Algorithms related to subgroups of the modular group

Constantin Cristian Caranica

Louisiana State University and Agricultural and Mechanical College

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_dissertations

Part of the Applied Mathematics Commons

Recommended Citation
https://digitalcommons.lsu.edu/gradschool_dissertations/854

This Dissertation is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Doctoral Dissertations by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
ALGORITHMS RELATED TO SUBGROUPS
OF THE MODULAR GROUP

A Dissertation
Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

in
The Department of Mathematics

by
Constantin C. Caranica
B.S. in Mathematics, Bucharest University, 1997
M.S. in Mathematics, Bucharest University, 2000
M.S. in Mathematics, Louisiana State University, 2005
August 2009
Acknowledgments

This dissertation would not have been possible without several contributions. First, I would like to thank my advisor Helena Verrill for her support and advice. Her trust in my work has given me a lot of autonomy. Her knowledge and patience were crucial to overcome difficulties in the course of my studies. I would also like to thank all the committee members Dr. Robert Perlis, Dr. William Adkins, Dr. Mark Davidson, Dr. Patrick Gilmer, and Dr. Warren Waggenspack.

I am also pleased to thank the Giles family for their continuous support and encouragement. Terry, Rose, Jon and Jordan Giles had faith in me and it is a pleasure to express them my deepest gratitude.

Finally, I would like to give a special thanks to my family and to my dear friend Liviu Mircea. Their help and sacrifice made this dissertation possible.
# Table of Contents

Acknowledgments ................................................. ii  
List of Tables ..................................................... iv  
List of Figures ...................................................... vi  
Abstract ............................................................ vii  

Chapter 1: Introduction ........................................... 1  
1.1 Basic Definitions and Terminology .......................... 3  
1.2 Linear Fractional Transformations ........................... 5  
1.3 Fundamental Domains and Special Polygons ............... 9  
1.4 Farey Symbols ................................................... 15  
1.5 Cusp Widths and Congruence Subgroups ..................... 18  
1.6 The Side Pairing Matrices ..................................... 21  
1.7 Geometric Invariants of a Subgroup and Their Relations .. 23  

Chapter 2: Generating Subgroups of a Certain Index .......... 26  
2.1 Trivalent Diagrams and Tree Diagrams ...................... 26  
2.2 Vidal’s Classification Principle ............................. 30  

Chapter 3: Membership Tests for Matrices in $PSL_2(\mathbb{Z})$ .... 40  
3.1 Lang, Lim and Tan’s Algorithm ............................. 40  
3.2 A New Algorithm ............................................... 47  
3.3 Small-Index Noncongruence Subgroups ..................... 55  
3.4 Conclusions ...................................................... 57  

References ......................................................... 58  

Appendix: Magma Programs ......................................... 60  

Vita ................................................................. 79
List of Tables

3.1 Noncongruence Subgroups of Index Less Than 10 . . . . . . . . . . 56
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A fundamental domain of $PSL_2(\mathbb{Z})$</td>
<td>9</td>
</tr>
<tr>
<td>1.2</td>
<td>A special polygon</td>
<td>15</td>
</tr>
<tr>
<td>1.3</td>
<td>Calculating the cusp widths</td>
<td>20</td>
</tr>
<tr>
<td>2.1</td>
<td>The tessellation inside the polygon</td>
<td>31</td>
</tr>
<tr>
<td>2.2</td>
<td>The tree diagram</td>
<td>32</td>
</tr>
<tr>
<td>2.3</td>
<td>The rotated tree diagram</td>
<td>32</td>
</tr>
<tr>
<td>2.4</td>
<td>The bipartite cuboid graph</td>
<td>32</td>
</tr>
<tr>
<td>2.5</td>
<td>Changing the marked edge and the cycle cuts</td>
<td>33</td>
</tr>
<tr>
<td>2.6</td>
<td>Reading the Farey labels</td>
<td>34</td>
</tr>
<tr>
<td>2.7</td>
<td>The $D$ triangle</td>
<td>34</td>
</tr>
<tr>
<td>2.8</td>
<td>A generic $f$-edge and the associated cusps</td>
<td>34</td>
</tr>
<tr>
<td>2.9</td>
<td>The rotated $f$-edge</td>
<td>35</td>
</tr>
<tr>
<td>2.10</td>
<td>The edge $e_1$</td>
<td>35</td>
</tr>
<tr>
<td>2.11</td>
<td>Adding a cusp between two given cusps</td>
<td>35</td>
</tr>
<tr>
<td>2.12</td>
<td>Adding a cusp to the right</td>
<td>36</td>
</tr>
<tr>
<td>2.13</td>
<td>Adding a cusp to the left</td>
<td>36</td>
</tr>
<tr>
<td>2.14</td>
<td>Rule 1</td>
<td>36</td>
</tr>
<tr>
<td>2.15</td>
<td>Rule 2</td>
<td>37</td>
</tr>
<tr>
<td>2.16</td>
<td>Finding the cusps</td>
<td>37</td>
</tr>
<tr>
<td>2.17</td>
<td>The cusps of $e_1$</td>
<td>37</td>
</tr>
<tr>
<td>2.18</td>
<td>The cusps around B</td>
<td>38</td>
</tr>
<tr>
<td>2.19</td>
<td>All the cusps</td>
<td>38</td>
</tr>
<tr>
<td>2.20</td>
<td>Reading the cusps</td>
<td>39</td>
</tr>
<tr>
<td>3.1</td>
<td>The tree of $f$-edges</td>
<td>41</td>
</tr>
<tr>
<td>3.2</td>
<td>The tree and the new edge</td>
<td>41</td>
</tr>
</tbody>
</table>
Abstract

