In this case, we add a label to each of the two vertices.

As will be described in Section 4, there is a finite-to-one surjective map between the set of tree diagrams and the set of conjugacy classes of finite index subgroups of $\mathrm{PSL}_2(\mathbb{Z})$. This map unfortunately is not injective.

Example 2.8. Consider the following tree diagrams with anti-clockwise orientation:



These two tree diagrams are not isomorphic: the right tree satisfies that if we move from a red vertex (following the orientation) we end on a free one, while this is not true for the left one. The bipartite cuboid graph attached to each one of them are the following:



where the edges labeled with the same name are glued together. The two new graphs are isomorphic under the map

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 2' & 1' & 6' & 5' & 3' & 4' \end{pmatrix}.$$

It is easy to verify that this map preserves orientations.

This is the first example where two non-isomorphic tree diagrams give conjugated subgroups (corresponding to subgroups of index 6). However, it seems that for larger index subgroups (when there are many free sides) the number of tree diagrams is much larger than the number of $\mathrm{SL}_2(\mathbb{Z})$ -classes of subgroups, for example there are 28 tree diagrams corresponding to subgroups of order 9 while only 14 $\mathrm{SL}_2(\mathbb{Z})$ -equivalence classes. See the data in Table 6.1.

Remark 2.9. Working with bipartite cuboid graphs has the advantage that their isomorphism classes are in bijection with $\mathrm{SL}_2(\mathbb{Z})$ conjugacy classes of subgroups of $\mathrm{PSL}_2(\mathbb{Z})$, but as already mentioned, has the disadvantage that it implies working with much larger graphs (which are not trees in general).

Let \mathfrak{T} be a tree diagram and ι its involution. The set of its external vertices is naturally decomposed as the disjoint union of three sets

$$(2) \quad V_e(\mathfrak{T}) = B \cup R_0 \cup R_1,$$

where B are the blue vertices, R_0 are the red vertices fixed by the involution, and R_1 are the red vertices not fixed by the involution (an even set). Let $b = |B|$ (its cardinality), $r = |R_0|$ and $f = \frac{|R_1|}{2}$. The main goal of the next section is to provide an algorithm to compute (equivalence classes of) tree diagrams with parameters (b, r, f) .

3. BI-VALENT TREES

Definition 3.1. A *bi-valent tree* is a tree satisfying that the valence function on vertices takes at most two different values.

Clearly the valence function on any tree with at least three vertices must take at least two different values, hence the valence function on a bi-valent tree with at least three vertices takes precisely two different values.

Definition 3.2. Let T be any bi-valent tree. The set of *external vertices* (resp. *internal vertices*) that we denote by V_e (resp. V_i) is the set of vertices of T having valence 1 (resp. having valence > 1).

Denote by $Bv(m, n)$ the set of bi-valent trees (up to isomorphism) with valences set $\{1, n\}$ made of m internal vertices.

Example 3.3. There is a unique bi-valent tree T in $Bv(2, 3)$ given by the graph shown in Figure 2.

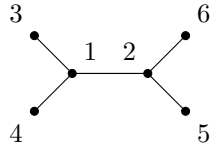


FIGURE 2. Unique tree in $Bv(2, 3)$.

Its adjacency matrix equals

$$\left(\begin{array}{cc|cccc} \textcolor{red}{0} & \textcolor{red}{1} & 1 & 1 & 0 & 0 \\ \textcolor{red}{1} & \textcolor{red}{0} & 0 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{array} \right),$$

where the red block is precisely the adjacency matrix of the internal vertices.

Lemma 3.4. Let T be a bi-valent tree, satisfying that the valence of any vertex belongs to the set $\{1, n\}$ for some $n > 1$. Then

$$|V_e| = (n - 2)|V_i| + 2.$$

Proof. By induction on the size $|V_i|$. If $|V_i| = 1$ then there must be n external vertices (all joined to the unique internal one), hence the result. Suppose that $m = |V_i| > 1$ and that the result holds for all bi-valent trees with at most $m - 1$ internal vertices. Let v, w be internal vertices joined by an edge (such a pair always exists because $m > 1$). Cut the tree into two disjoint trees T_1, T_2 , by removing the edge (v, w) and add to each of the new trees an external edge joining v (respectively w). Now each tree is again a bi-valent tree, say with r and s internal vertices (so $r + s = m$). The inductive hypothesis implies that

$$|V_e(T_1)| = |V_i(T_1)|(n - 2) + 2 \quad \text{and} \quad |V_e(T_2)| = |V_i(T_2)|(n - 2) + 2.$$

But $|V_i(T)| = |V_i(T_1)| + |V_i(T_2)|$ and $|V_e(T)| = |V_e(T_1)| + |V_e(T_2)| - 2$ (because we add a new external vertex to both T_1 and T_2), hence the result. \square

The lemma implies that if $T \in Bv(m, n)$ then its number of vertices equals

$$|V(T)| = (n - 1)m + 2.$$

3.1. Automorphisms of bi-valent trees. Let T be a bi-valent tree, and let $T_i = (V_i, E_i)$ be the sub-tree of internal vertices, namely the tree whose set of vertices $V_i = V_i(T)$ is the set of internal vertices of T , and whose set of edges E_i is the set of all edges of T between internal vertices. Given an automorphism σ of $\text{Aut}(T)$, it makes sense to restrict σ to the subgraph T_i , giving a sequence

$$(3) \quad 1 \longrightarrow \text{Aut}_e \longrightarrow \text{Aut}(T) \xrightarrow{\text{Res}} \text{Aut}(T_i),$$

where Res is the restriction map (a group morphism) and Aut_e is (by definition) its kernel. It turns out that the sequence (3) is exact and furthermore it splits. Before proving these facts we need a characterization of the subgroup Aut_e .

Recall that given two vertices of a tree, there exists a unique reduced path joining them, so there is a natural definition of *distance* between vertices. For latter purposes, if the bi-valent tree T has at least three vertices, we denote by

$$\Phi: V_e \rightarrow V_i,$$

the function which assigns to an external vertex v the unique internal vertex which is at distance one from it. Let $v \in V_e$ be an external vertex, and let $\text{Two}(v)$ be the set of external vertices at distance at most 2 from v . Since $v \in \text{Two}(v)$, $\text{Two}(v) \neq \emptyset$. Consider the following function

$$m: V_e \rightarrow [1, \dots, n], \quad m(v) = |\text{Two}(v)|.$$

For $1 \leq i \leq n$, let

$$V_e^{(i)} := \{v \in V_e \mid m(v) = i\}.$$

Then we can decompose the set of external vertices as the disjoint union of the sets $V_e^{(i)}$. For $i \in [1, \dots, n]$, define

$$r(i) = \max\{|S| : S \subseteq V_e^{(i)} \text{ and } d(v, w) \neq 2 \text{ for all } v, w \in S\}.$$

Lemma 3.5. *There is a non-canonical group isomorphism:*

$$\text{Aut}_e \simeq \prod_{i=1}^n \mathbb{S}_i^{r(i)},$$

where \mathbb{S}_i denotes the symmetric group on i elements.

Proof. Let $\sigma \in \text{Aut}(T)$ be an automorphism in the kernel of the restriction map and let $v \in V_e^{(i)}$ be an external vertex. Let $w = \Phi(v)$ the internal vertex adjacent to v . Since σ is in the kernel of the restriction map, $\sigma(w) = w$. Then $\sigma(v)$ is an external vertex which is also adjacent to w (since any automorphism preserves distances). In particular, $\sigma(v) \in \text{Two}(v) \subseteq V_e^{(i)}$ and σ induces a permutation of the elements of $\text{Two}(v)$ (a set with i elements). By definition, there exists elements $w_1, \dots, w_{r(i)}$ such that the set $V_e^{(i)}$ can be written as the disjoint union

$$V_e^{(i)} = \bigsqcup_{j=1}^{r(i)} \text{Two}(w_j),$$

so the action of σ on $V_e^{(i)}$ can be represented as $r(i)$ permutations on \mathbb{S}_i . It is clear that such map is bijective, i.e. permutations on the (disjoint union of the) sets $\text{Two}(w_j)$ induce an automorphism of the tree T which fixes all internal vertices. \square

Let T and T' be two bi-valent trees in $\text{Bv}(m, n)$. For the purposes of the present article, an *endomorphism* between T and T' is a bijective map between both the set of vertices of T to the set of vertices of T' and between the set of edges of T to the set of edges of T' (with the usual compatibility condition). We denote by $\text{End}(T, T')$ the set of all such maps.

For computational purposes, all trees considered will have its set of edges labeled by positive integers. This provides a total order on the graph's set of vertices.

Definition 3.6. Let T and T' be two trees with a total order on their set of vertices. We say that a map $\sigma \in \text{End}(T, T')$ is 2-ordered if it satisfies that for all pair of external vertices $v, w \in V_e(T)$ at distance two, it holds that if $v < w$ then $\sigma(v) < \sigma(w)$.

Proposition 3.7. *Let T and T' be bi-valent trees in $\text{Bv}(m, n)$ with a total order on its set of vertices. Let T_i (resp. T'_i) be the sub-tree of inner vertices of T (resp. T'_i be the sub-tree of inner vertices of T'). Let $\phi: T_i \rightarrow T'_i$ be a morphism of graphs. Then ϕ can be extended in a unique way to a 2-ordered morphism ψ of $\text{End}(T, T')$. Furthermore, if $T_i = T'_i$ and $T = T'$, the natural map $\psi: \text{Aut}(T_i) \rightarrow \text{Aut}(T)$ is a group morphism.*

Proof. To avoid confusion, we denote by val the valence function on either tree T or T' and by val_i the valence function on their sub-tree of internal vertices. Let $v \in V_e(T)$ be any external vertex and let $w = \Phi(v)$ be

the internal vertex adjacent to it. Since T is bi-valent and $w \in T_i$, $\text{val}(w) = n$ and $\text{val}_i(w) < n$ (because v is not in the internal sub-tree). The map ϕ preserves valences (since it is a morphism from T_i to T'_i) hence $\text{val}_i(\phi(w)) < n$ and so it must be joined to an external edge v' of T' . The two sets $\text{Two}(v)$ and $\text{Two}(v')$ have the same cardinality (equal to $n - \text{val}_i(w)$). Clearly any extension of ϕ must send the elements of $\text{Two}(v)$ to the ones in $\text{Two}(v')$, and since both sets are ordered sets, there exists a unique bijection between them preserving the order, providing the required extension.

When $T = T'$ and $T_i = T'_i$, a priori the map $\psi: \text{Aut}(T_i) \rightarrow \text{Aut}(T)$ is only a map of sets, but since the composition of two 2-ordered maps is a 2-ordered map, and from the uniqueness of the extension, the map ψ is actually a group morphism. \square

Theorem 3.8. *The sequence*

$$(4) \quad 1 \longrightarrow \text{Aut}_e \longrightarrow \text{Aut}(T) \xrightarrow{\text{Res}} \text{Aut}(T_i) \longrightarrow 1,$$

is exact. Furthermore, the map Res has a section $\psi: \text{Aut}(T_i) \rightarrow \text{Aut}(T)$, hence $\text{Aut}(T) \simeq \text{Aut}_e \rtimes \text{Aut}(T_i)$.

Proof. The existence of the section ψ follows from Proposition 3.7, providing the surjectivity of the map Res. The last statement is a well known fact of split short exact sequences of groups. \square

Recall that two graphs G_1 and G_2 are isomorphic if and only if their adjacency matrices are conjugated by a permutation matrix. The advantage of working with bi-valent trees is that the sub-tree of inner vertices determines its isomorphism class uniquely (so instead of computing with matrices of size $(n-1)|V_i| + 2$ times $(n-1)|V_i| + 2$, we can work with matrices of size $|V_i| \times |V_i|$, where $\{1, n\}$ is the valuation set). In our applications (to construct subgroups of $\text{PSL}_2(\mathbb{Z})$) n will be 3, so will roughly speaking half the size of our matrices.

Corollary 3.9. *Two bi-valent trees T and T' are isomorphic if and only if their internal vertices sub-trees T_i and T'_i are isomorphic.*

Proof. Clearly, if $\phi: T \rightarrow T'$ is an isomorphism of graphs, its restriction to T_i gives an isomorphism between T_i and T'_i . Conversely, let ϕ be an isomorphism between T_i and T'_i . By Proposition 3.7, the map ϕ extends to a morphism between T and T' , hence the statement. \square

3.2. Algorithm to compute bi-valent trees. Let $m \geq 1$ and $n > 1$ be positive integers. Lemma 3.4 implies that any element of $\text{Bv}(m, n)$ has $m(n-2) + 2$ external vertices. Trees will be represented by their adjacency matrix, with the convention that on a tree with $m(n-2) + 2$ vertices, the first m labels are for the internal vertices and the remaining ones for the external vertices.

