Random Generation of Subgroups of the Modular Group with a Fixed Isomorphism Type

Frédérique Bassino[a] Cyril Nicaud[b] Pascal Weil[a,c]

Submitted: Nov 15, 2023; Accepted: Sep 8, 2024; Published: Oct 4, 2024

Abstract

We show how to efficiently count and generate uniformly at random finitely generated subgroups of the modular group $\mathsf{PSL}_2(\mathbb{Z})$ of a given isomorphism type. The method to achieve these results relies on a natural map of independent interest, which associates with any finitely generated subgroup of $\mathsf{PSL}_2(\mathbb{Z})$ a graph which we call its silhouette, and which can be interpreted as a conjugacy class of free finite index subgroups of $\mathsf{PSL}_2(\mathbb{Z})$.

Mathematics Subject Classifications: 05A15,05A16, 05C30

1 Introduction

The modular group $\mathsf{PSL}_2(\mathbb{Z})$ is a fundamental object in the field of modular forms and hyperbolic geometry. It is well-known that $\mathsf{PSL}_2(\mathbb{Z})$ is isomorphic to the free product of two cyclic groups, of order 2 and 3 respectively. That is,

$$\mathsf{PSL}_2(\mathbb{Z}) = \langle a, b \mid a^2 = b^3 = 1\rangle.$$

The finitely generated subgroups of the modular group have been extensively studied and classified, leading to deep connections with various areas of mathematics, including number theory, algebraic geometry, and geometric group theory. Much work has been devoted in particular to the combinatorial study of the *finite index* subgroups of $\mathsf{PSL}_2(\mathbb{Z})$: exact enumeration results for the index n subgroups (Dey, 1965 [5]; Stothers, 1978 [23]) and results on the asymptotic behavior of that number as n tends to infinity (Newmann, 1976 [18], Müller & Schlage-Puchta, 2004 [17] and others). Here, we deal instead with *all* finitely generated subgroups of $\mathsf{PSL}_2(\mathbb{Z})$, without index restriction. Our motivation here is

[a] Université Sorbonne Paris Nord, LIPN, CNRS UMR 7030, F-93430 Villetaneuse, France (bassino@lipn.fr, pascal.weil@cnrs.fr).
[b] LIGM, Univ Gustave Eiffel, CNRS, ESIEE Paris, F-77454, Marne-la-Vallée, France (cyril.nicaud@univ-eiffel.fr).
[c] CNRS, ReLaX, IRL 2000, Siruseri, India.

https://doi.org/10.37236/12559

to advance the asymptotic study of finitely generated subgroups of $\mathsf{PSL}_2(\mathbb{Z})$, irrespective of their index; we refer the readers to [4] for first results in this direction. The asymptotic study of subgroups of infinite groups in general has received much attention, at least since Gromov's work on hyperbolic groups, see Ollivier's survey [19].

The main purpose of this paper is to present enumeration and random generation results for finitely generated subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ of a given size and isomorphism type, where both measures are natural parameters.

Let us first make the notions underlying these results more explicit. Since $\mathsf{PSL}_2(\mathbb{Z})$ is the free product of a copy of $\mathbb{Z}/2\mathbb{Z}$ and a copy of $\mathbb{Z}/3\mathbb{Z}$, Kurosh's theorem (see, *e.g.*, [14, 15, 20]) states that any finitely generated subgroup H of $\mathsf{PSL}_2(\mathbb{Z})$ is isomorphic to a free product of ℓ_2 copies of $\mathbb{Z}/2\mathbb{Z}$, ℓ_3 copies of $\mathbb{Z}/3\mathbb{Z}$ and r copies of $\mathbb{Z}$: the *isomorphism type* of H is the triple (ℓ_2, ℓ_3, r). It is a natural parameter, which generalizes the rank in free groups.

Our results also refer to a notion of size for finitely generated subgroups of $\mathsf{PSL}_2(\mathbb{Z})$, that we now explain: each finitely generated subgroup H of $\mathsf{PSL}_2(\mathbb{Z})$ can be represented uniquely by a finite edge-labeled graph $\Gamma(H)$, called its *Stallings graph.* Stallings graphs, and their effective construction, were first introduced by Stallings [22] to represent finitely generated subgroups of free groups. The idea of using finite graphs to represent subgroups of infinite, non-free groups first appeared in work of Gersten and Short [8, 21], Arzhantseva and Ol'shanskiĭ [2, 1], Gitik [9] and Kapovich [11]. Markus-Epstein [16] gave an explicit construction associating a graph with each subgroup of an amalgamated product of two finite groups, which is very close to the one used here. Here we follow the definition and construction of Kharlampovich, Miasnikov and Weil [12]. In a nutshell, the Stallings graph of a subgroup H of $\mathsf{PSL}_2(\mathbb{Z})$ is the fragment of the Schreier (or coset) graph of H, spanned by the cycles at vertex H reading a geodesic representative of an element of H, see Section 2.1 for more details. In particular, if H has finite index in $\mathsf{PSL}_2(\mathbb{Z})$, then the Stallings graph of H is its Schreier graph.

We take the number of vertices of $\Gamma(H)$ to be the *size* of the subgroup H. As we just saw, this parameter generalizes the index of finite index subgroups.

It is important to note that there are only finitely many subgroups of a given size and we assume the uniform distribution on this finite set. Contrary to what happens with finite index subgroups, $\mathsf{PSL}_2(\mathbb{Z})$ has infinitely many subgroups of a given isomorphism type, see Remark 6. So there can be no counting of subgroups of a given isomorphism type. Our objective will be, therefore, to count and randomly generate subgroups of a given size and isomorphism type.

In [3] the authors counted the finitely generated subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ by size and they showed how to generate uniformly at random a subgroup of a given size. They also computed the expected value of the isomorphism type of a random subgroup as a function of its size and proved a large deviations theorem for this isomorphism type. It follows that randomly generating a size n subgroup of $\mathsf{PSL}_2(\mathbb{Z})$ will, with high probability, yield a subgroup whose isomorphism type is close to the average value. In particular, this algorithm may not be suitable to test certain conjectures which are sensitive to isomorphism type, and it does not help generate uniformly at random subgroups of a given size and isomorphism type.

The proof strategy to obtain the results in [3] was based on counting Stallings graphs and using the classical tools of analytic combinatorics [6], in particular the notion of exponential generating series. In this paper, we use a completely different enumeration method for finitely generated subgroups of $\mathsf{PSL}_2(\mathbb{Z})$, to get a polynomial time random generation algorithm for subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ of a given size and isomorphism type. It turns out that we can proceed with direct computations and we therefore avoid introducing generating series. More precisely the proofs rely on a combination of graph decomposition techniques and combinatorial methods. As is classical in the field, these methods are used on labeled graphs (graphs equipped with a bijection from their vertex set to an initial segment of $\mathbb{N}$).

A key construction which occurs naturally in this approach is what we call the *silhouetting* of the Stallings graph of a finitely generated subgroup of $\mathsf{PSL}_2(\mathbb{Z})$. It consists in a sequence of “simplifications” of the graph, leading (except in extremal cases) to a uniform degree loop-free graph, which represents a conjugacy class of finite index, free subgroups of $\mathsf{PSL}_2(\mathbb{Z})$.

The operation of silhouetting is not just useful for our enumeration and random generation purpose: it also has very interesting algebraic and probabilistic properties. As an example of the former, we establish that silhouetting preserves the free rank component of the isomorphism type of a subgroup (Proposition 13). Probabilistic properties of the silhouetting operation, and their use in proving asymptotic properties of finitely generated subgroups of $\mathsf{PSL}_2(\mathbb{Z})$, are discussed in a separate paper [4].

Organization of the paper Readers can find in Sections 2.1 and 2.2 the precise definitions of the Stallings graph of a subgroup of $\mathsf{PSL}_2(\mathbb{Z})$ and its combinatorial type, and results from the literature relating this combinatorial information with algebraic properties of the subgroup such as its isomorphism type, its index or its freeness.

Section 3 introduces combinatorial operations on Stallings graphs. Iterating these operations leads to so-called *silhouette* graphs. The fine description of these operations is first exploited in Section 4 to give exact counting formulas for the number of subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ of a given combinatorial or isomorphism type.

In Section 3.3, we show that the iteration of the operations defined in Section 4 is a confluent process (Proposition 11), which leads to defining the *silhouette* of a given graph or subgroup. It is interesting to note that silhouetting preserves the free rank component of the isomorphism type of a subgroup (Proposition 13).

Finally, Section 5 uses the operations from Section 3 in a different way to design an algorithm (which includes a rejection algorithm component) to efficiently generate uniformly at random a subgroup of a given size and isomorphism type.

2 Preliminaries

We work with the following presentation of the modular group:

$$\mathsf{PSL}_2(\mathbb{Z}) = \langle a, b \mid a^2 = b^3 = 1 \rangle.$$

The elements of $\mathsf{PSL}_2(\mathbb{Z})$ are represented by words over the alphabet $\{a, b, a^{-1}, b^{-1}\}$. Since $a^{-1} = a$ in $\mathsf{PSL}_2(\mathbb{Z})$, we can eliminate the letter a^{-1} from this alphabet. Each element of $\mathsf{PSL}_2(\mathbb{Z})$ then has a unique shortest (or *normal*, or *geodesic*) representative, which is a freely reduced word without factors in $\{a^2, b^2, b^{-2}\}$. That is, the normal representatives are the words of length at most 1 and the words alternating letters a and letters in $\{b, b^{-1}\}$.

2.1 Stallings graph of a subgroup of $\mathsf{PSL}_2(\mathbb{Z})$

The *Schreier graph* (or *coset graph*) of a subgroup H of $\mathsf{PSL}_2(\mathbb{Z})$ is the graph whose vertices are the cosets Hg of H ($g \in \mathsf{PSL}_2(\mathbb{Z})$), with an a-labeled edge from Hg to Hga and a b-labeled edge from Hg to Hgb, for every $g \in G$. We think of b-edges as 2-way edges, reading b in the forward direction and b^{-1} in the backward direction. Since $a = a^2$, there is an a-edge from vertex v to vertex v' if and only there is one from v' to v: as a result, we think of the a-edges as undirected edges, that can be traveled in either direction, each time reading a. A path in such an edge-labeled graph is called a *cycle* if its initial vertex is equal to its final vertex. The sequence of edge labels along a path p spells a word w over alphabet $\{a, b, b^{-1}\}$, and we say that w *labels the path* p, or that p *reads* w.

Note that a word is in H if and only if it labels a cycle at vertex $v_0 = H$ in the Schreier graph of H. The *Stallings graph* of H, written $(\Gamma(H), v_0)$, is defined to be the fragment of the Schreier graph of H spanned by the cycles at v_0 reading the geodesic representatives of the elements of H, rooted at v_0: that is, the subgraph of the Schreier graph of H consisting of all the edges participating in a cycle at v_0 which reads a geodesic representation of an element of H, and of all the vertices adjacent to these edges. In particular, a word is in H if and only if its geodesic representative labels a cycle in $\Gamma(H)$ at vertex v_0. We refer the reader to Remark 2 and to [12] for more details on these graphs. We note in particular that H has a finite Stallings graph if and only if it is finitely generated, and that $\Gamma(H)$ is efficiently algorithmically computable if H is given by a finite set of generators (words on the alphabet $\{a, b, b^{-1}\}$) [12, 3].

Example 1. Figure 1 shows examples of Stallings graphs. To be more precise: the graphs in Figure 1 are *labeled graphs*, meaning that their vertices are labeled by an initial segment of $\mathbb{N}$, see more details on this useful notion further down in Section 2.3. The definition of Stallings graphs does not entail labeling vertices — only designating a base vertex.