Classifying subgroups of the modular group $PSL_2(\mathbb{Z})$ is a fundamental problem with applications to modular forms, in addition to its group-theoretic interest. While a lot of research has been done on the congruence subgroups of $PSL_2(\mathbb{Z})$, very little is known about noncongruence subgroups. The purpose of this thesis is to find and characterize small-index noncongruence subgroups of the modular group $PSL_2(\mathbb{Z})$.

We use the concept of Farey symbol to describe the subgroups of $PSL_2(\mathbb{Z})$. The first part contains results concerning the geometry of subgroups of $PSL_2(\mathbb{Z})$. The second part describes a graph-theoretical approach to finding all subgroups of a given index. In the third part we describe two algorithms for testing the membership of a matrix to a subgroup given by a Farey symbol. As an application we find the noncongruence subgroups of $PSL_2(\mathbb{Z})$ of index less than 10.
Chapter 1

Introduction

The special linear group $SL_2(\mathbb{Z})$ consists of the 2-by-2 integral matrices of determinant 1. Known also as the homogeneous modular group, $SL_2(\mathbb{Z})$ is one of the most important and most studied discrete groups. A finite index subgroup of $SL_2(\mathbb{Z})$ is said to be a congruence subgroup, if it contains the kernel of the natural homomorphism from $SL_2(\mathbb{Z})$ to $SL_2(\mathbb{Z}/N\mathbb{Z})$ for some positive integer $N$; otherwise, a finite-index subgroup is called a noncongruence subgroup. The existence of noncongruence subgroups of $SL_2(\mathbb{Z})$ was determined for the first time in 1886 by Fricke [Fri86] and Pick [Pic86]. As a matter of fact, noncongruence subgroups of $SL_2(\mathbb{Z})$ predominate congruence subgroups. In contrast, any finite index subgroup of $SL_n(\mathbb{Z})$ with $n \geq 3$ is congruence [BL84]. A theorem of Belyi implies that any compact complex smooth irreducible curve defined over $\overline{\mathbb{Q}}$ can be realized as a modular curve for a finite index subgroup of $SL_2(\mathbb{Z})$. Finite index subgroups arise in many other fields such as triangular groups, theory of translation surfaces or Galois coverings of the projective line.

Identifying the elements of $SL_2(\mathbb{Z})$ which differ by a sign, one obtains the inhomogeneous modular group $PSL_2(\mathbb{Z})$. The congruence and noncongruence subgroups of $PSL_2(\mathbb{Z})$ are defined in a similar manner. They correspond to the congruence and noncongruence subgroups of $SL_2(\mathbb{Z})$ after identifying the matrices that differ by sign. The majority of finite index subgroups of $PSL_2(\mathbb{Z})$ are noncongruence.

While many tried to describe the noncongruence subgroups, nobody tried to classify them by index. This thesis approaches this classification problem in three steps.
Using the classification principle of Vidal [Vid06], we transfer the problem of finding noncongruence subgroups to finding marked trivalent diagrams corresponding to noncongruence subgroups. For this we need to generate all trivalent diagrams of a given size. Although Vidal [Vid07] proposed an algorithm to generate all trivalent diagrams of a given size using dynamic programming methods, I provide an algorithm to generate all trivalent diagrams using a graph-theoretical method, a method which inductively generates diagrams of size $d$ from diagrams of size $d - 1$ and diagrams of size $d - 3$. The first program in the appendix provides the Magma code for this. To generate all trivalent diagrams of size 1 to $m$ just use the function gentrivdiag($m$).