Definition 3.10. Let $T \in \text{Bv}(m, n)$ and let $v \in V_e(T)$. The *extension of T at v* , that will be denoted by $\text{Ext}(T, v)$, is the tree obtained by adding $n-1$ new vertices v_1, \dots, v_{n-1} to the set of vertices of T , and $n-1$ edges to the set of edges of T , joining v with v_i for each $i = 1, \dots, n-1$.

It is clear from its definition that if $T \in \text{Bv}(m, n)$ and $v \in V_e(T)$ then $\text{Ext}(T, v) \in \text{Bv}(m+1, n)$.

Lemma 3.11. *Let $T \in \text{Bv}(m, n)$ with $m > 1$. Then there exists $v \in V_e$ such that $|\text{Two}(v)| = n-1$.*

Proof. Recall the definition of the map $\Phi: V_e \rightarrow V_i$, which assigns to an external vertex v its adjacent internal vertex. If $v \in V_e$, then $|\text{Two}(v)| = |\Phi^{-1}(\Phi(v))|$. Clearly,

$$|V_e| = \sum_{v \in V_i} |\Phi^{-1}(v)|.$$

If all sets $\text{Two}(v)$ have cardinality smaller than $n-1$, then $|\Phi^{-1}(v)| \leq n-2$, so

$$|V_e| \leq (n-2)|V_i| < (n-2)|V_i| + 2 = |V_e|,$$

a contradiction. \square

Theorem 3.12. *The following algorithm gives a set of representatives of isomorphism classes of elements in $\text{Bv}(m, n)$ for $m \geq 1$, $n > 1$:*

1: **if** $m = 1$ **then**


```

2:   return  $\begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & 1 & 0 & \dots & 0 \end{pmatrix}$  (an  $(n+1) \times (n+1)$  matrix)
3: end if
4:  $S \leftarrow \emptyset$ 
5: for  $T \in \text{Bv}(m-1, n)$  do
6:   for  $v \in V_e(T)$  do
7:      $S \leftarrow S \cup \text{Ext}(T, v)$ 
8:   end for
9: end for
10: Remove isomorphic elements from  $S$ .
11: return  $S$ 

```

Proof. If $m = 1$, the tree has a unique internal vertex, hence n external ones (joined to the internal vertex). In particular, there is only one possible tree whose adjacency matrix is the given one.

Assume now that $m > 1$. If $T \in \text{Bv}(m, n)$, by Lemma 3.11, there exists a vertex $v \in V_e(T)$ such that $\text{Two}(v)$ has exactly $n-1$ elements. Let $w = \Phi(v)$ be the internal vertex adjacent to v . Let T' be the tree obtained from T by removing all vertices of $\text{Two}(v)$, as well as the edges joining w with elements of $\text{Two}(v)$. Then $T' \in \text{Bv}(m-1, n)$ and clearly T is isomorphic to $\text{Ext}(T', w)$. In particular, T is obtained from extending an element of $\text{Bv}(m-1, n)$ by an external vertex, so T is isomorphic to an element of S . \square

Actually, the previous algorithm can be improved since clearly if v, w are two external vertices of a bivalent tree T at distance 2 (or equivalently $w \in \text{Two}(v)$), then the extension of T by v is isomorphic to the extension of T by w . More generally:

Proposition 3.13. *Let $T \in \text{Bv}(m, n)$, let $v, w \in V_e$ and let $\varphi \in \text{Aut}(T)$ be such that $\varphi(v) = w$. Then φ induces an isomorphism $\tilde{\varphi}: \text{Ext}(T, v) \rightarrow \text{Ext}(T, w)$.*

Proof. Let $H = \text{Ext}(T, v)$ and $G = \text{Ext}(T, w)$. Recall that H is obtained by adding $n-1$ vertices to T , say v_1, \dots, v_{n-1} , and G is obtained by adding $n-1$ vertices to T , say w_1, \dots, w_{n-1} . Let $\tilde{\varphi}: H \rightarrow G$ be the map given by

$$\tilde{\varphi}(v) = \begin{cases} \varphi(v), & \text{if } v \in T, \\ w_i, & \text{if } v = v_i. \end{cases}$$

It can easily be verified that $\tilde{\varphi}$ is a well-defined isomorphism between the two trees. \square

Remark 3.14. In the particular case when $w \in \text{Two}(v)$, the previous map corresponds precisely to the *flip* isomorphism in Aut_e (the kernel of the restriction map) sending $v \leftrightarrow w$ and fixing all other elements.

Define an equivalence relation on V_e by determining that $v \sim w$ if there exists $\sigma \in \text{Aut}_e$ such that $\sigma(v) = w$. The notation $[V_e/\sim]$ will be used to denote any set of representatives for the equivalent classes. Then we have the following improvement of Theorem 3.12.

Theorem 3.15. *The following algorithm gives the elements in $\text{Bv}(m, n)$ for $m \geq 1$, $n > 1$, up to isomorphism:*

```

1: if  $m = 1$  then
2:   return  $\begin{pmatrix} 0 & 1 & \dots & 1 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & 1 & 0 & \dots & 0 \end{pmatrix}$  (an  $(n+1) \times (n+1)$  matrix)
3: end if
4:  $S \leftarrow \emptyset$ 
5: for  $T \in \text{Bv}(m-1, n)$  do
6:   for  $v \in [V_e(T)/\sim]$  do
7:      $S \leftarrow S \cup \text{Ext}(T, v)$ 
8:   end for
9: end for
10: Remove isomorphic elements from  $S$ .
11: return  $S$ 

```

The last step of the algorithm is done by checking whether two elements of S have isomorphic sub-trees of inner vertices or not.

3.3. Orientations on bi-valent trees. Recall that the set of vertices on a bi-valent tree is the union of the external vertices V_e (those with valence one) with the internal ones V_i (those having valence greater than one).

Definition 3.16. An *orientation* on a bi-valent tree $T \in \text{Bv}(m, n)$ is an orientation on its set of internal vertices, i.e. give for each vertex $v \in V_i$ an ordering of the set (of size n) consisting of vertices adjacent to v .

The way we represent an orientation algorithmically is by adding to each internal vertex $v \in V_i$ an n -cycle permutation σ_v . Thus a bi-valent tree with an orientation is a pair consisting of the adjacency matrix together with a list of n -cycles indexed by the internal vertices.

Example 3.17. Let T the unique tree (up to isomorphisms) in $\text{Bv}(2, 3)$ (from Example 3.3). Since T is already presented as a planar graph, an orientation is given by the anti-clockwise choice at each vertex (as mentioned in Remark 2.3), namely

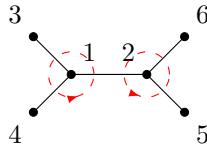


FIGURE 3. Unique oriented tree in $\text{Bv}(2, 3)$ except isomorphism.

We represent this orientation of T by the 3-cycles $\sigma_1 = (2, 3, 4)$ and $\sigma_2 = (1, 5, 6)$.

Definition 3.18. Let $T \in \text{Bv}(m, n)$ be a bi-valent tree with an orientation and let $p = \{v_1, \dots, v_n\}$ be a path between two external vertices. We say that the path p is *well oriented* if for each internal vertex v_i in the path p it holds that $\sigma_{v_i}(v_{i-1}) = v_{i+1}$ (i.e. the orientation on the vertex v sends the vertex of the path prior to v to the subsequent one).

Remark 3.19. If $n = 3$ (i.e. all vertices have valence 1 or 3, which is the case we are really interested in), a path is well oriented on a vertex v_i if and only if the 3-cycle σ_{v_i} equals (v_{i-1}, v_{i+1}, w) , where w is the third vertex adjacent to v_i .

Lemma 3.20. Let $T \in \text{Bv}(m, n)$ be a bi-valent tree with an orientation. Let $v \in V_e$ be an external vertex. Then there exists a unique $w \in V_e$ and a unique reduced and well oriented path p between v and w .

Proof. Let v_1 be the internal vertex adjacent to v , and let $v_2 = \sigma_{v_1}(v)$, the unique choice so that the path $\{v, v_1, v_2\}$ is well oriented. If v_2 is an external vertex, then the path $\{v, v_1, v_2\}$ is a well oriented path between external vertices. Otherwise, let $v_3 = \sigma_{v_2}(v_1)$ (once again this is the unique choice so that the path $\{v, v_1, v_2, v_3\}$ is well oriented) and continue this process. Since the number of vertices is finite, either at some point we reach an external vertex, or we get a “period”, i.e. there exist $i < j$ such that $v_i = v_j$. Since T is a tree, the only way to get a path starting and ending at the same vertex is that there exists an index t , with $i < t < j$ such that $v_{t-1} = v_{t+1}$, i.e. $v_{t+1} = \sigma_{v_t}(v_{t-1}) = v_{t-1}$, which cannot happen since the orientation at the vertex v_t is an n -cycle (so has no fixed points). \square

Definition 3.21. Let $T \in \text{Bv}(m, n)$ be an oriented bi-valent tree and let $v \in V_e$ be an external vertex. A vertex $w \in V_e$ is *to the right* of v if there exists a reduced and well oriented path p from v to w . Analogously, a vertex $w \in V_e$ is *to the left* of v if there exists a reduced and well oriented path from w to v .

In particular, we can define the function

$$(5) \quad r: V_e \rightarrow V_e, \quad r(v) = w \text{ (vertex to the right of } v\text{)}.$$

Respectively, we have a function $l: V_e \rightarrow V_e$ sending v to the vertex to the left of v . Recall our definition of the map $\Phi: V_e \rightarrow V_i$ which assigns to an external vertex the internal one adjacent to it.

Algorithm 3.22. Let $T \in \text{Bv}(m, n)$ a bi-valent tree with an orientation and v be an external vertex. The following algorithm computes the value of the function r at an external vertex v :

```

1:  $a \leftarrow v$ 
2:  $b \leftarrow \Phi(v)$ 
3:  $c \leftarrow \sigma_b(a)$ 
4: while  $c \in V_i$  do
5:    $a \leftarrow b$ 
6:    $b \leftarrow c$ 
7:    $c \leftarrow \sigma_b(a)$ 
8: end while
9: return  $c$ 

```

Lemma 3.23. Let $T \in \text{Bv}(m, n)$ be a bi-valent tree with an orientation, and let $v \in V_e$ be an external vertex. Then

$$V_e = \{r^i(v) : 0 \leq i < |V_e|\}.$$

Proof. Let $R(v) = \{v = v_1, v_2, \dots, v_N\}$ be the set of external vertices to the right of each other (so $r(v_N) = v$). Let p_i be the reduced and well oriented path between v_i and v_{i+1} for $i = 1, \dots, N-1$ and p_N the reduced and well oriented path between v_N and v_1 . Each path is made of edges, so let \tilde{E} be the union of all the (directed) edges appearing in p_i for any $1 \leq i \leq N$ and let \tilde{V} be the set of vertices of elements of \tilde{E} . Suppose we prove that for each $v \in \tilde{V}$ which is an internal vertex of V it holds that its n adjacent elements are also in \tilde{V} , then it must be the case that $\tilde{V} = V$. The reason is that given any $w \in V$, there is a unique reduced path joining v to w . By our assumption, all edges of the path are elements of \tilde{E} , so $w \in \tilde{V}$ as claimed.

To prove the stated property, note that since $r(v_N) = v$, the compositum $p_N \circ \dots \circ p_1$ is a path between v_1 and v_1 , hence is the trivial path, i.e. in this walk there is a complete cancellation of edges. In particular, if a directed edge lies in \tilde{E} , the edge with the opposite direction must also be a member of \tilde{E} .

Let $b \in \tilde{V}$ be an internal vertex which is part of a path p_i , say $\{a, b, c\}$ is part of p_i . Since paths are well oriented, $\sigma_b(a) = c$. The complete cancellation property implies that the (directed) edge (b, a) is part of some path p_j (for $1 \leq j \leq N$), i.e. $\{\sigma_b^{-1}(a), b, a\}$ appears in the path p_j (because p_j is also well oriented). This proves that if $(a, b) \in \tilde{E}$ then $(\sigma_b^{-1}(a), b)$ also belong to \tilde{E} . Repeating the argument, $(\sigma_b^{-i}(a), b) \in \tilde{E}$ for all $1 \leq i \leq n$, and since σ_b is an n -cycle, all these edges are different hence all the vertices adjacent to b are in \tilde{V} . \square

Remark 3.24. As mentioned before, we are mainly interested in the case $n = 3$. The way we compute representatives for elements in $\text{Bv}(m, 3)$ with an orientation is the following: start with a set of isomorphism classes representatives S for $\text{Bv}(m, 3)$ as described in Theorem 3.15. Given an element $T \in S$, compute for each internal vertex the two possible orientations, and store the resulting bi-valent trees with an orientation. This produces a set whose number of elements is $2^{|V_i|}$ times the size of S .