Remark 2. The definition of Stallings graphs given above is a generalization of that introduced by Stallings in 1983 [22] for finitely generated subgroups of free groups, and a particular instance of the definition first introduced by Gitik [9] in 1996 under the name of geodesic core, and systematized by Kharlampovich, Miasnikov and Weil [12] in 2017. Given a finitely presented group $G = \langle A \mid R \rangle$, a language L of representatives for G (a set of words over the alphabet $A \cup A^{-1}$) and a subgroup H, one considers the fragment of the Schreier graph of H spanned by the L-representatives of the elements of H. It is effectively computable if G is equipped with an automatic structure [12] and H is L-quasi-convex. In the particular case where $G = \mathsf{PSL}_2(\mathbb{Z})$, we take L to be the language of geodesics. It is well-known that G is automatic with respect to this language, and that

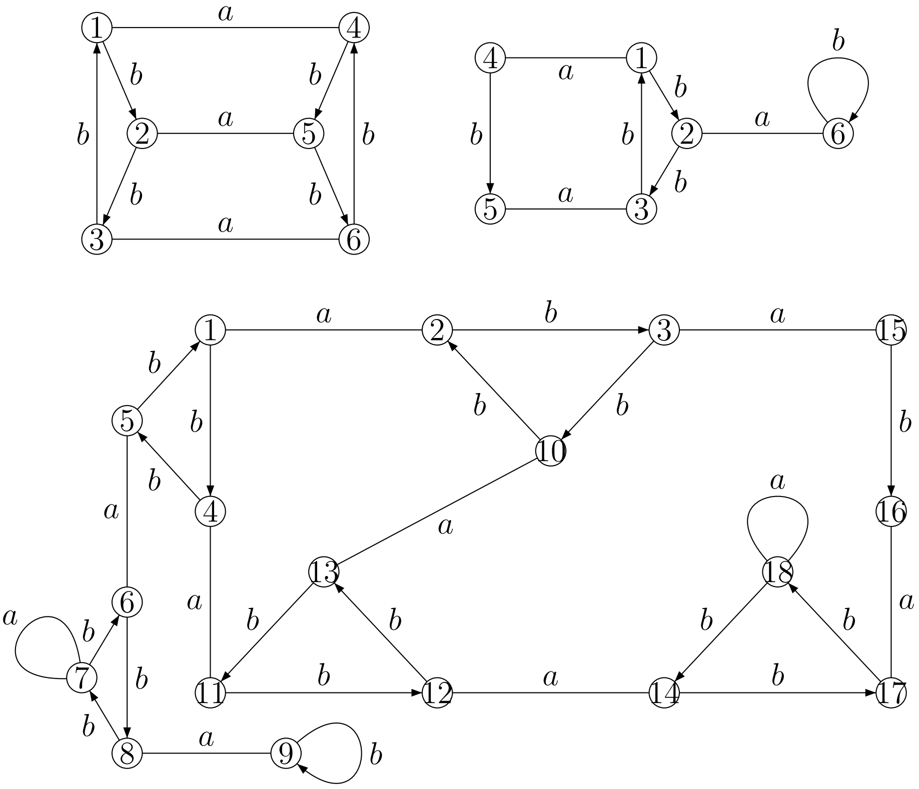

Figure 1: Top: the Stallings graphs of the subgroups $H = \langle abab^{-1}, babab\rangle$ and $K = \langle abab, b^{ab^{-1}}\rangle$ of $\mathsf{PSL}_2(\mathbb{Z})$, where g^h stands for $h^{-1}gh$. Bottom: the Stallings graph of $L = \langle b^{ab^{-1}ab}, a^{bab}, a^{(b^{-1}a)^3}, ab^{-1}abab^{-1}, (ab)^2(ab^{-1})^3\rangle$. In each case, the root is the vertex labeled 1.

every finitely generated subgroup of $\mathsf{PSL}_2(\mathbb{Z})$ is quasi-convex. The algorithm to compute the Stallings graph of a subgroup, given a tuple of its generators, is quite straightforward, we refer the reader to [3] for an outline.

It is immediate from the definition of Stallings graphs that $\Gamma(H)$ is connected and that its a-edges (respectively, b-edges) form a partial, injective map on the vertex set of the graph. Moreover, because $a^2 = b^3 = 1$, distinct a-edges are never adjacent to the same vertex: we distinguish therefore a-loops and so-called *isolated a-edges*. Similarly, if we have two consecutive b-edges, say, from v_1 to v_2 and from v_2 to v_3, then $\Gamma(H)$ also has a b-edge from v_3 to v_1. Thus each b-edge is either a loop, or an *isolated b-edge*, or a part of a b-triangle. Finally, every vertex except maybe the root vertex is adjacent to an a- and to a b-edge.

A rooted edge-labeled graph satisfying these conditions is called $\mathsf{PSL}_2(\mathbb{Z})$*-reduced* and it is not difficult to see that every finite $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph is the Stallings graph of a unique finitely generated subgroup of $\mathsf{PSL}_2(\mathbb{Z})$. That is, the mapping $H \mapsto (\Gamma(H), v_0)$ is a bijection between finitely generated subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ and $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs.

An edge-labeled graph is said to be $\mathsf{PSL}_2(\mathbb{Z})$*-cyclically reduced* if every vertex is adjacent to an a- and a b-edge or, equivalently, if it is $\mathsf{PSL}_2(\mathbb{Z})$-reduced when rooted at every one of its vertices. We also say that a finitely generated subgroup of $\mathsf{PSL}_2(\mathbb{Z})$ is

$\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced if its Stallings graph is.

Note that these two definitions, of $\mathsf{PSL}_2(\mathbb{Z})$-reducedness and $\mathsf{PSL}_2(\mathbb{Z})$-cyclic reducedness, are differently typed: $\mathsf{PSL}_2(\mathbb{Z})$-reducedness applies to rooted graphs, while the notion of $\mathsf{PSL}_2(\mathbb{Z})$-cyclic reducedness is for unrooted graphs.

Example 3. The $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs Γ with 1 or 2 vertices are represented in Figure 2.

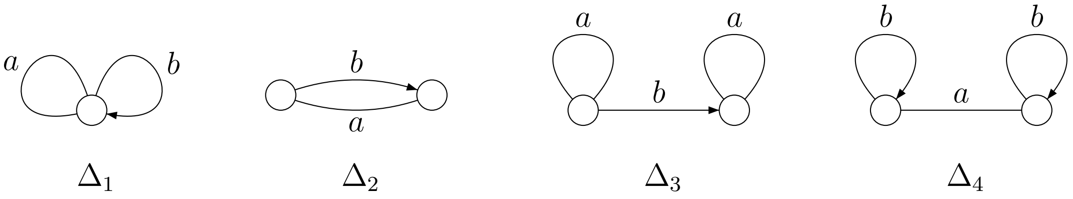

Figure 2: All $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs with at most 2 vertices.

There is only one with 1 vertex, and three with 2 vertices.

2.2 Combinatorial type, isomorphism type of a subgroup of $\mathsf{PSL}_2(\mathbb{Z})$

The *combinatorial type* of a $\mathsf{PSL}_2(\mathbb{Z})$-reduced rooted graph Γ is the tuple $(n, k_2, k_3, \ell_2, \ell_3)$ where n is the number of vertices of Γ, k_2 and k_3 are the numbers of isolated a- and b-edges, and ℓ_2 and ℓ_3 are the numbers of a- and b-loops. If Γ is a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced (unrooted) graph, it has the same combinatorial type wherever we root it, and we loosely talk of the combinatorial type of Γ. We also sometimes talk of the combinatorial type of a subgroup to mean the combinatorial type of its Stallings graph, and we refer to n (the number of vertices) as the *size* of the graph or even the *size* of the subgroup. See [3] for a discussion of the possible combinatorial types.

Let us also record the following results (see, *e.g.*, [3, Lemma 2.3, Propositions 2.7, 2.9, 8.18 and Section 8.2]).

Proposition 4. *A subgroup $H \leqslant \mathsf{PSL}_2(\mathbb{Z})$ has finite index n if and only if its Stallings graph is $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced and has combinatorial type of the form $(n, k_2, 0, \ell_2, \ell_3)$. It is free if and only if its combinatorial type is of the form $(n, k_2, k_3, 0, 0)$.*

Free $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced subgroups have even size. Free and finite index subgroups are $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced and their size is a multiple of 6. The combinatorial type of a free and finite index subgroup is of the form $(n, \frac{1}{2}n, 0, 0, 0)$.

$\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced subgroups are conjugates if and only if the (unrooted) edge-labeled graphs underlying their Stallings graphs are isomorphic.

By Kurosh's classical theorem on subgroups of free groups (*e.g.*, [15, Proposition III.3.6]), a subgroup H of $\mathsf{PSL}_2(\mathbb{Z})$ is isomorphic to a free product of r_2 copies of $\mathbb{Z}/2\mathbb{Z}$, r_3 copies of $\mathbb{Z}/3\mathbb{Z}$ and a free group of rank r, for some non-negative integers r_2, r_3, r. The triple (r_2, r_3, r), which characterizes H up to isomorphism (but not up to an automorphism of $\mathsf{PSL}_2(\mathbb{Z})$) is called the *isomorphism type* of H. We record the following connection between the combinatorial and the isomorphism types of a subgroup [3, Proposition 2.9].

Proposition 5. *Let H be a subgroup of $\mathsf{PSL}_2(\mathbb{Z})$ of size at least 2 and let $(n, k_2, k_3, \ell_2, \ell_3)$ be the combinatorial type of $\Gamma(H)$.*

If $\Gamma(H)$ is $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced, the isomorphism type of H is

$$\left(\ell_2, \ell_3, 1 + \frac{n - 2k_3 - 3\ell_2 - 4\ell_3}{6}\right).$$

If $\Gamma(H)$ is not $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced, the isomorphism type of H is

$$\left(\ell_2, \ell_3, \frac{1}{3} + \frac{n - 2k_3 - 3\ell_2 - 4\ell_3}{6}\right) \quad \textit{if the base vertex is adjacent to an a-edge}$$
$$\left(\ell_2, \ell_3, \frac{1}{2} + \frac{n - 2k_3 - 3\ell_2 - 4\ell_3}{6}\right) \quad \textit{if the base vertex is adjacent to a b-edge.}$$

Remark 6. In view of Propositions 4 and 5, we note that the size n of a finite index subgroup of isomorphism type (ℓ_2, ℓ_3, r) is $6(r-1) + 3\ell_2 + 4\ell_3$. It follows that there are only finitely many finite index subgroups of a given isomorphism type. In contrast, any isomorphism type (ℓ_2, ℓ_3, r) with $r \neq 0$ can be achieved by subgroups of infinitely many different sizes.

2.3 Labeled graphs

One of our objectives in this paper is to count subgroups by isomorphism type or by combinatorial type. Since subgroups are in bijection with $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs (their Stallings graphs), it is equivalent to count these graphs. For technical reasons, it is easier to count *labeled graphs*, that is, graphs whose vertex set is equipped with a (labeling) bijection onto a set of the form $[n] = \{1, \ldots, n\}$[1]. The graphs in Figure 1 are in fact labeled graphs.

Example 7. The $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs Δ_2 and Δ_3 in Example 3 admit two distinct labelings, while Δ_1 and Δ_4 have only one. We record here what we call the *preferred* labeling of Δ_2, where the b-edge goes from 1 to 2, see Figure 3.