One can do better by the following observation: if v is an internal vertex adjacent to two external vertices w_1, w_2 , then the two possible orientations at the vertex v are isomorphic (via the isomorphism sending $w_1 \leftrightarrow w_2$ and fixing the other vertices). For this reason, our code just picks one orientation for each internal vertex adjacent to two exterior ones. This trivial improvement gives in practice a huge saving.

3.4. Coloring external vertices. Recall from Definition 2.5 that a tree diagram is an element \mathfrak{T} of $\text{Bv}(m, 3)$ (for some m) with an orientation and a particular coloring on the set V_e (the external vertices of \mathfrak{T}). Let b, r, f be non-negative integers such that $b + r + 2f = m + 2 = |V_e|$. A coloring on V_e with parameters (b, r, f) is equivalent to a decomposition of the set V_e as a disjoint union of the form

$$V_e = B \cup R_0 \cup_i F_i,$$

(as in (2)) where

- (1) the set B has b vertices; its vertices are the blue ones,
- (2) the set R_0 has r vertices; its vertices are the red ones fixed by the involution ι , and
- (3) each set F_i has two red vertices, and the involution ι sends one to the other. The union of the sets F_i equals R_1 in (2). It is a set of size $2f$ corresponding to the “free sides”.

Given $T \in \text{Bv}(m, 3)$ with an orientation, the way to compute all possible coloring with parameters (b, r, f) is first to compute all possible subsets B of V_e of size b , and for each choice, compute all possible choices of the subset R_0 of r elements on its complement. The complement $S = V_e \setminus (B \cup R_0)$ has then $2f$ -elements. A way of decomposing the set S as a disjoint union of f subsets of size 2 will be called a *2-partition*.

Problem 3.25. *Given a set S with $2f$ elements, how to compute its 2-partitions?*

Definition 3.26. Let n be a positive integer. A $2n$ -tuple (v_1, \dots, v_{2n}) of integers is *ordered* if it satisfies the following two properties:

- The sub-tuple $(v_1, v_3, \dots, v_{2n-1})$ made of the odd entries is ordered.
- For any odd index i , $1 \leq i \leq 2n-1$, $v_i < v_{i+1}$.

Similarly, we say that an element of \mathbb{S}_{2f} is ordered if so is the $2f$ -tuple it represents.

Let $S = \{1, \dots, 2f\}$ and let P denote the set of ordered $2f$ -tuples made of elements of S . In particular, the coordinates of elements of S are all different. Let $\Psi: P \rightarrow \{2\text{-partitions}\}$ be the map given by

$$(6) \quad \Psi(v_1, \dots, v_{2f}) = \{v_1, v_2\} \cup \dots \cup \{v_{2f-1}, v_{2f}\}.$$

Lemma 3.27. *With the previous notation, the map Ψ is bijective.*

Proof. Injectivity: let (v_1, \dots, v_{2f}) and (w_1, \dots, w_{2f}) be elements of P such that

$$\Psi(v_1, \dots, v_{2f}) = \Psi(w_1, \dots, w_{2f}).$$

Then $\{\{v_1, v_2\}, \dots, \{v_{2f-1}, v_{2f}\}\} = \{\{w_1, w_2\}, \dots, \{w_{2f-1}, w_{2f}\}\}$, say $\{v_1, v_2\} = \{w_i, w_{i+1}\}$ for some odd index i . The second condition of an ordered tuple implies that $v_1 < v_2$ and $w_i < w_{i+1}$ so $v_1 = w_i$ and $v_2 = w_{i+1}$. Repeating this argument it follows that the set of odd entries of the first tuple must equal the set of odd entries of the second one, and since both sets are ordered, the tuples must be the same.

Surjectivity: let $S = \bigcup_{i=1}^f \{v_{2i-1}, v_{2i}\}$ be a 2-partition. Clearly we can order each set so that $v_{2i-1} < v_{2i}$ for all values of i , and we can also order the sets so that $v_{2i-1} < v_{2i+1}$ for all $i = 1, \dots, f-1$. By definition, the tuple (v_1, \dots, v_{2f}) is ordered (so is an element of P). \square

Proposition 3.28. *Let $T \in \text{Bv}(m, 3)$ be a bi-valent tree with an orientation, and let (b, r, f) be non-negative integers satisfying that $b + r + 2f = m + 2$. The following algorithm computes all possible coloring on V_e with parameters (b, r, f) :*

```

1:  $C \leftarrow \emptyset$ 
2: for  $B \subseteq V_e, |B| = b$  do
3:    $R = V_e \setminus B$ 
4:   for  $R_0 \subseteq R, |R_0| = r$  do
5:      $R_1 = R \setminus R_0$ 
6:     for  $\sigma \in \mathbb{S}_{R_1}$  do
7:       if  $\sigma$  is ordered then
8:          $C \leftarrow C \cup (B, R_0, \Psi(\sigma))$ 
9:       end if
10:    end for
11:  end for
12: end for
13: return  $C$ 

```

Let \mathfrak{T} be a tree diagram coming from a subgroup Δ of $\text{PSL}_2(\mathbb{Z})$. Then

$$(7) \quad [\text{PSL}_2(\mathbb{Z}) : \Delta] = 3m + b,$$

where m is the number of internal vertices of \mathfrak{T} and b is the number of blue external vertices (see for example the proposition in page 1078 of [9]). This implies that there are finitely many triples (b, r, f) corresponding to index d -subgroups of $\text{PSL}_2(\mathbb{Z})$.

Proposition 3.29. *Let d be positive integer. The following algorithm computes all possible triples (b, r, f) attached to index d subgroups of $\text{PSL}_2(\mathbb{Z})$:*

```

1:  $M \leftarrow \emptyset$ 

```

```

2: for  $\lceil \frac{d-2}{4} \rceil \leq m \leq \lfloor \frac{d}{3} \rfloor$  do
3:    $b \leftarrow d - 3m$ 
4:   for  $j = 0, \dots, \lfloor \frac{m-b}{2} \rfloor + 1$  do
5:      $M \leftarrow M \cup \{(b, m+2-b-2j, j)\}$ 
6:   end for
7: end for
8: return  $M$ 

```

Proof. Let Δ be a subgroup of $\text{PSL}_2(\mathbb{Z})$ of index d , and let $m+2$ be the number of cusps belonging to a special polygon. The formula $d = 3m + b$ together with the fact that the number of odd vertices cannot exceed the number of cusps, give the restriction

$$3m \leq d \leq 4m + 2.$$

This completes the proof. \square

Combining the different algorithms described in the present section, we can compute given a positive integer d the set of non-equivalent tree diagrams corresponding to subgroups of index d in $\text{PSL}_2(\mathbb{Z})$.

4. TREE DIAGRAMS, GENERALIZED FAREY SYMBOLS AND SUBGROUPS

As explained in the introduction, our tree diagram is obtained as the tree of f -edges of a special polygon. It is a natural question how to recover information of the special polygon that is missing in the tree diagram. For example, how can we compute its set of cusps? (note that since the special polygon attached to a subgroup is not unique, the answer will also not be unique).

As explained in [9], the cusps of a special polygon form what is called a *generalized Farey symbol*, since if $\frac{a}{b}$ and $\frac{c}{d}$ are two consecutive cusps of a special polygon then $|ad - bc| = 1$.

Definition 4.1. A *generalized Farey symbol* (g.F.s. for short) is an expression of the form

$$\{-\infty, c_2, c_3, \dots, c_n, \infty\}$$

where

- (1) c_2 and c_n are integers, and some $c_i = 0$,
- (2) $c_i = \frac{a_i}{b_i}$ are rational numbers in their reduced forms and ordered according to their magnitudes, such that

$$|a_i b_{i+1} - b_i a_{i+1}| = 1, \quad i = 2, 3, \dots, n-1.$$

Remark 4.2. Our definition is a little different from Kulkarni's one. The elements $-\infty$ and ∞ are identified as points on the polygon, so they correspond to the same point at infinity, but for algorithmic purposes it is better to represent them as different elements. In particular, we will represent $-\infty$ by the fraction $\frac{-1}{0}$ and ∞ by $\frac{1}{0}$. This is consistent with the GAP package developed in [5] for computing with generalized Farey symbols.

Any path between two cusps (members of a g.F.s.) will necessarily go through an end point of a tree diagram. As suggested in [9], one can add the information of the vertex color to the g.F.s. forming what is called a *Farey symbol*.

Definition 4.3. A *Farey symbol* is an expression of the form:

$$\underbrace{\{-\infty \quad c_2 \quad c_3 \quad c_4 \quad \dots \quad c_n \quad \infty\}}_{\begin{matrix} p_1 & p_2 & p_3 & p_4 & \dots & p_n \end{matrix}}$$

where

- $\{-\infty, c_2, \dots, c_n, \infty\}$ is a g.F.s.,
- each symbol p_i is one of \circ , \bullet or a number label, depending on whether the line on the boundary that joins c_i with c_{i+1} is even, odd or a free side (in which case the same label is used for its matching line).

Definition 4.4. Let \mathfrak{T} be a tree diagram, and let v, w be two external vertices next to each other (i.e. one is to the left or right of the other). Define the *bipartite distance* between v and w (denoted by $\text{bd}(v, w)$) as the distance between v and w in the tree obtained from \mathfrak{T} while constructing the bipartite cuboid graph before identifying the glued vertices.

Lemma 4.5. Let \mathfrak{T} be a tree diagram, and let v, w be external vertices next to each other. Let d denote the distance between v and w on the tree \mathfrak{T} . Then

$$(8) \quad \text{bd}(v, w) = 2d - 2 + \begin{cases} 0, & \text{if } v \text{ and } w \text{ are both even (red),} \\ 2, & \text{if } v \text{ and } w \text{ are both odd (blue),} \\ 1, & \text{otherwise.} \end{cases}$$

Proof. Since \mathfrak{T} is a tree, there is a unique reduced path between v and w , and since the distance between v and w is d , the path goes through $d - 1$ internal vertices. The bipartite graph is obtained by adding an extra (red) vertex between each pair of internal vertices, giving an extra $d - 2$ steps. At last, if either ending point is odd (i.e. blue), we need to add an extra vertex next to it. \square

Proposition 4.6. Let \mathfrak{T} be a tree diagram, and let v, w be external vertices next to each other. Let $x = \frac{a}{b}$, y and $z = \frac{c}{d}$ be three consecutive cusps of a special polygon attached to \mathfrak{T} satisfying that v is between x and y and that w is between y and z . Then

$$(9) \quad \text{bd}(v, w) = 2|ad - bc| + \begin{cases} 0, & \text{if } v \text{ and } w \text{ are both even (red),} \\ 2, & \text{if } v \text{ and } w \text{ are both odd (blue),} \\ 1, & \text{otherwise.} \end{cases}$$

Proof. Since everything is invariant under the action of $\text{SL}_2(\mathbb{Z})$, we can assume that y is the infinity cusp. Then x and z are both integers (i.e. $b = d = 1$) with $c < a$. The tessellation of the upper part of our special polygon then is one of the following:

- If v and w are both even (red) then it looks like Figure 4A. Note that

$$2(a - c) = \text{bd}(v, w)$$

(since there are two f -edges between consecutive red vertical lines).

- If v and w are both odd (blue) then the tessellation looks like Figure 4B. Note that $2(a - c) + 2 = \text{bd}(v, w)$ (since there are two f -edges between consecutive blue vertical lines).
- If v and w have different colors (say v is blue and w is red) then the tessellation looks like Figure 4C. Note that $2(a - c) + 1 = \text{bd}(v, w)$.

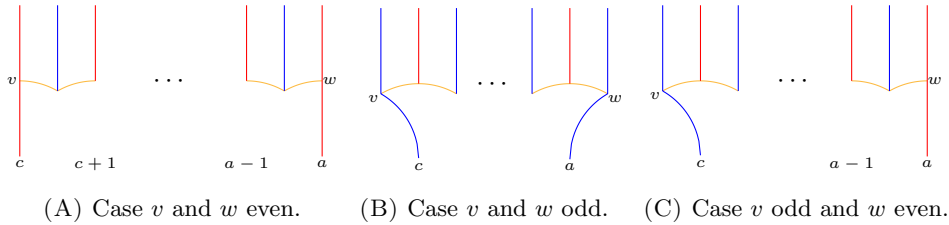


FIGURE 4. Relation between the distance and the crossed product.