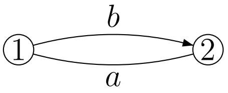

Figure 3: The preferred labeling of Δ_2

Since we are going to count graphs, rooted or not, labeled or not, it is important to clarify that we consider these combinatorial objects up to isomorphism. Concretely, if Γ and Γ' are graphs, an isomorphism from Γ to Γ' is a pair of bijections $\varphi = (\varphi_V, \varphi_E)$ from

[1] It is important to distinguish this notion of labeling, which injectively assigns an integer to each vertex, from the edge labeling used so far, where each edge is labeled by either the order 2 generator a of $\mathsf{PSL}_2(\mathbb{Z})$, or by its order 3 generator b and each path is labeled by a word.

the vertex set of Γ to the vertex set of Γ' and from the edge set of Γ to the edge set of Γ', respectively, which preserve the incidence relation (that is, if Γ has an edge e from vertex v to vertex w, then $\varphi_E(e)$ is an edge from $\varphi_V(v)$ to $\varphi_V(w)$. If Γ and Γ' are rooted, then φ_V must also map the root of Γ to the root of Γ'. If, finally, Γ and Γ' are edge-labeled, then φ_E must also preserve these labels.

It is important to note that, an n-vertex $\mathsf{PSL}_2(\mathbb{Z})$-reduced rooted graph admits exactly $n!$ distinct labeling functions. Let indeed v_0 be the root of Γ and let us fix a total order on the alphabet $\{a, b, b^{-1}\}$. Assigning to each vertex v the lexicographically least geodesic word labeling a path from v_0 to v, yields a total order on the vertex set of Γ. A labeling of Γ is therefore equivalent to a permutation of $[n]$. Another way of formulating this observation is that a $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph admits no non-trivial automorphism.

3 The silhouetting operation on $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs

We will see in Sections 4.1 and 5 that counting and randomly generating subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ reduces to counting and randomly generating labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs. Before we embark on this task, we introduce a combinatorial construction on this class of graphs.

More precisely, we define in Section 3.1 certain moves on a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph, depending on its geometry. They are used in Section 4.2 to count subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ and in Section 5 to randomly generate them. They also bring to the fore an interesting structure associated with a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph, which we call its *silhouette*. Some of its algebraic and combinatorial properties are discussed in Section 3.3.

Very roughly speaking, these moves "simplify" a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph by first iteratively removing all loops and the paths that lead to them, until we are left (except in degenerate cases) with a graph which consists only of b-triangles and paths connecting them. The process then "simplifies" these connecting paths so that the resulting graph consists only of b-triangles connected by isolated a-edges. As we know, such graphs represent conjugacy classes of free finite index subgroups (see Proposition 4).

For technical reasons, we use the notion of *weakly labeled* graphs [6, Definition II.1]: if Γ is a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of size m, a *weak labeling* of Γ is an injective map from the vertex set of Γ to $[n]$, where n is an integer at least equal to m. To lighten up notation, we often abusively identify the vertices of a weakly labeled graph with their labels. We also abusively write Δ_i $(i = 1, 2, 3, 4)$ for any weakly labeled version of the graphs in Example 3.

Observe that a weak labeling α of Γ gives rise to a labeling of Γ by a uniquely defined order-preserving bijection from the range of α to $[m]$. The labeled graph obtained this way is called the *normalization*[2] of Γ, denoted by $\mathrm{norm}(\Gamma)$.

[2] This operation is called *reduction* in [6, Section II.2.1].

3.1 Moves on a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph

Here we define so-called λ_3-, $\lambda_{2,1}$-, $\lambda_{2,2}$-, κ_3- and *exceptional* moves on weakly labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs[3]. But for the exceptional moves, each of these moves deletes vertices from the input graph without changing their label, so that the resulting graph is, again, weakly labeled.

In the following, Γ is a weakly labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$.

$\boldsymbol{\lambda_3}$-moves If Γ has a b-loop at vertex v (in fact, at the vertex labeled v) and there is an isolated a-edge between v and a distinct vertex w, then the (λ_3, v, w)*-move* consists in deleting vertex v and the adjacent edges, and adding an a-loop at vertex w. The resulting weakly labeled graph Δ (see Figure 4) is $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced and has combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_3$, where $\boldsymbol{\lambda}_3 = (-1, -1, 0, 1, -1)$.

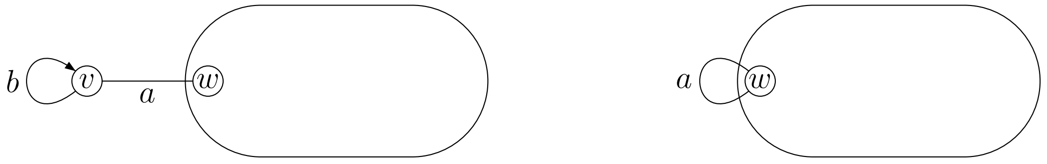

Figure 4: (λ_3, v, w)-move

Lemma 8. *Suppose that $n \geqslant 2$ and $\ell_3 > 0$. The λ_3-moves establish a bijection from the set of pairs (Γ, ℓ) with Γ a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and ℓ a b-loop in Γ, to the set of triples (Δ, ℓ', v) formed by a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph Δ with combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_3$, an a-loop ℓ' in Δ and an integer $v \in [n]$.*

Proof. Given a pair (Γ, ℓ) with Γ a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and ℓ a b-loop in Γ, we associate to it the triple (Δ, v, w) — and we write $(\Gamma, \ell) \longmapsto (\Delta, \ell', v)$ — defined as follows: v is the vertex carrying the loop ℓ in Γ; since $n > 1$, v is adjacent to an isolated a-edge and we let w be the other end of that a-edge; finally, we let Δ' be the weakly labeled graph obtained from Γ by a (λ_3, v, w)-move, $\Delta = \mathrm{norm}(\Delta')$ and ℓ' be the a-loop in Δ at the vertex labeled w in Δ'.

Conversely, let Δ be a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_3$, let ℓ' be an a-loop in Δ and let $v \in [n]$. Let Δ' be the weakly labeled graph obtained from Δ by "making space for v", that is, by incrementing the labels of all the vertices greater than or equal to v. Finally, let w be the label of the vertex of Δ' carrying the loop ℓ'. Now let Γ be the graph obtained from Δ' by deleting the loop ℓ' and adding vertex v, an isolated a-edge between v and w and a b-loop at v: it is directly verified that Γ is properly labeled, of combinatorial type $\boldsymbol{\tau}$, and that $(\Gamma, \ell) \longmapsto (\Delta, \ell', v)$. □

[3] The denomination of λ_3-move is chosen because these moves deal with loops labeled by the order 3 generator b, which are counted by the parameter ℓ_3. Similar justifications hold for the moves that deal with a-loops (counted by ℓ_2) and with isolated b-edges (counted by k_3).

$\boldsymbol{\lambda}_2$-moves Let Γ be a weakly labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs of size $n \geqslant 3$ and let v be a vertex carrying an a-loop. Two situations occur, depending on whether v sits on a b-triangle or not, giving rise to two flavors of λ_2-moves.

If v sits on a b-triangle, let w and w' be the other extremities of the b-edges ending and starting at v, respectively. Then $w \neq w'$ and Γ has a (non-isolated) b-edge from w' to w. The $(\lambda_{2,1}, v, w')$-*move* consists in removing from Γ vertex v and the adjacent edges (the a-loop ℓ and two b-edges). The resulting graph Δ (see Figure 5) is $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced, it has an isolated b-edge from w' to w and combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_{2,1}$, where $\boldsymbol{\lambda}_{2,1} = (-1, 0, 1, -1, 0)$.

If instead v does not sit on a b-triangle, there exist vertices w, w' such that v, w, w' are pairwise distinct, there is an isolated a-edge between w and w', and an isolated b-edge between v and w (two directions are possible for that edge). The $(\lambda_{2,2}, v \to w, w')$-*move* (respectively, $(\lambda_{2,2}, v \leftarrow w, w')$, depending on the orientation of the b-edge adjacent to v) consists in deleting from Γ the vertices v and w and the edges adjacent to them, and adding an a-loop ℓ' at w'. The resulting graph Δ (see Figure 5) is $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced and has combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_{2,2}$, where $\boldsymbol{\lambda}_{2,2} = (-2, -1, -1, 0, 0)$.

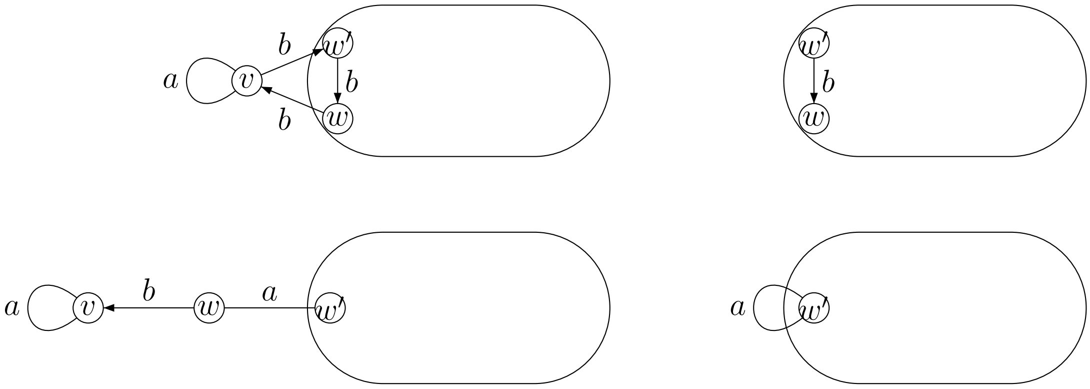

Figure 5: Above: $(\lambda_{2,1}, v, w')$-move. Below: $(\lambda_{2,2}, v \leftarrow w, w')$-move

Lemma 9. *Suppose that $n \geqslant 3$ and $\ell_2 > 0$.*

1. *The $\lambda_{2,1}$-moves establish a bijection from the set of pairs (Γ, ℓ) with Γ a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and ℓ an a-loop adjacent to a b-triangle in Γ, to the set of triples (Δ, e, v) formed by a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph Δ with combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_{2,1}$, an isolated b-edge e in Δ and an integer $v \in [n]$.*

2. *Similarly, the $\lambda_{2,2}$-moves establish a bijection from the set of pairs (Γ, ℓ) with Γ a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and ℓ an a-loop adjacent to an isolated b-edge in Γ, to the set of 4-tuples $(\Delta, \ell', v, w, \varepsilon)$ formed by a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph Δ with combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_{2,2}$, an a-loop ℓ' in Δ, distinct integers $v, w \in [n]$ and some $\varepsilon \in \{-1, +1\}$.*

Proof. Given a pair (Γ, ℓ) with Γ a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and ℓ an a-loop in Γ adjacent to a b-triangle, we associate to it the triple (Δ, e, v) — and we write $(\Gamma, \ell) \longmapsto (\Delta, e, v)$ — defined as follows: v is the vertex carrying the loop ℓ in Γ; since v is adjacent to a b-triangle and we let e be the b-edge in that triangle not adjacent to v (going from w' to w); finally, we let Δ' be the weakly labeled graph obtained from Γ by a $(\lambda_{2,1}, v, w')$-move, $\Delta = \mathrm{norm}(\Delta')$ and e be the isolated b-edge in Δ starting at the vertex labeled w' in Δ'.

Conversely, let Δ be a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_{2,1}$, let e be an isolated b-edge in Δ and let $v \in [n]$. Let Δ' be the weakly labeled graph obtained from Δ by "making space for v", that is, by incrementing the labels of all the vertices greater than or equal to v. Now let Γ be the graph obtained from Δ' by adding a new vertex v, completing e to a b-triangle through vertex v: it is directly verified that Γ is properly labeled, of combinatorial type $\boldsymbol{\tau}$, and that $(\Gamma, \ell) \longmapsto (\Delta, e, v)$. This completes the proof of the first statement.

The second statement is proved in a similar fashion. Given a pair (Γ, ℓ) with Γ a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and ℓ an a-loop adjacent to an isolated b-edge in Γ, we associate with it a 4-tuple $(\Delta, \ell', v, w, \varepsilon)$ as in the statement, where v is the vertex carrying ℓ, w is the other extremity of the adjacent isolated b-edge and ε records whether a $(\lambda_{2,2}, v \to w, w')$-move or a $(\lambda_{2,2}, v \leftarrow w, w')$ can be performed. The converse mapping, reconstructing (Γ, ℓ) from $(\Delta, \ell', v, w, \varepsilon)$ follows the same steps as for $\lambda_{2,1}$- or λ_3-moves. □

$\boldsymbol{\kappa}_3$-moves Let Γ be a weakly labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of size at least 4, and let v, w, v', w' be pairwise distinct vertices such that there is an isolated b-edge from v to w, and isolated a-edges connecting v and v' on the one hand, and w and w' on the other. The κ_3*-move* $(\kappa_3, v \to w, v', w')$ consists in deleting vertices v and w and the adjacent edges, and adding a new isolated a-edge between v' and w'. The resulting graph Δ (see Figure 6) is $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced and has combinatorial type $\boldsymbol{\tau} + \boldsymbol{\kappa}_3$, where $\boldsymbol{\kappa}_3 = (-2, -1, -1, 0, 0)$.

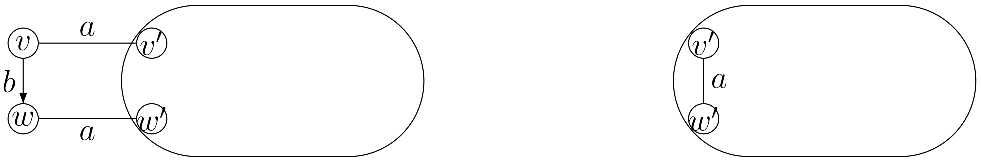

Figure 6: $(\kappa_3, v \to w, v', w')$-move

Similarly to the other moves, we record the following lemma.

Lemma 10. *Suppose that $n \geqslant 4$, $\ell_2 = 0$ and $k_3 > 0$. The κ_3-moves establish a bijection from the set of pairs (Γ, e) with Γ a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and e an isolated b-edge, to the set of triples $(\Delta, e', v, w, \varepsilon)$ formed by a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph Δ with combinatorial type $\boldsymbol{\tau} + \boldsymbol{\kappa}_3$, an isolated a-edge e' in Δ, distinct integers $v, w \in [n]$ and some $\varepsilon \in \{-1, +1\}$.*

Proof. The assumption that $\ell_2 = 0$ guarantees that every isolated b-edge is adjacent to two isolated a-edges, and the fact that $n \geqslant 4$ guarantees that these a-edges are distinct.

Now let (Γ, e) be a pair formed by a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph Γ of combinatorial type $\boldsymbol{\tau}$, and an isolated b-edge e in Γ, say, from vertex v to vertex w. Let v' and w' be the other extremities of the isolated a-edges adjacent to v and w, respectively. We associate with it the tuple $(\Delta, e', v, w, \varepsilon)$ — and we write $(\Gamma, e) \longmapsto (\Delta, e', v, w, \varepsilon)$ — where Δ' is the weakly labeled graph obtained from Γ by a $(\kappa_3, v \to w, v', w')$-move, $\Delta = \mathrm{norm}(\Delta')$, e' is the isolated a-edge in Δ adjacent to the vertex labeled v' in Δ', $\varepsilon = 1$ if $v' < w'$ and $\varepsilon = -1$ if $v' > w'$.

Conversely, let Δ be a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $\boldsymbol{\tau} + \boldsymbol{\kappa}_3$, let e' be an isolated a-edge in Δ, connecting vertices v' and w', and let v, w be distinct integers in $[n]$. Let Δ' be the weakly labeled graph obtained from Δ by "making space for v, w", that is, by incrementing the labels of all the vertices greater than or equal to $\max(v, w) - 1$ by 2 units, and those in $[\min(v, w), \max(v, w) - 2]$ by 1 unit. Let then Γ be the graph obtained from Δ' by deleting the a-edge e'; adding new vertices v, w and a b-edge from v to w; and adding a-edges between v and $\min(v', w')$ and between w and $\max(v', w')$ if $\varepsilon = 1$ — between v and $\max(v', w')$ and between w and $\min(v', w')$ if $\varepsilon = -1$. It is directly verified that Γ is properly labeled, of combinatorial type $\boldsymbol{\tau}$, and that $(\Gamma, e) \longmapsto (\Delta, e', v, w, \varepsilon)$. □

Exceptional moves Finally, we introduce three so-called *exceptional* moves. The first can be applied only to a weakly labeled version of the 1-vertex graph Δ_1 that does not use label 1, turning it into Δ_1 properly labeled.

The second can be applied only to a weakly labeled version of Δ_3, turning it into Δ_1 (with its only vertex labeled 1). This move can be seen as a degenerate version of a $\lambda_{2,2}$-move. Note that it modifies the combinatorial type by the addition of $\boldsymbol{exc} = (-1, 0, -1, -1, 1)$, the difference between the combinatorial types of Δ_1 and Δ_3.

The last exceptional move can be applied to any weakly labeled version of Δ_2 different from the so-called preferred labeling (see Example 7), turning it to that preferred labeling.

3.2 Silhouette graphs

We can see the moves described in Section 3.1 as a rewriting system on weakly labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs, which we show to be confluent (Section 3.3 below). The following definition will be convenient: we say that a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph Γ (weakly labeled or not) is a *silhouette graph* if it is equal to Δ_1 or Δ_2, or if it has combinatorial type $(n, n/2, 0, 0, 0)$ (where n is a positive multiple of 6, see Proposition 4). Observe that no move is defined on a silhouette graph of size at least 3, whichever way it is weakly labeled.

Silhouette graphs play a foundational role in the recursive process for the random generation of subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ described in Section 5. They also play a central role in [4].

3.3 Silhouetting a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph

In general, several moves can be applied to a weakly labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph Γ. Our next proposition states that, however, the end result of a maximal sequence of moves is independent of the choice of that maximal sequence.

Proposition 11. *Let Γ be a weakly labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph Γ. If Δ and Δ' are weakly labeled graphs obtained from Γ, respectively, after maximal sequences of λ_3-, $\lambda_{2,1}$-, $\lambda_{2,2}$- and κ_3- and exceptional moves, then $\Delta = \Delta'$.*

Proof. We proceed by induction on the number of vertices of Γ. The result is immediate if no move is possible on Γ. If Γ has 1 or 2 vertices, either no move is possible, or only one exceptional move is possible, and the result is again immediate.

Suppose now that Γ has $n \geqslant 3$ vertices and that the sequences of moves leading from Γ to Δ and Δ' start with the same move (the same type of move, with the same parameters), taking Γ to Γ'. Since Γ' is a weakly labeled graph with less than n vertices, the announced result holds by induction.

Finally, suppose that the first moves from Γ to Δ and Γ to Δ', say m and m', are distinct. Note that since Γ has size at least 3, neither m nor m' is an exceptional move. Let Γ_1 (resp. Γ'_1) be the weakly labeled graph obtained from Γ after the move m (resp. m'), so that there exists a maximal sequence of moves from Γ_1 to Δ (resp. Γ'_1 to Δ'). Let us consider the possible values of m and m'.

If $m = (\lambda_3, v, w)$ and $m' = (\lambda_3, v', w')$, there are two possibilities. If Γ is a weakly labeled version of Δ_4 (so that $v = w'$ and $v' = w$), then Γ_1 and Γ'_1 are weakly labeled versions of Δ_1, so $\Delta = \Gamma_1$, $\Delta' = \Gamma'_1$ and their normalizations are equal. If instead Γ is not a weakly labeled version of Δ_4, then v, v', w, w' are pairwise distinct, and the moves m and m' *commute* in the following sense: an m'-move is possible on Γ_1, leading to a weakly labeled graph Γ_2; an m-move is possible on Γ'_1, leading to a weakly labeled graph Γ'_2, and $\Gamma_2 = \Gamma'_2$. Let Δ'' be the graph obtained from Γ_2 by a maximal sequence of moves. Since Γ_1 and Γ'_1 are weakly labeled graphs with $n-1$ vertices, the induction hypothesis shows that $\Delta = \Delta''$ and $\Delta' = \Delta''$, so that $\Delta = \Delta'$.

We now verify that, similarly, other combinations of first moves m and m' also commute, leading to the same conclusion that the statement in the proposition holds, except in a few degenerate cases that yield the same conclusion by other arguments.

It is readily verified that any λ_3-move commutes with any $\lambda_{2,1}$-move.

If $m = (\lambda_3, v, w)$ and $m' = (\lambda_{2,2}, x \to y, y')$ (or $m' = (\lambda_{2,2}, x \leftarrow y, y')$), we again distinguish two cases. If $y' = v$, then $y = w$ and Γ consists of exactly three states x, y, y', with a b-edge between x and y, an a-edge between y and y', an a-loop at state x and a b-loop at state y', see Figure 7. Any maximal sequence of moves from Γ leads to Δ_1,

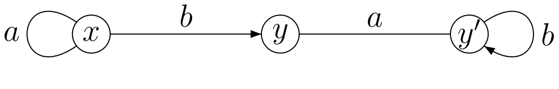

Figure 7: Case where a λ_3- and a $\lambda_{2,2}$-moves are possible

where the only vertex is labeled 1, and the announced statement holds. If instead $y' \neq v$, then Γ has more than three states and the moves m and m' commute.

If $m = (\lambda_3, v, w)$ and $m' = (\kappa_3, x \to y, x', y')$ (or $m' = (\kappa_3, x \leftarrow y, x', y')$), there are again two possibilities. If $v \neq x', y'$, then m and m' modify disjoint parts of Γ and they clearly commute. If instead $v = x'$, then $w = x$ — or if, symmetrically, $v = y'$ and $w = y$ —, see Figure 8, then a direct verification shows the following: m can be followed by a $(\lambda_{2,2}, w \to y, y')$-move (or a $(\lambda_{2,2}, w \leftarrow y, y')$-move, as the case may be), leading to a graph where vertices v, w, y have been deleted and y' carries an a-loop; and m' can be followed by a (λ_3, v, y')-move, leading to that same graph.

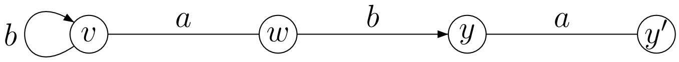

Figure 8: Case where a λ_3- and a κ_3-moves are possible

The only situation where two λ_2-moves do not commute is when they are both $\lambda_{2,1}$-moves or both $\lambda_{2,2}$-moves, modifying overlapping parts of Γ.

The first case arises if two a-loops sit on the same b-triangle, so that $m = (\lambda_{2,1}, v, w)$ and $m' = (\lambda_{2,1}, v', w)$ are possible, see Figure 9. If w also carries an a-loop (so that Γ has 3 vertices), then $\Delta = \Delta' = \Delta_1$. Otherwise, an isolated a-edge connects w and a vertex w', distinct from v, v', w, a $(\lambda_{2,2}, v' \to w, w)$-move is possible (or some other orientation of a b-edge between v' and w) after carrying out the m-move and, together these two moves amount to deleting vertices v, v' and w, and adding an a-loop at w'.