\square

From equations (8) and (9) it follows that if $x = \frac{a}{b}$, y and $z = \frac{c}{d}$ are three consecutive cusps with v between x and y and w between y and z , then

$$(10) \quad |ad - bc| = d(v, w) - 1.$$

These relations determine the g.F.s. uniquely (up to $\text{SL}_2(\mathbb{Z})$ -equivalence).

Theorem 4.7. Let \mathfrak{T} be a tree diagram, and $V_e = \{v_1, \dots, v_m\}$ denote the set of external colored vertices. Assume that the set V_e is ordered, i.e. v_{i+1} is to the right of v_i for $1 \leq i \leq m$ (with the convention that $v_{m+1} = v_1$). The following algorithm computes a g.F.s. attached to \mathfrak{T} :

```

1:  $F \leftarrow \{\frac{-1}{0}, \frac{0}{1}\}.$ 
2: if  $m = 2$  then
3:   return  $F \cup \{\frac{1}{0}\}.$ 
4: else
5:    $d \leftarrow d(v_1, v_2).$ 
6:    $F \leftarrow F \cup \{\frac{1}{d-1}\}.$ 
7:   for  $i = 2, \dots, m-2$  do
8:      $d \leftarrow d(v_i, v_{i+1}).$ 
9:      $\alpha \leftarrow \text{Num}(F[i+1]).$ 
10:     $\beta \leftarrow \text{Den}(F[i+1]).$ 
11:     $\gamma \leftarrow \text{Num}(F[i]).$ 
12:     $\delta \leftarrow \text{Den}(F[i]).$ 
13:     $\begin{pmatrix} y \\ x \end{pmatrix} \leftarrow \begin{pmatrix} \alpha & -\beta \\ \gamma & -\delta \end{pmatrix}^{-1} \begin{pmatrix} 1-d \\ -1 \end{pmatrix}.$ 
14:     $F \leftarrow F \cup \{\frac{x}{y}\}.$ 
15:   end for
16: end if
17: return  $F \cup \{\frac{1}{0}\}.$ 

```

Proof. Up to $\text{SL}_2(\mathbb{Z})$ -equivalence, we can always assume that the first cusp (to the left of v_1) equals ∞ and the next one equals 0. Since the number of cusps equals the number of external vertices, if there are only two such cusps, we are done. Suppose that there are at least three cusps, say $\{\frac{-1}{0}, \frac{0}{1}, \frac{a}{b}\}$, for some positive integers a, b . The integers a, b satisfy the relations:

$$\begin{cases} |-b + 0 \cdot a| &= d(v_1, v_2) - 1, \\ 0 \cdot b - a &= -1, \end{cases}$$

where the second condition comes from (10). Then $a = 1$ and $b = d(v_1, v_2) - 1$. If we constructed the first elements of our g.F.s., say $\{\frac{-1}{0}, \frac{0}{1}, x_1, \dots, x_i\}$, with $x_{i-1} = \frac{\alpha}{\beta}$ and $x_i = \frac{\gamma}{\delta}$, the next element $x_{i+1} = \frac{X}{Y}$ is a solution of the system

$$\begin{cases} |\alpha Y - \beta X| &= d(v_i, v_{i-1}) - 1, \\ \gamma Y - \delta X &= -1. \end{cases}$$

The fact that $x_{i+1} > x_{i-1}$ together with the fact that both the numerators and denominators of x_{i-1} and x_{i+1} are positive integers (because $i \geq 2$) imply that $|\alpha Y - \beta X| = -\alpha Y + \beta X$. \square

Definition 4.8. A *Kulkarni diagram* is an object consisting of a tree diagram, together with a choice of a g.F.s. (denoted by G) attached to it.

The particular choice of the g.F.s. is not important, a different choice will correspond to another subgroup of $\text{PSL}_2(\mathbb{Z})$ which is conjugate (by a matrix in $\text{SL}_2(\mathbb{Z})$) to it. The elements of G will be called *cusps*.

4.1. Some important algorithms. The purpose of the present section is to gather together some algorithms to compute with Kulkarni diagrams as well as with its attached subgroup Δ . The most important algorithms are that of computing generators for Δ , and an algorithm to determine whether an element $g \in \text{PSL}_2(\mathbb{Z})$ belongs to Δ or not (crucial to compute the passport attached to a Kulkarni diagram).

To make statements more clear, we will use the following notation: \mathcal{K} denotes a Kulkarni diagram, $G = \{-\infty, c_2, \dots, c_{m+2}, \infty\}$ denotes its g.F.s. and Δ the subgroup attached to it. As proven in [9, Theorem 6.1], the group Δ has a set of generators $\{\alpha_1, \dots, \alpha_{m+2}\}$ indexed by the elements of G , representing how the paths in the boundary of a special polygon are glued together.

Algorithm 4.9 (Generator of a cusp). *Let \mathcal{K} be a kulkarni diagram, $V_e = \{v_1, \dots, v_{m+2}\}$ be its set of (external) colored and ordered vertices and let G be its underlying g.F.s. Given a cusp $c_k \in G$, the following algorithm computes the generator corresponding to the cusp c_k :*

```

1:  $a \leftarrow \text{Num}(c_{k+1}).$ 
2:  $b \leftarrow \text{Den}(c_{k+1}).$ 
3:  $c \leftarrow \text{Num}(c_k).$ 

```



```

4:  $d \leftarrow \text{Den}(c_k)$ .
5: if  $v_k$  is “even” then
6:    $g \leftarrow \begin{pmatrix} ab+cd & -c^2-a^2 \\ b^2+d^2 & -ab-cd \end{pmatrix}$ .
7: else if  $v_k$  is “odd” then
8:    $g \leftarrow \begin{pmatrix} ab+cb+cd & -c^2-ac-a^2 \\ b^2+bd+d^2 & -ab-ad-cd \end{pmatrix}$ ;
9: else
10:   $v_k$  is identified with some  $v_j$ 
11:   $a' \leftarrow \text{Num}(c_{j+1})$ .
12:   $b' \leftarrow \text{Den}(c_{j+1})$ .
13:   $c' \leftarrow \text{Num}(c_j)$ .
14:   $d' \leftarrow \text{Den}(c_j)$ .
15:   $g \leftarrow \begin{pmatrix} a'b+c'd & -c'c-a'a \\ d'd+b'b & -ab'-cd' \end{pmatrix}$ .
16: end if
17: return  $g$ .

```

Proof. See [9, Theorem 6.1]. □

Algorithm 4.10 (Cusps representatives). *Let \mathcal{K} be a Kulkarni diagram and let $V_e = \{v_1, \dots, v_{m+2}\}$ be its set of (external) colored and ordered vertices. The following algorithm computes the subgroup of \mathbb{S}_{m+2} whose orbits correspond to equivalent cusps:*

```

1:  $S \leftarrow \emptyset$ .
2: for  $v_k \in V_e$  do
3:   if  $v_k$  is “even” or “odd” then
4:      $S \leftarrow S \cup \{(k, k+1)\}$ .
5:   else
6:      $v_k$  is identified with some  $v_j$ 
7:      $S \leftarrow S \cup \{(k, j+1), (k+1, j)\}$ 
8:   end if
9: end for
10: return  $\langle S \rangle$ .

```

Proof. Clear from the gluing of the border paths. □

Let C_1, \dots, C_t be the cusp representatives of the Kulkarni diagram \mathcal{K} , i.e. each set C_i is made of elements of G which are in the i -th orbit under the action of the group S computed using Algorithm 4.10.

Problem 4.11. *How to compute $W(C_i)$, the width of C_i ?*

As before, let $G = \{c_1 = -\infty, c_2, c_3, \dots, c_{m+2}, c_{m+3} = \infty\}$. Let $d: G \rightarrow \mathbb{Z}$ be the function

$$(11) \quad d(c_i) = |a_{i-1}b_{i+1} - a_{i+1}b_{i-1}|,$$

where a_i (resp. b_i) is the numerator (resp. denominator) of the cusp c_i , and the indices are taken “in a cyclic order” (i.e. $c_{m+3} = c_1$).

Algorithm 4.12 (Width of a cusp). *Let \mathcal{K} be a Kulkarni diagram, $V_e = \{v_1, \dots, v_{m+2}\}$ be its colored and ordered external vertices and let c_{i_1}, \dots, c_{i_r} be the cusps belonging to the class C . The following algorithm computes the width of C :*

```

1:  $W \leftarrow 0$ 
2: for  $j = 1, \dots, r$  do
3:    $e \leftarrow 0$ 
4:   if  $v_{i_{j-1}}$  is “odd” then
5:      $e \leftarrow e + 1$ 
6:   end if
7:   if  $v_{i_j}$  is “odd” then
8:      $e \leftarrow e + 1$ 
9:   end if

```

```

10:  $W \leftarrow W + d(c_{i_j}) + e/2$ 
11: end for
12: return  $W$ 

```

Proof. See the proposition of page 1079 in [9]. □

4.2. The word problem. The classical solution to determine whether an element $g \in \text{PSL}_2(\mathbb{Z})$ belongs to Δ (the subgroup attached to the Kulkarni diagram \mathcal{K}) is to produce a “reduction algorithm” modulo elements of Δ . Based on the article [11], in [10] (Algorithm, page 13) the authors give such an algorithm. The problem with the stated method is that it has some “border” conditions which are not clearly stated, so we take the opportunity to add to it the missing details. If a and b are integers, we denote by $\text{quo}(a, b)$ the cusp $\frac{a}{b}$, with the usual convention that $-\infty = \frac{-1}{0}$ and $\infty = \frac{1}{0}$.

Theorem 4.13 (Reduction). *Let \mathcal{K} be a Kulkarni diagram, let Δ be its attached group, G be its $g.F.s.$ and let C denote the vector consisting of the coloring on G . Let $g \in \text{PSL}_2(\mathbb{Z})$ be any element. The following algorithm (Reduction) gives a reduced element in the coset Δg :*

```

1:  $N \leftarrow |G|$ 
2: if  $g[2, 1] = 0$  then
3:    $\text{case} \leftarrow 1$ 
4: else if  $g[2, 2] = 0$  then
5:    $\text{case} \leftarrow 2$ 
6: else
7:    $\text{case} \leftarrow \text{“generic”}$ 
8: end if
9: if  $\text{case} = \text{“generic”}$  then
10:  if  $g[1, 2]/g[2, 2] > g[1, 1]/g[2, 1]$  then
11:     $x \leftarrow \text{Reduction} \left( \begin{pmatrix} -g[1, 2] & g[1, 1] \\ -g[2, 2] & g[2, 1] \end{pmatrix} \right)$ 
12:    return  $\begin{pmatrix} -x[1, 2] & x[1, 1] \\ -x[2, 2] & x[2, 1] \end{pmatrix}$ 
13:  end if
14: else if  $\text{case} = 1$  then
15:  if  $g[1, 1] > 0$  then
16:    if  $g[1, 2]/g[2, 2] < 0$  then
17:       $x \leftarrow \text{Reduction} \begin{pmatrix} g[1, 2] & -g[1, 1] \\ g[2, 2] & -g[2, 1] \end{pmatrix}$ 
18:      return  $\begin{pmatrix} -x[1, 2] & x[1, 1] \\ -x[2, 2] & x[2, 1] \end{pmatrix}$ 
19:    end if
20:  else
21:    if  $g[1, 2]/g[2, 2] > G[N - 1]$  then
22:       $g \leftarrow -g$ 
23:    else
24:       $x \leftarrow \text{Reduction} \begin{pmatrix} -g[1, 2] & g[1, 1] \\ -g[2, 2] & g[2, 1] \end{pmatrix}$ 
25:      return  $\begin{pmatrix} -x[1, 2] & x[1, 1] \\ -x[2, 2] & x[2, 1] \end{pmatrix}$ 
26:    end if
27:  end if
28: else if  $\text{case} = 2$  then
29:  if  $g[1, 2] > 0$  then
30:    if  $g[1, 1]/g[2, 1] > 0$  then
31:       $x \leftarrow \text{Reduction} \begin{pmatrix} -g[1, 2] & g[1, 1] \\ -g[2, 2] & g[2, 1] \end{pmatrix}$ 
32:      return  $\begin{pmatrix} -x[1, 2] & x[1, 1] \\ -x[2, 2] & x[2, 1] \end{pmatrix}$ 
33:    else
34:       $g \leftarrow -g$ 
35:    end if

```

```

36:  else
37:    if  $g[1, 1]/g[2, 1] > G[N - 1]$  then
38:       $x \leftarrow \text{Reduction} \begin{pmatrix} g[1, 2] & -g[1, 1] \\ g[2, 2] & -g[2, 1] \end{pmatrix}$ 
39:      return  $\begin{pmatrix} -x[1, 2] & x[1, 1] \\ -x[2, 2] & x[2, 1] \end{pmatrix}$ 
40:    end if
41:  end if
42: end if
43:  $a \leftarrow \text{quo}(g[1, 2], g[2, 2])$ 
44:  $b \leftarrow \text{quo}(g[1, 1], g[2, 1])$ 
45: if  $a \in G$  and  $b \in G$  then
46:   return  $g$ 
47: else
48:   if  $\text{case} = 1$  then
49:      $c \leftarrow N$ 
50:   else if  $\text{case} = 2$  then
51:      $c \leftarrow 2$ 
52:   else if  $b \in G$  then
53:      $c \leftarrow \text{Position}(G, b)$ 
54:   else
55:      $l \leftarrow G \cup \{b\}$ 
56:      $\text{Sort}(l)$ 
57:      $c \leftarrow \text{Position}(l, b)$ 
58:   end if
59: end if
60: if not  $C[c - 1] = \text{"odd"}$  then
61:    $y \leftarrow \text{CuspGenerators}(G[c - 1])$ 
62: else
63:    $m \leftarrow (\text{Num}(G[c - 1]) + \text{Num}(G[c])) / (\text{Den}(G[c - 1]) + \text{Den}(G[c]))$ 
64:   if  $g[2, 1] = 0$  or  $b > m$  then
65:      $y \leftarrow \text{CuspGenerator}(G[c - 1])$ 
66:   else
67:      $y \leftarrow \text{CuspGenerators}(G[c - 1])^{-1}$ 
68:   end if
69: end if
70: return  $\text{Reduction}(y * g)$ 

```

Proof. Let $g = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, and let $\alpha = \text{quo}(a, c)$, $\beta = \text{quo}(b, d)$ be the cusps corresponding to the first and the second column. The boundary issues appear when either α or β are the infinity cusp (depending also on whether they correspond to the $-\infty$ or the ∞ cusp). The first case corresponds to $\alpha = \infty$, the second case to $\beta = \infty$ and the remaining ones to the “generic” case. In [10] the authors start assuming that $\beta < \alpha$ (following their main reference [11]). If we multiply the matrix g on the right by $S = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ has the effect of interchanging $\alpha \leftrightarrow \beta$, so in case $\alpha < \beta$ we reduce the matrix gS and multiply the reduced matrix by S to the right. Multiplication by S sends

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \rightarrow \begin{pmatrix} -b & a \\ -d & c \end{pmatrix}.$$

This justifies the steps 10 – 12 of the algorithm. In the non-generic cases, this idea might not be enough to get the “right” reduction step (as explained in [11], the idea behind the algorithm is to shorten a distance). Suppose that $|\alpha| = \infty$ (so we are in the first case), then the border condition is the following:

- If $\alpha = \infty$ and β is positive, we reduce g .
- If $\alpha = \infty$ and β is negative, we reduce $-gS$ (i.e. instead of considering the pair (∞, β) we work with $(\beta, -\infty)$).
- If $\alpha = -\infty$ and β is not larger than all elements of G , we reduce gS (i.e. $(\beta, -\infty)$).

- If $\alpha = -\infty$ and β is larger than all elements of G , we reduce $-g$ (i.e. (∞, β)).

Similarly, when $|\beta| = \infty$ (the second case), the border condition is the following:

- If $\beta = -\infty$ and α is no larger than all elements of G , we reduce g .
- If $\beta = -\infty$ and α is larger than all elements of G , we reduce $-gS$ (i.e. we replace the pair $(\alpha, -\infty)$ by (∞, α)).
- If $\beta = \infty$ and $\alpha < 0$, we reduce $-g$ (replacing (α, ∞) by $(\alpha, -\infty)$).
- If $\beta = \infty$ and $\alpha > 0$, we reduce gS (replacing (α, ∞) by (∞, α)).

In all the above cases, we end up in a situation where $\beta < \alpha$. The rest of the algorithm mimics the one presented in [10]. \square

Let $g \in \text{PSL}_2(\mathbb{Z})$ and let $g' = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ be its reduction (i.e. the output of the last algorithm).

Theorem 4.14. *With the previous notation, an element $g \in \text{PSL}_2(\mathbb{Z})$ belongs to Δ if and only if one of the following is true:*

- (1) $g' = \pm \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$,
- (2) $\begin{pmatrix} b & a \\ d & c \end{pmatrix}$ is a free side paired with $(0, \infty)$,
- (3) $g' = \pm \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, and 0 and ∞ are adjacent vertices with an even pairing between them.

Proof. See [11, Theorem 3.2]. \square

The last two results provide a solution to the word problem, namely determining whether an element $\alpha \in \text{PSL}_2(\mathbb{Z})$ belongs to Δ or not. We end this section with an algorithm to compute coset representatives for $\Delta \backslash \text{PSL}_2(\mathbb{Z})$.

Proposition 4.15 (Coset representatives). *Let \mathcal{K} be a Kulkarni diagram, let Δ be the attached group and G be its g.f.s. Let C be the vector made of the coloring of G . The following algorithm gives a complete set of representatives for $\Delta \backslash \text{PSL}_2(\mathbb{Z})$:*

```

1:  $N \leftarrow |G|$ 
2:  $T \leftarrow \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ 
3:  $\text{reps} \leftarrow \emptyset$ 
4: for  $i = 1, \dots, N - 1$  do
5:   if  $i = 1$  then
6:      $w \leftarrow |\text{Num}(G[N - 1]) * \text{Den}(G[2]) - \text{Den}(G[N - 1]) * \text{Num}(G[2])|$ 
7:   else
8:      $w \leftarrow |\text{Num}(G[i - 1]) * \text{Den}(G[i + 1]) - \text{Den}(G[i - 1]) * \text{Num}(G[i + 1])|$ 
9:   end if
10:   $A \leftarrow \begin{pmatrix} -\text{Num}(G[i]) & \text{Num}(G[i + 1]) \\ -\text{Den}(G[i]) & \text{Den}(G[i + 1]) \end{pmatrix}$ 
11:  if  $C[i] = \text{"odd"}$  then
12:     $\text{reps} \leftarrow \text{reps} \cup \{A * T^{-1}\}$ 
13:  end if
14:  for  $j \in [0, \dots, w - 1]$  do
15:     $\text{reps} \leftarrow \text{reps} \cup \{A * T^j\}$ 
16:  end for
17: end for
18: return  $\text{reps}$ 

```

Proof. The result is stated in §5.3 of [10], although there is a minor mistake on it (which is why we present a proof). Let \mathcal{F} be the hyperbolic triangle with vertices ρ, i, ∞ as in Figure 1 and let P be the special polygon attached to \mathcal{T} with cusps the g.f.s. of \mathcal{K} . If $c_i = \frac{a_i}{b_i}$ and $c_{i+1} = \frac{a_{i+1}}{b_{i+1}}$ are two consecutive vertices of the special polygon attached to \mathcal{K} (so they are two consecutive cusps), then the matrix

$$A = \begin{pmatrix} -a_i & a_{i+1} \\ -b_i & b_{i+1} \end{pmatrix}$$

sends the line joining ∞ and 0 to the line joining c_i to c_{i+1} (note that the sign in the first column is missing in [10]). The image under A^{-1} of P then contains the cusp $A^{-1} \cdot c_{i-1}$ (a positive integer) next to ∞ next to

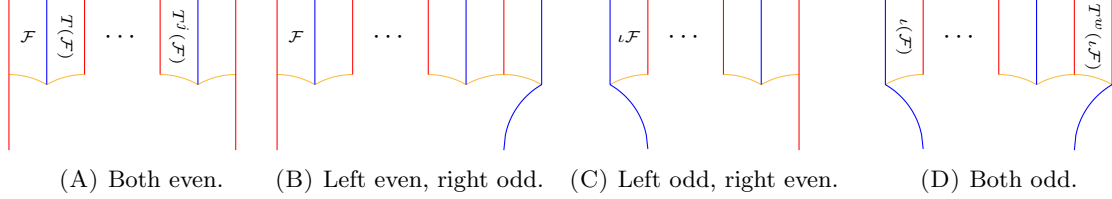


FIGURE 5. Representatives near the infinity cusp.

the cusp 0. If $c_{i-1} = \frac{a_{i-1}}{b_{i-1}}$, then

$$A^{-1} \cdot c_{i-1} = w = |a_{i-1}b_{i+1} - b_{i-1}a_{i+1}|.$$

Suppose that the line joining c_i and c_{i+1} is even, then the translates of \mathcal{F} appearing in $A^{-1}P$ are either of the form

- $T^j(\mathcal{F} \cup T\iota(\mathcal{F}))$ if the line joining c_i to c_{i-1} is also even (so they are translates of a fundamental domain for $\text{PSL}_2(\mathbb{Z})$), where $0 \leq j \leq w-1$ (see Figure 5A),
- $T^j(\mathcal{F} \cup T\iota(\mathcal{F}))$, where $0 \leq j \leq w-1$ together with $T^w\mathcal{F}$ (see Figure 5B) if the line joining c_i to c_{i-1} is odd. Note that the “other” part of the fundamental domain will appear while considering the cusp c_{i-1} , so we do not add it to the list to avoid repetitions.

In this case it is clear that the hyperbolic triangles in the special polygon P containing the infinity point are the translates by $A * T^j$ of \mathcal{F} (together with $T\iota(\mathcal{F})$) for $0 \leq j \leq w-1$.

Suppose otherwise that the line joining c_i and c_{i+1} is odd. Then the translates of \mathcal{F} appearing in $A^{-1}P$ belong to one of the following two cases:

- $T^{-1}(T\iota(\mathcal{F}))$ together with $T^j(\mathcal{F} \cup T\iota(\mathcal{F}))$, where $0 \leq j \leq w-1$ (see Figure 5C) if the line joining c_i to c_{i-1} is even. Note that we must add the first element as it was excluded from the opposite situation (namely left side even and right one odd).
- $T^{-1}(T\iota(\mathcal{F}))$ together with $T^w\mathcal{F}$ and $T^j(\mathcal{F} \cup T\iota(\mathcal{F}))$ for $0 \leq j \leq w-1$ (see Figure 5D). Once again, to avoid repetitions, we do not add the representative $T^w\mathcal{F}$.

In this second case the hyperbolic triangles in the special polygon P containing the infinity point are the translates of $A * T^j$ for $-1 \leq j \leq w-1$ of \mathcal{F} (together with $T\iota(\mathcal{F})$). \square

4.3. Relation with geometry. From a finite index subgroup Δ of $\text{PSL}_2(\mathbb{Z})$ one can construct the algebraic curve given by the quotient $\mathcal{C}_\Delta := \Delta \backslash \mathcal{H}^*$. Most of the geometric invariants of \mathcal{C}_Δ can be read from the Kulkarni diagram \mathcal{K} . Define the following quantities:

- $e_2 = |\{\text{“red” vertices fixed by the involution in } R\}| = |R_0|$.
- $e_3 = |\{\text{“blue” vertices in } V_e\}|$.
- t = the number of orbits of the group obtained as the output of Algorithm 4.10.
- $d = [\text{PSL}_2(\mathbb{Z}) : \Delta]$.

Then in [9, §7] it is proven that the curve \mathcal{C}_Δ satisfies the following properties:

- (1) the number of branch points of $\mathcal{H} \rightarrow \mathcal{C}_\Delta$ of order 2 equals e_2 ,
- (2) the number of branch points of $\mathcal{H} \rightarrow \mathcal{C}_\Delta$ of order 3 equals e_3 ,
- (3) the number of inequivalent cusps of \mathcal{C}_Δ equals t , and
- (4) the following formula holds:

$$2g(\mathcal{C}_\Delta) + t - 1 = \frac{1}{2}|\{\text{“red” vertices non-fixed by the involution in } R\}| = f,$$

where $g(\mathcal{C}_\Delta)$ is the genus of \mathcal{C}_Δ .

In particular, the colored set V_e contains enough information to compute $g(\mathcal{C}_\Delta)$.