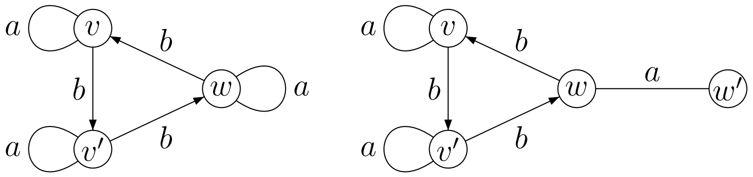

Figure 9: Cases where $\lambda_{2,1}$-moves interfere with each other

In the second case, we have, say, $m = (\lambda_{2,2}, x \to y, z)$ and $m' = (\lambda_{2,2}, x' \to y', z')$ (or any other combination of directions for the b-edges between x and y on one hand, and x' and y' on the other) satisfying $y' = z$ and $z' = y$. Then Γ has exactly 4 vertices, see Figure 10, and applying either move to Γ yields a weakly labeled version of Δ_3, on which one can only apply an exceptional move. It follows that $\Delta = \Delta' = \Delta_1$.

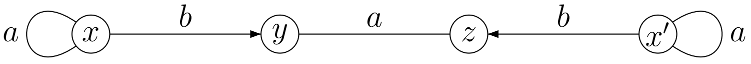

Figure 10: Case where $\lambda_{2,2}$-moves interfere with each other

It is directly verified that any $\lambda_{2,1}$-move commutes with any κ_3-move. Consider now the case where $m = (\lambda_{2,2}, v \to w, w')$ and $m' = (\kappa_3, x \to y, x', y')$ (or any other combination of directions for the b-edges between v and w, and between x and y). If $w \neq x'$ (and hence $w' \neq x$), then m and m' modify disjoint parts of Γ and clearly commute. If instead $w = x'$ and $w' = x$ (see Figure 11), a direct verification shows that after applying either m or m', a $\lambda_{2,2}$-move can be applied (a $(\lambda_{2,2}, x \to y, y')$-move can be applied to Γ_1 and a $(\lambda_{2,2}, v \to w, y')$-move to Γ'_1), leading to the same weakly labeled graph with an a-loop at vertex y': again m and m' commute.

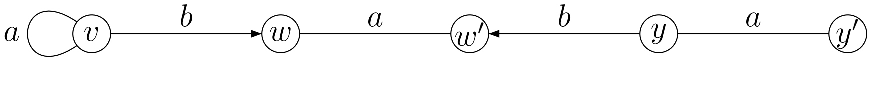

Figure 11: Case where a $\lambda_{2,2}$- and a κ_3-moves are possible

Similarly, suppose that $m = (\kappa_3, v \to w, v', w')$ and $m' = (\kappa_3, x \to y, x', y')$ (or any other combination of directions for the b-edges). We distinguish three cases. If $x, x', y, y' \notin \{v, v', w, w'\}$, then m and m' clearly commute. If $(w, w') = (x', x)$ and $y \neq v'$ (see Figure 12), then a $(\kappa_3, x \to y, v', y')$-move can be applied to Γ_1 and a $(\kappa_3, v \to w, v', y')$-move to Γ'_1, leading to the same weakly labeled graph (with an a-edge between v' and y', and no vertices labeled x, y, v, w). If now $(w, w') = (x', x)$ and $y = v'$ (so that $y' = v$), then Γ has exactly four vertices, both Γ_1 and Γ'_1 are weak labelings of Δ_2, on which only an exceptional move is defined, so that $\Delta = \Delta'$.

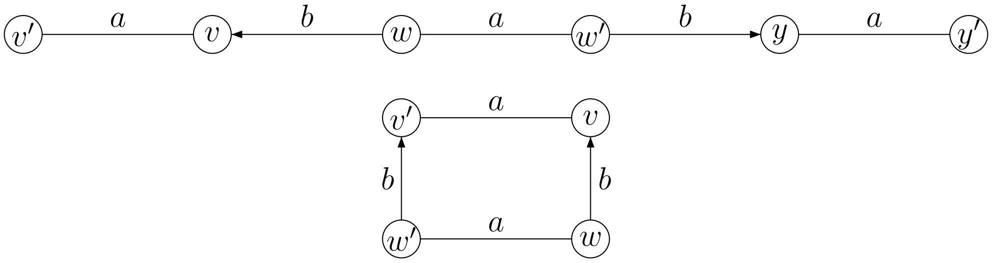

Figure 12: Cases where two κ_3-moves are possible

This concludes the proof of the proposition. □

If Γ is a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph, we define the *silhouette* $\mathsf{silh}(\Gamma)$ of Γ to be the labeled graph resulting from the application of a maximal sequence of moves, followed by a normalization: Proposition 11 guarantees that $\mathsf{silh}(\Gamma)$ is well defined.

Example 12. Consider the three (labeled) Stallings graphs in Figure 1. The first is equal to its own silhouette, and is also equal to the silhouette of the third. The silhouette of the second graph is Δ_2 (with its preferred labeling).

The silhouetting operation preserves some important algebraic information about a subgroup, namely the free rank component of its isomorphism type.

Proposition 13. *Let H be a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced subgroup of $\mathsf{PSL}_2(\mathbb{Z})$, with Stallings graph Γ and isomorphism type (ℓ_2, ℓ_3, r). Then $\mathsf{silh}(\Gamma) = \Delta_1$ if and only if $r = 0$, and $\mathsf{silh}(\Gamma) = \Delta_2$ if and only if $r = 1$. In all other cases, $r \geqslant 2$ and $\mathsf{silh}(\Gamma)$ has isomorphism type $(0, 0, r)$.*

Proof. Let $\boldsymbol{\tau}$ be the combinatorial type of Γ. Proposition 5 shows that the free rank r in the isomorphism type of H is a function of $\boldsymbol{\tau}$; more precisely, if $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$, then $6(r-1) = n - 2k_3 - 3\ell_2 - 4\ell_3 = \varphi(\boldsymbol{\tau})$, and we observe that φ is a linear map.

By construction, $\mathsf{silh}(\Gamma)$ is obtained from Γ by a succession of λ_3-, $\lambda_{2,1}$-, $\lambda_{2,2}$-, κ_3-moves and maybe one exceptional move (followed by normalization). Each of these moves modifies the combinatorial type by adding to it the vector $\boldsymbol{\lambda}_3$, $\boldsymbol{\lambda}_{2,1}$, $\boldsymbol{\lambda}_{2,2}$, $\boldsymbol{\kappa}_3$ or $\boldsymbol{exc}$. Every one of these vectors lies in the kernel of φ, so the free rank component of the isomorphism types of Γ and $\mathsf{silh}(\Gamma)$ coincide.

It is immediate that this free rank component is 0 for Δ_1, 1 for Δ_2 and $1 + n/6 \geqslant 2$ for each silhouette graph of size $n > 2$. The proposition follows immediately. □

4 Counting subgroups by isomorphism and by combinatorial type

Our aim in this section is to count subgroups of a given size, under some additional constraint: with a fixed isomorphism type or with a fixed combinatorial type. Since each subgroup is uniquely represented by its Stallings graph, *i.e.*, by a $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph, this is equivalent to counting these graphs (up to isomorphism).

As noted in Section 2.3, an n-vertex $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph admits exactly $n!$ distinct labelings. As a result, our strategy to count n-vertex $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs will be to count labeled n-vertex $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs, and then divide that count by $n!$. The same applies for the counting of n-vertex $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs of a particular combinatorial type, or for n-vertex rooted $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs. Note that there is no such easy correlation between the number of labeled and unlabeled (non-rooted) cyclically reduced graphs, as counting is perturbed by the existence of symmetries (automorphisms).

Thus our task reduces to counting labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs. It further reduces to counting labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs, as we explain below.

4.1 Reduction to the count of labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs

If $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ is a tuple of integers, we let $H(\boldsymbol{\tau})$ (respectively, $L(\boldsymbol{\tau})$, $s(\boldsymbol{\tau})$) be the number of subgroups (respectively, labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs, labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs) of combinatorial type $\boldsymbol{\tau}$.

Example 14. In view of Example 3, the non-zero values of H, L and s for tuples $(n, k_2, k_3, \ell_2, \ell_3)$ where $n = 1, 2$ are as follows:

- $H(\boldsymbol{\tau}) = L(\boldsymbol{\tau}) = 1$ for $\boldsymbol{\tau} = (1,0,0,0,0), (1,0,0,1,1), (1,0,0,1,0), (1,0,0,0,1)$ and $s(\boldsymbol{\tau}) = 1$ for $\boldsymbol{\tau} = (1,0,0,1,1)$;
- $L(\boldsymbol{\tau}) = 4$ and $H(\boldsymbol{\tau}) = s(\boldsymbol{\tau}) = 2$ for $\boldsymbol{\tau} = (2,1,1,0,0), (2,0,1,2,0)$;
- $L(\boldsymbol{\tau}) = 2$ and $H(\boldsymbol{\tau}) = s(\boldsymbol{\tau}) = 1$ for $\boldsymbol{\tau} = (2,1,0,0,2)$;
- $L(\boldsymbol{\tau}) = 2$ and $H(\boldsymbol{\tau}) = 1$ for $\boldsymbol{\tau} = (2,0,1,1,0), (2,1,0,0,1)$;

We first establish the connection between the parameters $H(\boldsymbol{\tau})$, $L(\boldsymbol{\tau})$ and $s(\boldsymbol{\tau})$.

Proposition 15. *Let $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ be a combinatorial type with $n \geqslant 2$. The numbers $H(\boldsymbol{\tau})$, $L(\boldsymbol{\tau})$ and $s(\boldsymbol{\tau})$, respectively of subgroups, labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs and labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs of combinatorial type $\boldsymbol{\tau}$ are related as follows.*

$$L(\boldsymbol{\tau}) = n \cdot s(n, k_2, k_3, \ell_2, \ell_3) + (\ell_2 + 1) \cdot s(n, k_2, k_3, \ell_2 + 1, \ell_3) + (\ell_3 + 1) \cdot s(n, k_2, k_3, \ell_2, \ell_3 + 1)$$

$$H(\boldsymbol{\tau}) = \frac{1}{n!} L(\boldsymbol{\tau}).$$

Proof. Let (Γ, v) be a $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph with $n \geqslant 2$ vertices, such that Γ is not $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced. Then v is adjacent to an a-edge but no b-edge, or the opposite. Adding a b-loop at v in the first case, an a-loop in the second case, yields a rooted $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph (Γ', v). Conversely, if Γ' is $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced, we get $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs either by rooting Γ' at any one of its vertices, or by rooting Γ' at a vertex that carries a loop and deleting that loop. The first equality follows directly.

The second equality follows from the first since a size n $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph has $n!$ distinct labelings (see Section 2.3). □

Based on Proposition 5, which relates the isomorphism type and the combinatorial type of a subgroup, we get the following statement.

Proposition 16. *Let $\boldsymbol{\sigma} = (\ell_2, \ell_3, r)$ be an isomorphism type and let $k_2 = \frac{1}{2}(n - \ell_2)$.*

The number of $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced subgroups of size n and isomorphism type $\boldsymbol{\sigma}$ is $n \cdot s(n, k_2, k_3, \ell_2, \ell_3)$, where $k_3 = \frac{1}{2}(n - 3\ell_2 - 4\ell_3 - 6r + 6)$.

The number of non-$\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced subgroups of size n and isomorphism type $\boldsymbol{\sigma}$, where the base vertex is adjacent to an a-edge, is $(\ell_3 + 1) \cdot s(n, k_2, k_3', \ell_2, \ell_3 + 1)$, where $k_3' = \frac{1}{2}(n - 3\ell_2 - 4\ell_3 - 6r + 2)$.