5. PASSPORTS AND EQUIVALENCE CLASSES OF SUBGROUPS

Let G a group and H be a subgroup of index d . Let $X = \{g_1, \dots, g_d\}$ be a set of left coset representatives for G/H . Without loss of generality, we can assume that $g_1 \in H$. Then

$$G = \bigsqcup_{i=1}^d g_i H.$$

There is a natural well known homomorphism of groups

$$\begin{aligned} \theta_H : G &\longrightarrow \mathbb{S}_X \simeq \mathbb{S}_d \\ g &\longmapsto \sigma_g, \end{aligned}$$

where σ_g is the automorphism of X satisfying that $g \cdot g_i H = \sigma_g(g_i) \cdot H$ for all i . The morphism θ_H has important properties (see for example [7, Theorems 5.3.1 and 5.3.2]), namely:

- It determines the subgroup H (with our choice it is precisely the group of elements in G fixing g_1).
- Any conjugate of H by an element of G equals the group of elements in G fixing g_i for some $1 \leq i \leq d$.
- If we denote by H^N the biggest normal subgroup of G contained in H (also known as the *normal core of H in G*), then it coincides with $\text{Ker}(\theta_H)$.
- If we denote by $\Sigma = \text{Im } \theta_H \leq \mathbb{S}_d$ (that is isomorphic to G/H^N) then Σ acts transitively on \mathbb{S}_d .

Recall the following well known result on groups.

Proposition 5.1. *Let G be a group, and let H, K be two index d subgroups of G . H is conjugated to K by an element of G if and only if there exists $\sigma \in \mathbb{S}_d$ such that $\theta_H = \sigma \theta_K \sigma^{-1}$.*

Proof. The proof is similar to that of [7, Theorem 5.3.3], although in such statement the author considers only faithful representations. Start supposing that there exists $g \in G$ such that $K = gHg^{-1}$. If $\{h_1, \dots, h_d\}$ is a set of left coset representatives for H then $\{gh_1g^{-1}, \dots, gh_dg^{-1}\}$ is a set of left coset representatives for K . Let $t \in G$ such that $t(h_i \cdot H) = h_j \cdot H$, then

$$gtg^{-1}(gh_i g^{-1}) \cdot K = (gh_j g^{-1}) \cdot K.$$

Thus if σ denotes the element of \mathbb{S}_d given by left multiplication by g , we get that $\sigma \theta_K \sigma^{-1} = \theta_H$ for the chosen set of representatives.

Conversely, suppose that there exists $\sigma \in \mathbb{S}_d$ such that $\theta_H = \sigma \theta_K \sigma^{-1}$. Without loss of generality (after reordering the left coset representatives of K), we can furthermore assume that $\theta_H = \theta_K$ and that the first coset representative for H lies in H . In particular, if $h \in H$, $\theta_H(h)(1) = 1$. Then $\theta_K(h)(1)$, so if k_1 is the first left coset representative for K , $h(k_1 \cdot K) = k_1 \cdot K$, so $h \in k_1 K k_1^{-1}$ and $H \subseteq k_1 K k_1^{-1}$. Since both groups have the same index in G , they must be equal. \square

For a general group G and a subgroup H , it is hard to describe the map θ_H , but if G has a small (and known) number of generators, then it is enough to compute the permutation corresponding to each generator. This is indeed the case for $G = \text{PSL}_2(\mathbb{Z})$, which is generated by the elements

$$(12) \quad S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}.$$

Then the pair $(\theta_H(S), \theta_H(T)) \in \mathbb{S}_d \times \mathbb{S}_d$ up to simultaneous conjugation determines the $\text{SL}_2(\mathbb{Z})$ -equivalence class of H . It is also useful to compute the permutation corresponding to the element $R = ST$ (which is the composition of the two other permutations), as it is an element of order 3 in $\text{PSL}_2(\mathbb{Z})$ (with together with S generates the group).

Definition 5.2. A *passport* is a tuple $(\sigma_S, \sigma_R, \sigma_T)$ in \mathbb{S}_d^3 up to simultaneous conjugation satisfying that

- $\sigma_S^2 = 1$,
- $\sigma_R^3 = 1$,
- $\sigma_S \sigma_T = \sigma_R$,
- $\Sigma = \langle \sigma_S, \sigma_R \rangle$ is transitive.

An important result of Millington ([12, Theorem 1]) implies that the the passport contains a lot of arithmetic information of the curve $H \backslash \mathcal{H}$.

Theorem 5.3. *There is a one-to-one correspondence between subgroups H of index d in $\mathrm{PSL}_2(\mathbb{Z})$ and equivalence classes of passports $(\sigma_S, \sigma_R, \sigma_T)$. Furthermore, keeping the previous notation,*

- (1) e_2 and e_3 are the number of elements fixed by the permutations σ_S and σ_R respectively,
- (2) σ_T has t disjoint cycles (where t equals the number of cusps).
- (3) Let d_1, \dots, d_t be the lengths of the disjoint cycle decomposition of σ_T (so $d = \sum_{i=1}^t d_i$). Then $\{d_1, \dots, d_t\}$ equals the set (with repetitions) $\{w(C_1), \dots, w(C_t)\}$ of cusp widths.

Using the algorithms described in the previous section, it is easy to give an algorithm to, given a Kulkarni diagram \mathcal{K} together with an element $\kappa \in \mathrm{PSL}_2(\mathbb{Z})$, compute the permutation σ_κ : use the algorithm of Proposition 4.15 to compute a set $\{g_1, \dots, g_d\}$ of left coset representative for $\Delta \backslash \mathrm{PSL}_2(\mathbb{Z})$. For each $1 \leq i \leq d$, use Theorem 4.14 to determine the unique j in $\{1, \dots, d\}$ such that $g_j^{-1} \kappa g_i$ belongs to Δ . Then $\sigma_\kappa(i) = j$. This algorithm is implemented in our GAP package to compute the passport attached to a Kulkarni diagram.

5.1. Congruence subgroups. Recall the following well known definition.

Definition 5.4. A subgroup Δ of $\mathrm{SL}_2(\mathbb{Z})$ is called a *congruence subgroup* if there exists a positive integer N such that the group

$$\Gamma(N) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathrm{SL}_2(\mathbb{Z}) : N \mid b, N \mid c, a \equiv 1 \pmod{N}, d \equiv 1 \pmod{N} \right\},$$

is contained in Δ . If such N does not exist, we say that the subgroup is a non-congruence subgroup. The subgroup $\Gamma(N)$ is known as the *main congruence subgroup of level N* .

The main interest in congruence subgroups is that they have many endomorphisms (given by the Hecke operators). In particular, if Δ is a congruence subgroup, then the curve \mathcal{C}_Δ is defined over a cyclotomic extension, and its Jacobian is isogenous to abelian varieties having very special properties (they are what is called of GL_2 -type, see for example [14]).

Definition 5.5. Let \mathcal{K} be a Kulkarni diagram and $\{C_1, \dots, C_t\}$ be the set of inequivalent cusps. The *generalized level* of \mathcal{K} is defined to be the least common multiple of $\{W(C_1), \dots, W(C_t)\}$.

Proposition 5.6. *If Δ is a congruence subgroup of generalized level N , then $\Gamma(N) \subseteq \Delta$.*

Proof. See [17, Theorem 2]. □

For completeness, we include the following algorithm due to Hsu for determining whether a group Δ is a congruence group or not.

Theorem 5.7. *Let Δ a subgroup of $\mathrm{PSL}_2(\mathbb{Z})$ and let*

$$\sigma_L = \theta_\Delta \left(\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right), \quad \sigma_R = \theta_\Delta \left(\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \right).$$

Let $N = e \cdot m$ be the order of L , where e is a power of 2 and m is an odd positive integer. Then the following algorithm determines whether Δ is a congruence subgroup or not:

- 1: **if** $e = 1$ **then**
- 2: **if** $\sigma_R^2 \sigma_L^{-(N+1)/2} = 1$ **then**
- 3: **return** true
- 4: **end if**
- 5: **else if** $m = 1$ **then**
- 6: $z \leftarrow \min\{x \in \mathbb{N} : N \mid 1 - 5x\}$
- 7: $\tau \leftarrow \sigma_L^{20} \sigma_R^z \sigma_L^{-4} \sigma_R^{-1}$
- 8: **if** $[\tau, \sigma_L \sigma_R^{-1} \sigma_L] = \tau^{-2}$ **and** $[\sigma_R, \tau] = \sigma_R^{24}$ **and** $(\tau \sigma_R^5 \sigma_L \sigma_R^{-1} \sigma_L)^3 = 1$ **then**
- 9: **return** true
- 10: **end if**
- 11: **else**
- 12: $z \leftarrow \min\{x \in \mathbb{N} : e \mid 1 - 5x\}$
- 13: $c \leftarrow \text{Solve} \begin{cases} x \equiv 0 \pmod{e} \\ x \equiv 1 \pmod{m} \end{cases}$

n	$\#\mathcal{K}(n)$	$SL_2(\mathbb{Z})$ -classes	$GL_2(\mathbb{Z})$ -classes
2	1	1	1
3	2	2	2
4	2	2	2
5	1	1	1
6	9	8	8
7	8	6	4
8	8	7	6
9	54	14	12
10	101	27	19
11	80	26	16
12	440	80	63
13	790	133	73
14	770	170	106
15	3184	348	213
16	6540	765	428
17	6582	1002	533
18	28958	2176	1277
19	61072	4682	2410
20	68920	6931	3679

TABLE 6.1. Number of Kulkarni diagram vs SL_2 and GL_2 conjugacy classes.

```

14:   $d \leftarrow \text{Solve} \begin{cases} x \equiv 1 \pmod{e} \\ x \equiv 0 \pmod{m} \end{cases}$ 
15:   $a \leftarrow \sigma_L^c$ 
16:   $b \leftarrow \sigma_R^c$ 
17:   $l \leftarrow \sigma_L^d$ 
18:   $r \leftarrow \sigma_R^d$ 
19:   $s \leftarrow l^{20}r^z l^{-4}r^{-1}$ 
20:  if  $[a, r] = 1$  and  $(ab^{-1}a)^4 = 1$  and  $(ab^{-1}a)^2 = (b^{-1}a)^3 = (b^2a^{-(m+1)/2})^3$  and  $[s, lr^{-1}l] = s^{-2}$  and
     $[r, s] = r^{24}$  and  $(lr^{-1}l)^2 = (sr^5lr^{-1}l)^3$  then
21:    return true
22:  end if
23: end if
24: return false

```

Proof. See [8, Theorem 2.4] and §3 of loc. cit. for the implementation. \square

6. SOME NUMERICAL DATA

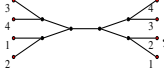
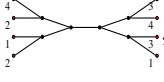
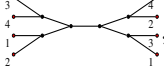
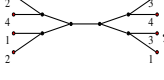
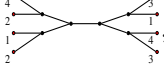
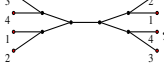
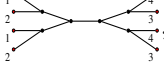
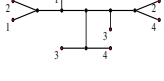
We have systematically run our algorithm to compute Kulkarni diagrams for subgroups up to index 20. For each of them, we computed the number of conjugacy classes for SL_2 and GL_2 -subgroups. The results are presented in Table 6.1. Note that the number of Kulkarni diagrams grows much faster than the real number of equivalent classes of subgroups. To our knowledge this is a phenomena that was not observed before (since in the article published by Kulkarni only subgroups with small index are computed). As mentioned in the introduction, all these information can be downloaded from the GitHub repository <https://github.com/vendramin/subgroups>.

In Table 6.2 we give all the information obtained from each $SL_2(\mathbb{Z})$ -equivalence class for subgroups up to index 7 (as it might be useful to the reader). Some information phenomena obtained from our database:

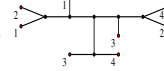
- Among all subgroups (up to $SL_2(\mathbb{Z})$ -equivalence), there are 90 congruence ones (from a total of 16381 subgroups), i.e. congruence groups are very rare even for small indices.
- Up to index 20, there are 2410 groups corresponding to curves of genus 1 (from a total of 16381 subgroups).

- There are only 9 non-conjugate subgroups whose curve has genus 2. They all correspond to subgroups of index 18 (as the example found in [1]).