The number of non-$\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced subgroups of size n and isomorphism type $\boldsymbol{\sigma}$, where the base vertex is adjacent to a b-edge, is $(\ell_2 + 1) \cdot s(n, k_2, k_3'', \ell_2 + 1, \ell_3)$, where $k_3'' = \frac{1}{2}(n - 3\ell_2 - 4\ell_3 - 6r + 4)$.

Propositions 15 and 16 effectively reduce the counting of subgroups to the counting of labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs of a given combinatorial type, which is investigated in Section 4.2 below.

4.2 Counting labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs

Let $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ be a combinatorial type. We give (multi-)recurrence relations to compute $s(\boldsymbol{\tau})$ when $n > 2$. For $n \leqslant 2$, see Example 14.

The bijections established in Lemmas 8, 9 and 10 show the following.

Proposition 17. *Let $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ be a combinatorial type such that $n \geqslant 2$ and $\ell_3 > 0$. Let Δ be a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau} + \boldsymbol{\lambda}_3$. Then the set of labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs Γ of combinatorial type $\boldsymbol{\tau}$, such that* *a λ_3-move takes Γ to Δ, has $\frac{n\cdot(\ell_2+1)}{\ell_3}$ elements. More generally,*

$$\begin{aligned} s(\boldsymbol{\tau}) &= \frac{n\cdot(\ell_2+1)}{\ell_3}\ s(\boldsymbol{\tau}+\boldsymbol{\lambda}_3),\ \textit{that is:} \\ s(n,k_2,k_3,\ell_2,\ell_3) &= \frac{n\cdot(\ell_2+1)}{\ell_3}\ s(n-1,k_2-1,k_3,\ell_2+1,\ell_3-1). \end{aligned} \tag{1}$$

Proposition 18. *Let $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ be a combinatorial type such that $n \geqslant 3$ and $\ell_2 > 0$. Let Δ be a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}+\boldsymbol{\lambda}_{2,1}$ (resp. $\boldsymbol{\tau}+\boldsymbol{\lambda}_{2,2}$). Then the set of labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs Γ of combinatorial type $\boldsymbol{\tau}$, such that a $\lambda_{2,1}$-move (resp. a $\lambda_{2,2}$-move) takes Γ to Δ, has $\frac{n\cdot(k_3+1)}{\ell_2}$ (resp. $2n\cdot(n-1)$) elements. More generally,*

$$\begin{aligned} s(\boldsymbol{\tau}) &= \frac{n\cdot(k_3+1)}{\ell_2}\ s(\boldsymbol{\tau}+\boldsymbol{\lambda}_{2,1}) \\ &\quad + 2n\cdot(n-1)\ s(\boldsymbol{\tau}+\boldsymbol{\lambda}_{2,2}),\ \textit{that is:} \\ s(n,k_2,k_3,\ell_2,\ell_3) &= \frac{n\cdot(k_3+1)}{\ell_2}\ s(n-1,k_2,k_3+1,\ell_2-1,\ell_3) \\ &\quad + 2n\cdot(n-1)\ s(n-2,k_2-1,k_3-1,\ell_2,\ell_3). \end{aligned} \tag{2}$$

Proposition 19. *Let $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ be a combinatorial type such that $n \geqslant 4$, $\ell_2 = 0$ and $k_3 > 0$. Let Δ be a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}+\boldsymbol{\kappa}_3$. Then the set of labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs Γ of combinatorial type $\boldsymbol{\tau}$, such that a κ_3-move takes Γ to Δ, has $\frac{2n\cdot(n-1)(k_2-1)}{k_3}$ elements. More generally, if $\ell_2 = 0$, we have*

$$\begin{aligned} s(\boldsymbol{\tau}) &= 2\ \frac{n\cdot(n-1)(k_2-1)}{k_3}\ s(\boldsymbol{\tau}+\boldsymbol{\kappa}_3),\ \textit{that is:} \\ s(n,k_2,k_3,0,\ell_3) &= 2\ \frac{n\cdot(n-1)(k_2-1)}{k_3}\ s(n-2,k_2-1,k_3-1,0,\ell_3). \end{aligned} \tag{3}$$

We can use Equations (1), (2) and (3) to compute the coefficient $s(n, k_2, k_3, \ell_2, \ell_3)$, where $n \geqslant 3$: if one of k_3, ℓ_2 or ℓ_3 is greater than zero, we can apply at least one of these equations, thus reducing the first argument of the coefficients to compute by 1 or 2.

More precisely, one may first iterate the use of Equation (1) until $n \leqslant 2$ or $\ell_3 = 0$. One can then use repeatedly Equation (2), thus reducing the computation of $s(\boldsymbol{\tau})$ to the computation of a number of smaller coefficients, until $n \leqslant 2$ or $\ell_2 = 0$ (note that Equation (2) never increases ℓ_3). Finally, if $n \geqslant 3$ and $\ell_2 = \ell_3 = 0$, then in fact $n \geqslant 4$ and one can use repeatedly Equation (3) until $n \leqslant 2$ or $k_3 = \ell_2 = \ell_3 = 0$. The computation of the coefficients when $n \leqslant 2$ was done in Example 14. As for the coefficients of the form $s(n, k_2, 0, 0, 0)$ ($n > 2$), we note that they count the size n labeled silhouette graphs.

The latter numbers were computed in [3, Appendices A.3 and A.4] (see also the computation by Stothers [24] of the number of finite index, free subgroups of $\mathsf{PSL}_2(\mathbb{Z})$, that is, of subgroups whose Stallings graph is a silhouette graph of size at least 3).

Proposition 20. *Let t_2 (respectively, t_3) be given, for $n \geqslant 1$, by*[4]

$$t_2(2n) = \frac{(2n)!}{2^n\, n!} = \prod_{1 \leqslant i \leqslant n} (2i-1), \quad \text{and } t_3(3n) = \frac{(3n)!}{3^n\, n!} = \prod_{1 \leqslant i \leqslant n} (3i-1)(3i-2).$$

Then the number $s(6n, 3n, 0, 0, 0)$ of size $6n$ labeled silhouette graphs ($n \geqslant 1$) satisfies the following recurrence relation:

$$s(6n, 3n, 0, 0, 0) = t_2(6n)\, t_3(6n) - \sum_{m=1}^{n-1} t_2(6m)\, t_3(6m)\, s\big(6(n-m), 3(n-m), 0, 0, 0\big).$$

5 Random generation of subgroups of $\mathsf{PSL}_2(\mathbb{Z})$

Our objective in this section is to produce an algorithm which generates uniformly at random subgroups of $\mathsf{PSL}_2(\mathbb{Z})$ with a given size and isomorphism type.

As we saw in Example 14, there are exactly four size 1 subgroups, with pairwise distinct combinatorial and isomorphism type: the trivial subgroup, the subgroups generated by a and b, respectively, and $\mathsf{PSL}_2(\mathbb{Z})$ itself. We now concentrate on generating subgroups of size at least 2, and we assume that the parameters $L(\boldsymbol{\tau})$ and $s(\boldsymbol{\tau})$ have been pre-computed for all types of sufficient size.

Like in Section 4, generating uniformly at random a subgroup of a given combinatorial or isomorphism type reduces to randomly generating a labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph of a given combinatorial type and, before that, to randomly generating a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of a given type. Indeed, the label-forgetting map, from the set of labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs of combinatorial type $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ to the set of $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs of type $\boldsymbol{\tau}$, is such that the inverse image of each $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph of type $\boldsymbol{\tau}$ contains exactly $n!$ elements (see the discussion at the end of Section 2.3).

As we saw in Proposition 5, the isomorphism class of a $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph is determined by its combinatorial type, and a given size and isomorphism type arises for a finite number of combinatorial types only. As a result, we only need to randomly generate a $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph with a given combinatorial type, and this starts with randomly generating a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs of a given combinatorial type.

We first deal with the particular case of labeled silhouette graphs, then proceed to the general case of labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs and, finally, to labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs.

5.1 Random labeled silhouette graphs

If s, t are permuations of $[n]$, we denote by $\Gamma(s,t)$ the labeled graph with vertex set $[n]$ and, for each $i \in [n]$, an a-labeled edge from i to $s(i)$ and a b-labeled edge from i to $t(i)$.

[4] What is written $t_2(2n)$ (respectively, $t_3(3n)$, $s(6n, 3n, 0, 0)$) here, is written $t_2^{(0)}(2n)$ (respectively, $t_3^{\mathsf{fr\text{-}fi}}(3n)$, $g_{\mathsf{pr}}^{\mathsf{fr\text{-}fi}}(6n)$) in [3].

Let n be a positive multiple of 6. The procedure `random_silhouette_graph`(n) to generate a size n labeled silhouette graph, summarized below, is well known (see [3] for instance). If s is a permutation on n elements, we denote by $\mathsf{shuffle}(s)$ the permutation $t^{-1}st$ where t is a random permutation on n elements.

```
Algorithm 1: random_silhouette_graph(n)
do
    $s_2 = \mathsf{shuffle}((1\ 2)\,(3\ 4)\ldots(n-1\ n))$
    $s_3 = \mathsf{shuffle}((1\ 2\ 3)\,(4\ 5\ 6)\ldots(n-2\ n-1\ n))$
while $\Gamma(s_2, s_3)$ is not connected
return $\Gamma(s_2, s_3)$
```

Note that the random permutations s_2 and s_3 may well determine a disconnected graph, but the proof of [3, Proposition 8.18] shows that this happens with vanishing probability (precisely: with probability $\frac{5}{36}n^{-1} + o(n^{-1})$). Therefore this algorithm (a rejection algorithm) produces a silhouette graph after k iterations, with $\mathbb{E}(k) \sim 1$.

5.2 Random $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graphs

We exploit, again, the bijections established in Section 3.1, which we already used to derive the recurrence relations in Section 4.2. This yields Algorithm 2, to randomly generate a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $(\boldsymbol{\tau})$.

In the description of this algorithm, we use the following notation: if v is an integer, shift_v is the map defined on integers by $\mathsf{shift}_v(x) = x$ if $x < v$ and $\mathsf{shift}_v(x) = x+1$ if $x \geqslant v$; if v, w are distinct integers, $\mathsf{shift}_{v,w}$ is the map defined on integers by $\mathsf{shift}_{v,w}(x) = x$ if $x < \min(v, w)$, $\mathsf{shift}_{v,w}(x) = x+1$ if $\min(v, w) \leqslant x < \max(v, w)-1$, and $\mathsf{shift}_{v,w}(x) = x+2$ if $x \geqslant \max(v, w) - 1$. Note that shift_v "pushes" all integers greater than or equal to v by one unit, so that the range of shift_v misses v; similarly, the range of $\mathsf{shift}_{v,w}$ misses v and w.

We extend this notation to any graph Δ labeled by integers: if v is an integer, the graph $\mathsf{shift}_v(\Delta)$ is a relabeling of the vertices of Δ using shift_v on each vertex label; if v and w are two distinct integers, the graph $\mathsf{shift}_{v,w}(\Delta)$ is a relabeling of the vertices of Δ using $\mathsf{shift}_{v,w}$ on each vertex label.

Theorem 21. *Algorithm 2,* `random_cyclically_reduced_graph`*, produces, on input a combinatorial type $\boldsymbol{\tau}$, a random $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of type $\boldsymbol{\tau}$.*

Proof. Let $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$. We proceed by induction on n. Note that Algorithm `random_cyclically_reduced_graph` is recursive, and that, for any $\boldsymbol{\tau}$, only one of the outer `if` statements (Lines 1, 3, 5, 7, 9, 14 and 26) holds. Moreover, the algorithm stops immediately after the `if` statements of Lines 1, 3, 5, 7, and strictly decreases the value of n for the other ones. As a result, Algorithm `random_cyclically_reduced_graph` stops on any input $\boldsymbol{\tau}$ (which is a proper combinatorial type).