Let us state some properties regarding the genus 2 curves. Here is a list of their Kulkarni diagrams together with their passports:

- (1) The tree diagram equals , the g.F.s. equals $\{\infty, 0, 1, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, 2, 3, \infty\}$ and its passport equals
 - $\sigma_S = (1, 5)(2, 8)(3, 15)(4, 9)(6, 12)(7, 16)(10, 14)(11, 17)(13, 18),$
 - $\sigma_R = (1, 8, 4)(2, 15, 7)(3, 18, 14)(5, 12, 9)(6, 16, 11)(10, 17, 13),$
 - $\sigma_T = (1, 2, 3, 13, 14, 15, 16, 17, 18, 10, 11, 12, 4, 5, 6, 7, 8, 9).$
- (2) The tree diagram equals , the g.F.s. equals $\{\infty, 0, 1, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, 2, 3, \infty\}$ and its passport equals
 - $\sigma_S = (1, 5)(2, 8)(3, 15)(4, 18)(6, 12)(7, 16)(9, 13)(10, 14)(11, 17),$
 - $\sigma_R = (1, 8, 4)(2, 15, 7)(3, 18, 14)(5, 12, 9)(6, 16, 11)(10, 17, 13),$
 - $\sigma_T = (1, 2, 3, 4, 5, 6, 7, 8, 18, 10, 11, 12, 13, 14, 15, 16, 17, 9).$
- (3) The tree diagram equals , the g.F.s. equals $\{\infty, 0, 1, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, 2, 3, \infty\}$ and its passport equals
 - $\sigma_S = (1, 5)(2, 8)(3, 15)(4, 10)(6, 12)(7, 16)(9, 14)(11, 17)(13, 18),$
 - $\sigma_R = (1, 8, 4)(2, 15, 7)(3, 18, 14)(5, 12, 9)(6, 16, 11)(10, 17, 13),$
 - $\sigma_T = (1, 2, 3, 13, 4, 5, 6, 7, 8, 10, 11, 12, 14, 15, 16, 17, 18, 9).$
- (4) The tree diagram equals , the g.F.s. equals $\{\infty, 0, 1, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, 2, 3, \infty\}$ and its passport equals
 - $\sigma_S = (1, 5)(2, 8)(3, 15)(4, 14)(6, 12)(7, 16)(9, 13)(10, 18)(11, 17),$
 - $\sigma_R = (1, 8, 4)(2, 15, 7)(3, 18, 14)(5, 12, 9)(6, 16, 11)(10, 17, 13),$
 - $\sigma_T = (1, 2, 3, 10, 11, 12, 13, 18, 4, 5, 6, 7, 8, 14, 15, 16, 17, 9).$
- (5) The tree diagram equals , the g.F.s. equals $\{\infty, 0, 1, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, 2, 3, \infty\}$ and its passport equals
 - $\sigma_S = (1, 10)(2, 8)(3, 15)(4, 18)(5, 13)(6, 12)(7, 16)(9, 14)(11, 17),$
 - $\sigma_R = (1, 8, 4)(2, 15, 7)(3, 18, 14)(5, 12, 9)(6, 16, 11)(10, 17, 13),$
 - $\sigma_T = (1, 2, 3, 4, 10, 11, 12, 14, 15, 16, 17, 5, 6, 7, 8, 18, 9, 13).$
- (6) The tree diagram equals , the g.F.s. equals $\{\infty, 0, 1, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, 2, 3, \infty\}$ and its passport equals
 - $\sigma_S = (1, 10)(2, 8)(3, 15)(4, 13)(5, 14)(6, 12)(7, 16)(9, 18)(11, 17),$
 - $\sigma_R = (1, 8, 4)(2, 15, 7)(3, 18, 14)(5, 12, 9)(6, 16, 11)(10, 17, 13),$
 - $\sigma_T = (1, 2, 3, 9, 14, 15, 16, 17, 4, 10, 11, 12, 18, 5, 6, 7, 8, 13).$
- (7) The tree diagram equals , the g.F.s. equals $\{\infty, 0, 1, \frac{4}{3}, \frac{3}{2}, \frac{5}{3}, 2, 3, \infty\}$ and its passport equals
 - $\sigma_S = (1, 14)(2, 8)(3, 15)(4, 18)(5, 10)(6, 12)(7, 16)(9, 13)(11, 17),$
 - $\sigma_R = (1, 8, 4)(2, 15, 7)(3, 18, 14)(5, 12, 9)(6, 16, 11)(10, 17, 13),$
 - $\sigma_T = (1, 2, 3, 4, 14, 15, 16, 17, 9, 10, 11, 12, 13, 5, 6, 7, 8, 18).$
- (8) The tree diagram equals , the g.F.s. equals $\{\infty, 0, \frac{1}{2}, 1, \frac{3}{2}, 2, 3, 4, \infty\}$ and its passport equals

- $\sigma_S = (1, 7)(2, 10)(3, 14)(4, 17)(5, 18)(6, 11)(8, 13)(9, 15)(12, 16),$
- $\sigma_R = (1, 10, 6)(2, 14, 9)(3, 17, 13)(4, 18, 16)(5, 11, 7)(8, 15, 12),$
- $\sigma_T = (1, 2, 3, 4, 5, 6, 7, 18, 12, 13, 14, 15, 16, 17, 8, 9, 10, 11).$

(9) The tree diagram equals , the g.f.s. equals $\{\infty, 0, \frac{1}{2}, 1, \frac{3}{2}, 2, 3, 4, \infty\}$ and its passport equals

- $\sigma_S = (1, 7)(2, 10)(3, 14)(4, 17)(5, 16)(6, 11)(8, 13)(9, 15)(12, 18),$
- $\sigma_R = (1, 10, 6)(2, 14, 9)(3, 17, 13)(4, 18, 16)(5, 11, 7)(8, 15, 12),$
- $\sigma_T = (1, 2, 3, 4, 12, 13, 14, 15, 18, 5, 6, 7, 16, 17, 8, 9, 10, 11).$

The order of the image of θ_Δ in each case equals 1008, 258048, 486, 4896, 258048, 648, 258048, 4896, 4896 respectively. Furthermore, the fourth, the eighth one and the ninth ones are isomorphic to $\mathrm{PGL}_2(\mathbb{F}_{17})$. Then this three groups must be the ones found by Atkin and Swinnerton-Dyer in [1, Table 1] (corresponding to a curve defined over the cubic field with generating polynomial $x^3 - 3x + 1$).

Recall that the modular curve attached to a congruence subgroup has many endomorphisms (corresponding to the so called Hecke operators), while modular curves for non-congruence subgroups tend to not have endomorphisms at all (there are no Hecke operators in the new part, as proved in [2]). In particular, for each of the previous genus 2 curves, if there is no contribution from smaller levels, one expects the Jacobian to be absolutely simple when the subgroup is a non-congruence one.

The third one and the seventh groups are congruence ones, hence they do have many endomorphisms. Let us study what happens for the remaining seven ones. Here is an elementary result to determine whether a group is contained in a larger (proper) subgroup of $\mathrm{PSL}_2(\mathbb{Z})$ in terms of the passport representation.

Lemma 6.1. *Let Δ be a subgroup of $\mathrm{PSL}_2(\mathbb{Z})$ of index d , with coset representatives $\{\mu_1, \dots, \mu_d\}$. Let σ_S, σ_T be its permutation representation. Then there exists a subgroup $\tilde{\Delta}$ of $\mathrm{PSL}_2(\mathbb{Z})$ of index n (with $d \mid n$) containing Δ if and only if the set $\{\mu_1, \dots, \mu_d\}$ can be written as a disjoint union of n sets each of them with $\frac{d}{n}$ elements satisfying that σ_S and σ_T preserve the partition.*

Proof. If there exists such a subgroup $\tilde{\Delta}$, let $\{g_1, \dots, g_n\}$ be coset representatives for $\tilde{\Delta} \backslash \mathrm{PSL}_2(\mathbb{Z})$ and $\{h_1, \dots, h_{\frac{d}{n}}\}$ be coset representatives for $\Delta \backslash \tilde{\Delta}$, so that $\{h_j g_i\}$ are representatives for $\Delta \backslash \mathrm{PSL}_2(\mathbb{Z})$. In particular, the set of representatives for $\Delta \backslash \mathrm{PSL}_2(\mathbb{Z})$ can be written as the disjoint union

$$(13) \quad \bigcup_{i=1}^n \{h_1 g_i, \dots, h_{\frac{d}{n}} g_i\}.$$

Let ν be any element of $\mathrm{PSL}_2(\mathbb{Z})$. If $\tilde{\Delta} g_i \nu = \tilde{\Delta} g_j$ then the sets

$$\{\Delta h_1 g_i \nu, \dots, \Delta h_{\frac{d}{n}} g_i \nu\} \text{ and } \{\Delta h_1 g_j, \dots, \Delta h_{\frac{d}{n}} g_j\}$$

must be the same, so σ_S and σ_T preserve the partition (13).

Reciprocally, let $\{g_1, \dots, g_d\}$ be coset representatives of $\Delta \backslash \mathrm{PSL}_2(\mathbb{Z})$, and let $\theta_\Delta: \mathrm{PSL}_2(\mathbb{Z}) \rightarrow \mathbb{S}_d$ be as before (identifying the set $\{g_1, \dots, g_d\}$ with the set $\{1, \dots, d\}$). Assume that $g_1 \in \Delta$, so that Δ corresponds precisely to the elements of $\mathrm{PSL}_2(\mathbb{Z})$ whose image under θ_Δ fix 1.

By hypothesis (after relabeling the indexes if necessary) the set $\{1, \dots, d\}$ can be written as the disjoint union of the n sets $\{1, \dots, \frac{d}{n}\} \cup \dots \cup \{d - \frac{d}{n} + 1, \dots, d\}$, and our representation θ_Δ induces a morphism $\tilde{\theta}_\Delta$ of $\mathrm{PSL}_2(\mathbb{Z})$ onto \mathbb{S}_n (via the action on the previous sets). Let $\tilde{\Delta}$ the set of elements of $\mathrm{PSL}_2(\mathbb{Z})$ that preserve the set $\{1, \dots, \frac{d}{n}\}$. Clearly $\tilde{\Delta}$ contains Δ , and the index of $\tilde{\Delta}$ on $\mathrm{PSL}_2(\mathbb{Z})$ is n because the image of θ_Δ is a transitive group (so the same holds for $\tilde{\theta}_\Delta$). \square

Let us compute for each of our genus 2 subgroups Δ their “old” part. If there exists a subgroup $\tilde{\Delta}$ with index 9 in $\mathrm{PSL}_2(\mathbb{Z})$, then the lemma implies that our set of 18 elements can be split into 9 sets of size 2, which are preserved under the action of $\mathrm{PSL}_2(\mathbb{Z})$. Note that σ_T is an 18-cycle in all cases. Suppose that such a partition exists, and that $\{a, b\}$ is a pair of elements of it. Since σ_T acts transitively, there exists i such that $b = \sigma_T^i(a)$. Applying σ_T^i to the pair $\{a, b\}$ we obtain the pair $\{\sigma_T^i(a), \sigma_T^i(b)\} = \{b, \sigma_T^{2i}(a)\}$, so $a = \sigma_T^{2i}(a)$, hence $i = 9$. In particular, the partition must be of the form $\{a, \sigma_T^9(a)\}$ for a varying in the set

$\{1, \dots, 18\}$. Then we are led to verify whether the permutation σ_S preserves this partition or not. Consider each of the nine cases separately.

- (1) We obtain the partition:

$$\{1, 10\}\{2, 11\}\{3, 12\}\{4, 13\}\{5, 14\}\{6, 15\}\{7, 16\}\{8, 17\}\{9, 18\}.$$

Clearly the element σ_S preserves the partition, so this contains the subgroup of index 9 with passport $\sigma_S = (1, 5)(2, 8)(3, 6)(4, 9)$ and $\sigma_R = (1, 8, 4)(2, 6, 7)(3, 9, 5)$. It matches the 12th element in the GitHub repository of subgroups of index 9. It is easy to verify that it corresponds to a genus 1 curve. In particular, the Jacobian of the curve is isogenous to the product of two elliptic curves.

- (2) A similar computation as the previous one proves that this second group is contained in an index 9 subgroup with the same passport as the previous case. Once again, the surface is isogenous to the product of two elliptic curves.
- (3) In this case, the partition obtained from σ_T is not compatible with the permutation σ_S , hence the group is not contained in a group of index 9 (recall that this group is a congruence one).
- (4) The group is not contained in an index 9 subgroup because once again the partition is not compatible with the action of σ_S .
- (5) The group is contained in a group of index 9 with the same passport as the first two cases.
- (6) In this case, the group is contained in a subgroup of index 9, with passport $\sigma_S = (2, 8)(3, 6)$, $\sigma_R = (1, 8, 4)(2, 6, 7)(3, 9, 5)$. Such a group corresponds to a curve of genus zero.
- (7) The group is contained in a group of index 9 with the same passport as the first two cases.
- (8) The group is not contained in an index 9 subgroup because the partition is not compatible with the action of σ_S .
- (9) The group is not contained in an index 9 subgroup because the partition is not compatible with the action of σ_S .

Although the unique subgroup of index 2 has genus zero, a similar computation shows that the groups 1, 3, 6 are contained in the subgroup of index 2, while the other ones are not.

Verifying which groups are contained in subgroups of index 6, we see that the only ones are the first one (contained in the subgroup with passport $\sigma_S = (1, 5)(2, 3)(4, 6)$ and $\sigma_R = (1, 3, 4)$) and the sixth one, contained in the subgroup with passport $\sigma_S = (1, 4)(2, 5)(3, 6)$ and $\sigma_R = (1, 5, 3)(2, 6, 4)$, corresponding to the last subgroup of index 6 in the GitHub repository (see also Table 6.2). Such a group corresponds to a curve of genus one, hence the surface attached to it is isogenous to the product of two elliptic curves as well. We deduce that the only possible absolutely simple surface from the list of index 18 subgroups are the ones corresponding to the groups 6, 8 and 9 which match the ones found by Atkin and Swinnerton-Dyer (defined over a cubic field), while all other ones are isogenous to the product of two elliptic curves.

REFERENCES

- [1] A. O. L. Atkin and H. P. F. Swinnerton-Dyer. Modular forms on noncongruence subgroups. In *Combinatorics (Proc. Sympos. Pure Math., Vol. XIX, Univ. California, Los Angeles, Calif., 1968)*, pages 1–25. , 1971.
- [2] G. Berger. Hecke operators on noncongruence subgroups. *C. R. Acad. Sci. Paris Sér. I Math.*, 319(9):915–919, 1994.
- [3] D. Berghaus, H. Monien, and D. Radchenko. On the computation of modular forms on noncongruence subgroups, 2022.
- [4] D. Berghaus, H. Monien, and D. Radchenko. A database of modular forms on noncongruence subgroups, 2023.
- [5] A. Dooms, E. Jespers, and A. Konovalov. From Farey symbols to generators for subgroups of finite index in integral group rings of finite groups. *J. K-Theory*, 6(2):263–283, 2010.
- [6] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.12.2*, 2022.
- [7] M. Hall, Jr. *The theory of groups*. Chelsea Publishing Co., New York, 1976. Reprinting of the 1968 edition.
- [8] T. Hsu. Identifying congruence subgroups of the modular group. *Proc. Amer. Math. Soc.*, 124(5):1351–1359, 1996.
- [9] R. S. Kulkarni. An arithmetic-geometric method in the study of the subgroups of the modular group. *Amer. J. Math.*, 113(6):1053–1133, 1991.
- [10] C. A. Kurth and L. Long. Computations with finite index subgroups of $\mathrm{PSL}_2(\mathbb{Z})$ using Farey symbols. In *Advances in algebra and combinatorics*, pages 225–242. World Sci. Publ., Hackensack, NJ, 2008.
- [11] M. L. Lang, C.-H. Lim, and S. P. Tan. An algorithm for determining if a subgroup of the modular group is congruence. *Journal of The London Mathematical Society-second Series*, 51:491–502, 1995.
- [12] M. H. Millington. Subgroups of the classical modular group. *J. London Math. Soc. (2)*, 1:351–357, 1969.
- [13] M. Newman. Asymptotic formulas related to free products of cyclic groups. *Math. Comp.*, 30(136):838–846, 1976.

Index	Kulkarni Graph	g.F.S.	Passport	Genus	e_2	e_3	Cusps & Width	Generators	Congruence
2		$\{\infty, 0, \infty\}$	$(1, 2), (), (1, 2)$	0	0	2	$(\{\infty, 0\}, 2)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 1 & -1 \end{pmatrix}$	Yes
3		$\{\infty, 0, 1, \infty\}$	$(), (1, 3, 2), (1, 3, 2)$	0	3	0	$(\{\infty, 0, 1\}, 3)$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & -2 \\ 1 & -1 \end{pmatrix}$	Yes
3		$\{\infty, 0, 1, \infty\}$	$(2, 3), (1, 3, 2), (1, 2)$	0	1	0	$(\{\infty, 0\}, 2), (\{1\}, 2)$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix}$	Yes
4		$\{\infty, 0, 1, \infty\}$	$(2, 3), (2, 4, 3), (1, 2, 4, 3)$	0	2	1	$(\{\infty, 0, 1\}, 4)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 2 & -1 \end{pmatrix}, \begin{pmatrix} 1 & -2 \\ 1 & -1 \end{pmatrix}$	Yes
4		$\{\infty, 0, 1, \infty\}$	$(1, 2)(3, 4), (2, 4, 3), (1, 2, 3)$	0	0	1	$(\{\infty, 0\}, 3), (\{2\}, 3)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix}$	Yes
5		$\{\infty, 0, 1, \infty\}$	$(1, 2)(3, 4), (2, 5, 4), (1, 2, 5, 3, 4)$	0	1	2	$(\{\infty, 0, 1\}, 5)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -2 \end{pmatrix}, \begin{pmatrix} 1 & -2 \\ 1 & -1 \end{pmatrix}$	Yes
6		$\{\infty, 0, 1, \infty\}$	$(1, 2)(3, 4)(56), (2, 6, 4), (1, 2, 5, 6, 3, 4)$	0	0	3	$(\{\infty, 0, 1\}, 6)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -2 \end{pmatrix}, \begin{pmatrix} 1 & -3 \\ 1 & -2 \end{pmatrix}$	Yes
6		$\{\infty, 0, 1, 2, \infty\}$	$(1, 5), (1, 5, 3)(2, 6, 4), (1, 2, 6, 4, 5, 3)$	0	4	0	$(\{\infty, 0, 1, 2\}, 6)$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -5 \end{pmatrix}, \begin{pmatrix} 2 & -5 \\ 2 & -5 \end{pmatrix}$	Yes
6		$\{\infty, 0, 1, 2, \infty\}$	$(2, 5)(4, 6), (1, 5, 3)(2, 6, 4), (1, 2, 4, 5, 3)$	0	2	0	$(\{\infty, 0, 1, \}, 5), (\{2\}, 5)$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -1 \end{pmatrix}, \begin{pmatrix} 2 & -1 \\ 3 & -1 \end{pmatrix}$	Yes
6		$\{\infty, 0, 1, 2, \infty\}$	$(2, 5)(3, 6), (1, 5, 3)(2, 6, 4), (1, 2, 3)(4, 5, 6)$	0	2	0	$(\{\infty, 0, \}, 3), (\{1, 2\}, 3)$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 3 & -1 \\ 3 & -5 \end{pmatrix}, \begin{pmatrix} 2 & -5 \\ 2 & -5 \end{pmatrix}$	Yes
6		$\{\infty, 0, 1, 2, \infty\}$	$(1, 6)(2, 5), (1, 5, 3)(2, 6, 4), (1, 2)(3, 6, 4, 5)$	0	2	0	$(\{\infty, \}, 2)(\{0, 1, 2\}, 4)$	$\begin{pmatrix} 1 & 2 \\ 3 & -5 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -5 \end{pmatrix}, \begin{pmatrix} 2 & -5 \\ 2 & -5 \end{pmatrix}$	Yes
6		$\{\infty, 0, 1, 2, \infty\}$	$(1, 3)(2, 5)(4, 6), (1, 5, 3)(2, 6, 4), (1, 2, 4, 5)$	0	0	0	$(\{\infty, 1\}, 4)(\{0\}, 1)(\{2\}, 4)$	$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \begin{pmatrix} 3 & -1 \\ 3 & -1 \end{pmatrix}$	Yes
6		$\{\infty, 0, 1, 2, \infty\}$	$(1, 6)(2, 5)(3, 4), (1, 5, 3)(2, 6, 4), (1, 2)(3, 6)(4, 5)$	0	0	0	$(\{\infty, \}, 2)(\{0, 2\}, 2)(\{1\}, 2)$	$\begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}, \begin{pmatrix} 3 & -2 \\ 3 & -2 \end{pmatrix}$	Yes
6		$\{\infty, 0, 1, 2, \infty\}$	$(1, 4)(2, 5)(3, 6), (1, 5, 3)(2, 6, 4), (1, 2, 3, 4, 5, 6)$	1	0	0	$(\{\infty, 0, 1, 2\}, 6)$	$\begin{pmatrix} 2 & 1 \\ 2 & 1 \end{pmatrix}, \begin{pmatrix} 3 & -1 \\ 3 & 0 \end{pmatrix}$	Yes
7		$\{\infty, 0, 1, 2, \infty\}$	$(1, 2)(3, 6), (2, 6, 4)(3, 7, 5), (1, 2, 3, 7, 5, 6, 4)$	0	3	1	$(\{\infty, 0, 1, 2\}, 7)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -5 \end{pmatrix}, \begin{pmatrix} 2 & -5 \\ 2 & -5 \end{pmatrix}$	Yes
7		$\{\infty, 0, 1, 2, \infty\}$	$(2, 6)(3, 4), (1, 6, 4)(2, 7, 5), (1, 2, 7, 5, 6, 3, 4)$	0	3	1	$(\{\infty, 0, 1, 2\}, 7)$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -5 \end{pmatrix}, \begin{pmatrix} 2 & -5 \\ 2 & -5 \end{pmatrix}$	Yes
7		$\{\infty, 0, 1, 2, \infty\}$	$(1, 2)(3, 6)(5, 7), (2, 6, 4)(3, 7, 5), (1, 2, 3, 5, 6, 4)$	0	1	1	$(\{\infty, 0, 1, \}, 6)(\{2\}, 6)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -4 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$	No
7		$\{\infty, 0, 1, 2, \infty\}$	$(1, 2)(3, 6)(4, 7), (2, 6, 4)(3, 7, 5), (1, 2, 3, 4)(5, 6, 7)$	0	1	1	$(\{\infty, 0, \}, 4)(\{1, 2\}, 4)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 3 & -1 \\ 2 & -3 \end{pmatrix}, \begin{pmatrix} 3 & -1 \\ 2 & -3 \end{pmatrix}$	No
7		$\{\infty, 0, 1, 2, \infty\}$	$(1, 2)(3, 6)(4, 5), (2, 6, 4)(3, 7, 5), (1, 2, 3, 7, 4)(5, 6)$	0	1	1	$(\{\infty, 0, 2\}, 5)(\{1\}, 5)$	$\begin{pmatrix} -1 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 3 & -2 \\ 2 & -3 \end{pmatrix}, \begin{pmatrix} 3 & -2 \\ 1 & -2 \end{pmatrix}$	No
7		$\{\infty, 0, 1, 2, \infty\}$	$(2, 6)(3, 4)(5, 7), (1, 6, 4)(2, 7, 5), (1, 2, 5, 6, 3, 4)$	0	1	1	$(\{\infty, 0, 1, \}, 6)(\{2\}, 6)$	$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 3 & -3 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$	No

TABLE 6.2. Table of Subgroups up to index 7 with all the computed information.

- [14] G. Shimura. *Introduction to the arithmetic theory of automorphic functions*, volume 11 of *Publications of the Mathematical Society of Japan*. Princeton University Press, Princeton, NJ, 1994. Reprint of the 1971 original, Kanô Memorial Lectures, 1.
- [15] F. Strömberg. Noncongruence subgroups and Maass waveforms. *J. Number Theory*, 199:436–493, 2019.
- [16] S. A. Vidal. Sur la classification et le denombrement des sous-groupes du groupe modulaire et de leurs classes de conjugaison, 2007.
- [17] K. Wohlfahrt. An extension of F. Klein’s level concept. *Illinois J. Math.*, 8:529–535, 1964.

(N. Mayorga) FAMAF-CIEM, UNIVERSIDAD NACIONAL DE CÓRDOBA. C.P:5000, CÓRDOBA, ARGENTINA.
Email address: nmayorga@unc.edu.ar

(A. Pacetti) CENTER FOR RESEARCH AND DEVELOPMENT IN MATHEMATICS AND APPLICATIONS (CIDMA), DEPARTMENT OF MATHEMATICS, UNIVERSITY OF AVEIRO, 3810-193 AVEIRO, PORTUGAL
Email address: apacetti@ua.pt

(L. Vendramin) DEPARTMENT OF MATHEMATICS, VRIJE UNIVERSITEIT BRUSSEL, PLEINLAAN 2, 1050 BRUSSEL, BELGIUM
Email address: Leandro.Vendramin@vub.be