The statement of the theorem holds trivially if $n \leqslant 2$. Let us now assume that $n > 2$.

Algorithm 2: `random_cyclically_reduced_graph`$(\boldsymbol{\tau})$

```
if $\boldsymbol{\tau} = (1,0,0,1,1)$ then
    return the unique labeled $\Delta_1$
if $\boldsymbol{\tau} = (2,1,1,0,0)$ then
    return any one of the two labeled $\Delta_2$
if $\boldsymbol{\tau} = (2,0,1,2,0)$ then
    return any one of the two labeled $\Delta_3$
if $\boldsymbol{\tau} = (1,0,0,1,1)$ then
    return the unique labeled $\Delta_4$
// At this stage $n$ is necessarily greater than 2
if $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ and $\ell_3 > 0$ then
    $\Delta$ = random_cyclically_reduced_graph($\boldsymbol{\tau} + \boldsymbol{\lambda}_3$)
    $w$ = uniform random vertex with an $a$-loop $\ell'$ in $\Delta$
    $v$ = uniform random integer in $\{1, \dots, n\}$
    return $\Gamma$ constructed from $\Delta$ by relabeling its vertices using $\mathsf{shift}_v$, removing the $a$-loop $\ell'$
      at $\mathsf{shift}_v(w)$, adding a new vertex labeled $v$ and a $b$-loop at $v$, and adding an $a$-edge
      between $v$ and $\mathsf{shift}_v(w)$
if $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, 0)$ and $\ell_2 > 0$ then
    $x$ = uniform integer in $[s(n, k_2, k_3, \ell_2, 0)]$
    if $x \leqslant n \cdot (k_3 + 1) \cdot s(n-1, k_2, k_3+1, \ell_2 - 1, 0)$ then
        $\Delta$ = random_cyclically_reduced_graph($\boldsymbol{\tau} + \boldsymbol{\lambda}_{2,1}$)
        $(w \xrightarrow{b} w')$ = uniform random isolated $b$-edge in $\Delta$
        $v$ = uniform random integer in $\{1, \dots, n\}$
        return $\Gamma$ constructed from $\Delta$ by relabeling its vertices using $\mathsf{shift}_v$, adding a new
          vertex labeled $v$ and an $a$-loop at $v$, and adding $b$-edges from $\mathsf{shift}_v(w')$ to $v$ and from
          $v$ to $\mathsf{shift}_v(w)$
    else
        $\Delta$ = random_cyclically_reduced_graph($\boldsymbol{\tau} + \boldsymbol{\lambda}_{2,2}$)
        $w'$ = uniform random vertex with a $a$-loop $\ell'$ in $\Delta$
        $(v, w)$ = uniform random pair of distinct integers in $\{1, \dots, n\}$
        return $\Gamma$ constructed from $\Delta$ by removing the $a$-loop $\ell'$, relabeling the vertices of $\Delta$
          using $\mathsf{shift}_{v,w}$, adding new vertices labeled $v$ and $w$, an $a$-edge between $w$ and
          $\mathsf{shift}_{v,w}(w')$, a $b$-edge between $v$ and $w$ (choosing orientation uniformly at random)
          and an $a$-loop at $v$
if $\boldsymbol{\tau} = (n, k_2, k_3, 0, 0)$ and $k_3 > 0$ then
    $\Delta$ = random_cyclically_reduced_graph($\boldsymbol{\tau} + \boldsymbol{\kappa}_3$)
    $(v' \overset{a}{-} w')$ = uniform random isolated $a$-edge in $\Delta$
    $(v, w)$ = uniform random pair of distinct integers in $\{1, \dots, n\}$
    return $\Gamma$ constructed from $\Delta$ by removing the the $a$-edge $e'$, relabeling the vertices using
      $\mathsf{shift}_{v,w}$, adding a $b$-edge between $v$ and $w$ (choosing its orientation uniformly at random)
      and $a$-edges between $v$ and $\mathsf{shift}_{v,w}(v')$, and between $w$ and $\mathsf{shift}_{v,w}(w')$, respectively
// At this stage $\ell_2 = \ell_3 = k_3 = 0$
return random_silhouette_graph($n$)
```

If $\ell_3 > 0$ (that is, if the condition of Line 9 holds), Lemma 8 describes a bijection between the set of pairs (Γ, ℓ), where Γ is a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and ℓ is a b-loop in Γ, and the set of triples (Δ, ℓ', v) where Δ is a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of type $\boldsymbol{\tau} + \boldsymbol{\lambda}_3$, ℓ' is an a-loop in Δ and $v \in [n]$. This bijection, between two finite sets, preserves uniformity. Thus the first steps in this case (selecting uniformly at random Δ, ℓ' and v) translate into the selection, uniformly at random, of a pair (Γ, ℓ) where Γ has combinatorial type $\boldsymbol{\tau}$ and ℓ is one of the ℓ_3 b-loops in Γ (a number that depends on $\boldsymbol{\tau}$ but not on Γ). Forgetting the ℓ-component of this pair yields a randomly chosen $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of type $\boldsymbol{\tau}$.

The reasoning is exactly similar if the condition of Line 26 holds, relying on Lemma 10.

For the condition of Line 14, we need to handle the two options, corresponding to $\lambda_{2,1}$- and $\lambda_{2,2}$-moves. The set of pairs (Γ, ℓ), where Γ is a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph of combinatorial type $\boldsymbol{\tau}$ and ℓ is an a-loop in Γ, is partitioned in two subsets S_1 and S_2: $(\Gamma, \ell) \in S_1$ if ℓ is adjacent to a b-triangle, and $(\Gamma, \ell) \in S_2$ if ℓ is adjacent to an isolated b-edge. Lemma 9 describes the sets S_1 and S_2 are in bijection with. This determines the cardinalities of S_1 and S_2, which correspond precisely to the probability tested at Line 16. The reasoning is then identical to Lines 9 and 26.

Finally, if none of these conditions holds (so that $\boldsymbol{\tau}$ is the combinatorial type of a silhouette graph), the algorithm uses the `return` command on Line 31 to produce a random silhouette graph, see Section 5.1. □

5.3 Random subgroups of $\mathsf{PSL}_2(\mathbb{Z})$

We show how to randomly generate subgroups of a given combinatorial type, and then of a given size and isomorphism type.

Random generation for a given combinatorial type

Let $\boldsymbol{\tau} = (n, k_2, k_3, \ell_2, \ell_3)$ be a combinatorial type. The formula for the number $L(\boldsymbol{\tau})$ of labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graphs of type $\boldsymbol{\tau}$, in Proposition 15 above, suggests the following algorithm to draw uniformly at random a labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph of combinatorial type $\boldsymbol{\tau}$.

(1) Draw an integer $0 \leqslant p < L(\boldsymbol{\tau})$ uniformly at random.

(2) If $p < n \cdot s(\boldsymbol{\tau})$ and q is the quotient of p by $s(\boldsymbol{\tau})$ (so that $0 \leqslant q < n$), draw uniformly at random a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $\boldsymbol{\tau}$ and root it at vertex $q + 1$.

(3) If $n \cdot s(\boldsymbol{\tau}) \leqslant p < n \cdot s(\boldsymbol{\tau}) + (\ell_2 + 1) \cdot s(n, k_2, k_3, \ell_2 + 1, \ell_3)$ and q is the quotient of $p - n \cdot s(\boldsymbol{\tau})$ by $s(n, k_2, k_3, \ell_2 + 1, \ell_3)$ (so that $0 \leqslant q \leqslant \ell_2$), draw uniformly at random a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $(n, k_2, k_3, \ell_2 + 1, \ell_3)$ (as in Section 5.2), delete the $(q + 1)$st a-loop (following the order of vertex labels) and root the graph at the vertex where that loop used to be.

(4) If $n \cdot s(\boldsymbol{\tau}) + (\ell_2+1) \cdot s(n, k_2, k_3, \ell_2+1, \ell_3) \leqslant p$ and q is the quotient of $p - n \cdot s(\boldsymbol{\tau}) - (\ell_2+1) \cdot s(n, k_2, k_3, \ell_2+1, \ell_3)$ by $s(n, k_2, k_3, \ell_2, \ell_3+1)$ (so that $0 \leqslant q \leqslant \ell_3$), draw uniformly at random a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $(n, k_2, k_3, \ell_2, \ell_3+1)$ (as in Section 5.2), delete the $(q+1)$st b-loop (following the order of vertex labels) and root the graph at the vertex where that loop used to be.

To draw uniformly at random a subgroup of combinatorial type $\boldsymbol{\tau}$, we first draw a labeled $\mathsf{PSL}_2(\mathbb{Z})$-reduced graph of type $\boldsymbol{\tau}$, and then forget the labeling.

Remark 22. To draw uniformly at random a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced subgroup of combinatorial type $\boldsymbol{\tau}$, the algorithm is modified as follows: in step (1), one draws an integer p between 0 and $n \cdot s(\boldsymbol{\tau}) - 1$; one then applies only step (2).

Random generation for a given size and isomorphism type

Now let n be a positive integer and let $\boldsymbol{\sigma} = (\ell_2, \ell_3, r)$ be an isomorphism type. Let $k_2 = \frac{1}{2}(n - \ell_2)$, $k_3 = \frac{1}{2}(n - 3\ell_2 - 4\ell_3 - 6r + 6)$, $k_3' = \frac{1}{2}(n - 3\ell_2 - 4\ell_3 - 6r + 2)$ and $k_3'' = \frac{1}{2}(n - 3\ell_2 - 4\ell_3 - 6r + 4)$.

Proposition 16 suggests the following algorithm to draw uniformly at random a subgroup of size n and isomorphism type $\boldsymbol{\sigma}$.

(1) Draw uniformly at random an integer p between 0 and

$$n \cdot s(n, k_2, k_3, \ell_2, \ell_3) + (\ell_3+1) \cdot s(n, k_2, k_3', \ell_2, \ell_3+1) + (\ell_2+1) \cdot s(n, k_2, k_3'', \ell_2+1, \ell_3) - 1.$$

(2) If $p < n \cdot s(n, k_2, k_3, \ell_2, \ell_3)$, draw uniformly at random a labeled rooted $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $(n, k_2, k_3, \ell_2, \ell_3)$.

(3) If $n \cdot s(\boldsymbol{\tau}) \leqslant p < n \cdot s(\boldsymbol{\tau}) + (\ell_3+1) \cdot s(n, k_2, k_3', \ell_2, \ell_3+1)$ and q is the quotient of $p - n \cdot s(\boldsymbol{\tau})$ by $s(n, k_2, k_3', \ell_2, \ell_3+1)$ (so that $0 \leqslant q \leqslant \ell_3$), draw uniformly at random a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $(n, k_2, k_3', \ell_2, \ell_3+1)$, delete the $(q+1)$st b-loop (following the order of vertex labels) and root the graph at the vertex where that loop used to be.

(4) If $n \cdot s(\boldsymbol{\tau}) + (\ell_3+1) \cdot s(n, k_2, k_3', \ell_2, \ell_3+1) \leqslant p$ and q is the quotient of $p - n \cdot s(\boldsymbol{\tau}) - (\ell_3+1) \cdot s(n, k_2, k_3', \ell_2, \ell_3+1)$ by $s(n, k_2, k_3'', \ell_2+1, \ell_3)$ (so that $0 \leqslant q \leqslant \ell_2$), draw uniformly at random a labeled $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced graph with combinatorial type $(n, k_2, k_3, \ell_2+1, \ell_3)$, delete the $(q+1)$st a-loop (following the order of vertex labels) and root the graph at the vertex where that loop used to be.

This algorithm can be modified as in Remark 22 to draw uniformly at random a $\mathsf{PSL}_2(\mathbb{Z})$-cyclically reduced subgroup of a given isomorphism type.

5.4 Implementation and complexity remarks

Two models of computation to measure complexity

For the complexity analysis, we consider two classical models: the *unit-cost model* (also known as *RAM model*) where each elementary operation, including operations on integers, takes $O(1)$ time; and the *bit-cost* model where an integer N is encoded using $O(\log N)$ space, the number of bits of its representation, and where arithmetic operations are not performed in constant time anymore. This is more realistic in our settings because we are led to handling large integers. For instance, $O(n \log n)$ bits are required to represent the number of size n silhouette graphs (see [3, Proposition 8.18]). To simplify the discussion below, we use the following classical notation: for any $\alpha \geqslant 0$, a non-negative sequence u_n is in $\widetilde{O}(n^\alpha)$ if there exist constants $C, \beta > 0$ such that $u_n \leqslant C\, n^\alpha \log^\beta n$ for all n sufficiently large. Informally, it means that u_n is in $O(n^\alpha)$ "up to a poly-logarithmic factor". Note that, in the bit-cost model, adding or multiplying two numbers encoded with at most N bits costs $\widetilde{O}(N)$ time. It is elementary for addition and a consequence of, for instance, Harvey and van der Hoeven's result [10] for multiplication. Comparing two numbers encoded with at most N bits costs $O(N)$ time.

In the unit-cost model, we consider that, for any positive integer n, we can generate uniformly at random an integer in $[n]$ in $O(1)$ time. In the bit-cost model, we consider that we can generate uniformly at random a bit value of $\{0, 1\}$ in $O(1)$ time. If n is a positive integer encoded with N bits, we can therefore produce an element of $[n]$ uniformly at random using a rejection algorithm consisting in repeatedly generating a number made of N independent random bits until the result is in $[n]$. The expected running time of this algorithm is $O(N)$ as the expected number of attempts is at most 2.

Precomputing

Since the parameters n, ℓ_2, ℓ_3, k_2 and k_3 are non-negative and satisfy $n = 2k_2 + \ell_2$ and $n \geqslant 2k_3 + \ell_3$, there are at most n^4 non-zero values for $s(n, k_2, k_3, \ell_2, \ell_3)$ for a given positive integer n. They can be computed recursively using Equations (1), (2) and (3), the base cases being either trivial (for $n \geqslant 2$) or given by Proposition 20. This yields an $O(n^4)$ time and space algorithm in the unit-cost model and $\widetilde{O}(n^5)$ time and space algorithm in the bit-cost model.

Random generation

We assume in this section that all the required values of $s(n, k_2, k_3, \ell_2, \ell_3)$ have been precomputed and are accessible in time $O(1)$ in the unit-cost model, $\widetilde{O}(1)$ in the bit-cost model.

Algorithm `random_cyclically_reduced_graph` was written with the proof of Theorem 21 in mind. This is the reason why, in particular, it calls for randomly choosing an integer v, or integers v and w. One can also choose $v = n$, or $v = n$ and $w = n - 1$, that is, disregard the randomness of the labeling of the graph we constuct, and add a very last step to the algorithm, which relabels Γ by a random permutation. This is a classic trick in the literature on the random generation of labeled combinatorial objects (see for instance [7, footnote on p. 12]).

In the unit-cost model, `random_silhouette_graph` runs in $O(n)$ average time, using the Fisher-Yates shuffling algorithm [13, p.145] and the fact that the number of iterations in `random_silhouette_graph` is bounded in expectation, see Section 5.1. If we use the trick mentioned above (relabeling the graph at the end), every call of the function `random_cyclically_reduced_graph` is performed in $O(1)$. As each call decreases the value of n by at least 1, Algorithm `random_cyclically_reduced_graph` runs in $O(n)$ expected time.

In the bit-cost model, observe that n is encoded using $O(\log n)$ bits, so that all arithmetic operations on n, ℓ_2, ℓ_3, k_2, k_3 are performed in $\widetilde{O}(1)$ time. The bottleneck for the running time of the algorithm lies therefore in Lines 15-16, as generating x and comparing x with the threshold in Line 16 both cost $\widetilde{O}(n)$ time. The overall expected running time of the algorithm in the bit-cost model is therefore $\widetilde{O}(n^2)$.

This process (generating x and comparing it with a theshold) can be improved using the following idea. Assume that we have two large integers s and t, and the sum $s+t$ has already been computed. Let $z_0 \cdots z_{N-1}$ be the binary encoding of $s+t$ (with $z_0 = 1$). Let also $s_0 \cdots s_{N-1}$ be the binary encoding of s (here s_0 may be 0). Generating x in $[s+t]$ and comparing it to s, amounts to simulating a Bernoulli law of parameter $\frac{s}{s+t}$ (in the bit-cost model). This is performed using Algorithm `bernoulli_attempt`, which generates a uniform random integer, say x, between 0 and $2^N - 1$ bit by bit, halting as soon as we are guaranteed that one of the three possible situations holds : $x \geqslant s+t$ (Failure), $x < s$ (`True`) and $s \leqslant x < s+t$ (`False`).

Algorithm 3: `bernoulli_attempt`(x, y, N)

```
$smaller_{s+t} = \emptyset$
$smaller_s = \emptyset$
for $i \in \{0, \ldots, N-1\}$ do
    $bit$ = Uniform($\{0,1\}$)
    if $smaller_{s+t} = \emptyset$ then
        if $bit > z_i$ then return Failure
        if $bit < z_i$ then $smaller_{s+t}$ = True
    if $smaller_s = \emptyset$ then
        if $bit > s_i$ then $smaller_s$ = False
        if $bit < s_i$ then $smaller_s$ = True
    if $smaller_{s+t} \neq \emptyset$ and $smaller_s \neq \emptyset$ then
        return $smaller_s$
if $smaller_{s+t} = \emptyset$ then
    return Failure // the generated number is $\geqslant s+t$
return False // the generated number is equal to $s$
```

The main algorithm to simulate the Bernoulli law of parameter $\frac{s}{s+t}$ consists in repeatedly calling `bernoulli_attempt` until the result is not Failure. The analysis of the expected number of bits generated in the process is straightforward, as $smaller_{s+t}$ and $smaller_s$ are determined with probability $\frac{1}{2}$ at each iteration of the first loop: the number of iterations required to determine each one of them is bounded above by a geometric law of parameter $\frac{1}{2}$. Since $s+t > 2^{N-1}$, the expected number of calls to `bernoulli_attempt` is bounded above by a constant. Hence the expected bit-cost complexity of our procedure to simulate the Bernoulli law is $\widetilde{O}(1)$.

To implement the announced improvement, we modify the precomputation step by storing not only the values of $s(n, k_2, k_3, \ell_2, \ell_3)$, but also their bit-lengths and the values $n \cdot (k_3+1) \cdot s(n-1, k_2, k_3+1, \ell_2-1, 0)$ (used Line 16). Simulating a Bernoulli law of parameter $n \cdot (k_3+1) \cdot s(n-1, k_2, k_3+1, \ell_2-1, 0)/s(n, k_2, k_3, \ell_2, \ell_3)$ instead of performing Lines 15 and 16, lowers the expected time complexity of `random_cyclically_reduced_graph` to $\widetilde{O}(n)$.

Remark 23. When $\ell_3 > 0$, one can directly choose ℓ_3 a-loops of the graph Δ built at Line 10 to apply the inverse of a λ_3-move ℓ_3 times directly. This does not change the overall complexity of `random_cyclically_reduced_graph` in both models of computation. Similarly, if $\ell_2 = \ell_3 = 0$ and $k_3 = n/2$, the generated graph is a cycle made of an alternation of a-transitions and b-transitions, which could be generated directly without making several recursive call. Again, this does not change the complexity.

References

[1] G. N. Arzhantseva. Generic properties of finitely presented groups and Howson's theorem. *Comm. Algebra*, 26(11):3783–3792, 1998.

[2] G. N. Arzhantseva and A. Y. Ol'shanskiĭ. Generality of the class of groups in which subgroups with a lesser number of generators are free. *Mat. Zametki*, 59(4):489–496, 1996.

[3] F. Bassino, C. Nicaud, and P. Weil. Statistics of subgroups of the modular group. *Int. J. Algebra Comput.*, 31(8):1691–1751, 2021.

[4] F. Bassino, C. Nicaud, and P. Weil. Silhouettes and generic properties of subgroups of the modular group. `arXiv:2311.08021`, 2023.

[5] I. M. S. Dey. Schreier systems in free products. *Proc. Glasgow Math. Assoc.*, 7(2):61–79, 1965.

[6] P. Flajolet and R. Sedgewick. *Analytic combinatorics*. Cambridge University Press, 2009.

[7] P. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theor. Comput. Sci.*, 132(2):1–35, 1994.

[8] S. M. Gersten and H. B. Short. Rational subgroups of biautomatic groups. *Ann. of Math.*, 134(1):125–158, 1991.

[9] R. Gitik. Nielsen generating sets and quasiconvexity of subgroups. *J. Pure Appl. Algebra*, 112(3):287–292, 1996.

[10] D. Harvey and J. van der Hoeven. Integer multiplication in time $O(n \log n)$. *Annals of Mathematics*, 193(2):563–617, 2021.

[11] I. Kapovich. Detecting quasiconvexity: algorithmic aspects. In *Geometric and computational perspectives on infinite groups (Minneapolis, MN and New Brunswick, NJ, 1994)*, volume 25 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 91–99. Amer. Math. Soc., Providence, RI, 1996.

[12] O. Kharlampovich, A. Miasnikov, and P. Weil. Stallings graphs for quasi-convex subgroups. *J. Algebra*, 488:442–483, 2017.

[13] D. E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms (third edition).* Addison-Wesley, 1989.

[14] A. G. Kurosh. *The theory of groups. Vol. II.* Chelsea Publishing Company, New York, N.Y., 1956. Translated from the Russian and edited by K. A. Hirsch.

[15] R. C. Lyndon and P. E. Schupp. *Combinatorial group theory.* Springer-Verlag, 1977.

[16] L. Markus-Epstein. Stallings foldings and subgroups of amalgams of finite groups. *Internat. J. Algebra Comput.*, 17(8):1493–1535, 2007.

[17] T. W. Müller and J.-C. Schlage-Puchta. Classification and statistics of finite index subgroups in free products. *Adv. Math.*, 188(1):1–50, 2004.

[18] M. Newman. Asymptotic formulas related to free products of cyclic groups. *Math. Comp.*, 30(136):838–846, 1976.

[19] Y. Ollivier. *A January 2005 invitation to random groups*, volume 10 of *Ensaios Matemáticos [Mathematical Surveys].* Sociedade Brasileira de Matemática, 2005.

[20] J. J. Rotman. *An introduction to the theory of groups*, volume 148 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, fourth edition, 1995.

[21] H. Short. Quasiconvexity and a theorem of Howson's. In *Group theory from a geometrical viewpoint (Trieste, 1990)*, pages 168–176. World Sci. Publ., River Edge, NJ, 1991.

[22] J. R. Stallings. Topology of finite graphs. *Invent. Math.*, 71(3):551–565, 1983.

[23] W. W. Stothers. The number of subgroups of given index in the modular group. *Proc. Roy. Soc. Edinburgh Sect. A*, 78(1-2):105–112, 1977/78.

[24] W. W. Stothers. Free subgroups of the free product of cyclic groups. *Math. Comp.*, 32(144):1274–1280, 1978.