The second step goes from a marked trivalent diagram to a Farey symbol, which is a method of describing a subgroup of $PSL_2(\mathbb{Z})$. In chapter 2, I provide an algorithm for obtaining the Farey symbol corresponding to a marked trivalent diagram. To my knowledge this is the first algorithm that performs this task. The code for this algorithm is in the second program in the appendix. Just use travdiag($G$, cycles$G$, Vsat$G$, ein, d), where $G$ is the diagram, cycles$G$ is a list of its cycles and Vsat$G$ is a list of white vertices of degree 2. The function returns the cusp vertices without $\infty$ and the list of Farey labels.

The third step is to test whether a subgroup $\Phi \subseteq PSL_2(\mathbb{Z})$ is congruence or not. Lang, Lim and Tan [LLT95] provided an algorithm for testing the congruence of a subgroup. In chapter 3, I will provide another algorithm for testing the congruence of a subgroup of $PSL_2(\mathbb{Z})$. The third program in the appendix provides the code for Lang, Lim and Tan’s algorithm. Just use the function resneed($V$, FS, n), where $V$ is the set of $n + 1$ vertices without $-\infty$ and $\infty$ and FS is the sequence of Farey labels.
All these algorithms tied together provide the tools for finding the noncongruence subgroups of a given index. A table of noncongruence subgroups of index less than 10 is provided at the end of the thesis.

1.1 Basic Definitions and Terminology

Let $\Gamma$ denote the inhomogeneous modular group $PSL_2(\mathbb{Z}) = SL_2(\mathbb{Z})/\pm I$ acting on the upper half plane $H = \{z \in \mathbb{C} \mid Im(z) > 0\}$ via $g(z) = \frac{az+b}{cz+d}$ for $g = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \Gamma$.

For a positive integer $N$, the principal congruence subgroups are

$$\Gamma(N) = \{ (\begin{pmatrix} a & b \\ c & d \end{pmatrix}) \in \Gamma \mid (\begin{pmatrix} a & b \\ c & d \end{pmatrix}) \equiv (\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) \mod N \}$$

$$\Gamma_0(N) = \{ (\begin{pmatrix} a & b \\ c & d \end{pmatrix}) \in \Gamma \mid c \equiv 0 \mod N \}$$

$$\Gamma^0(N) = \{ (\begin{pmatrix} a & b \\ c & d \end{pmatrix}) \in \Gamma \mid b \equiv 0 \mod N \}$$

$$\Gamma_1(N) = \{ (\begin{pmatrix} a & b \\ c & d \end{pmatrix}) \in \Gamma \mid c \equiv 0 \mod N \text{ and } a \equiv 1 \mod N \}$$

A congruence subgroup is any subgroup $\Phi \subseteq PSL_2(\mathbb{Z})$ with $\Gamma(N) \subseteq \Phi$ for some positive integer $N$. The smallest $N$ such that $\Gamma(N) \subseteq \Phi$ is known as the (arithmetic) level of the subgroup. When we refer to “the principal congruence subgroup of level $N$”, we are referring to $\Gamma(N)$, even though the other subgroups listed above are known as principal congruence subgroups.

Classifying subgroups of $PSL_2(\mathbb{Z})$ is a fundamental problem in group theory and number theory with numerous applications to the theory of modular forms. This problem can be split into classifying congruence and noncongruence subgroups, but first we need a method of determining when a subgroup is congruence. Such an algorithm was given in [LLT95] and I will provide a similar algorithm in part three of the thesis.

The motivation for splitting up the classification of subgroups into congruence and noncongruence subgroups is as follows: note that $\Gamma(N)$ is the kernel of the map

$$\varphi : PSL_2(\mathbb{Z}) \rightarrow PSL_2(\mathbb{Z}/\mathbb{N})$$
Then, classifying the congruence subgroups of $PSL_2(\mathbb{Z})$ can be reduced to classifying the subgroups of the finite groups $PSL_2(\mathbb{Z}/\mathbb{N}\mathbb{Z})$, which is a relatively easier task. As for noncongruence subgroups, little work has been done on classifying them.

Another fundamental problem is to find a set of independent generators for a given subgroup. In [Kul91], Kulkarni introduces the concept of Farey symbol, which gives us a convenient method of finding a set of independent generators. These Farey symbols turn out to be very useful in describing the fundamental properties of the finite-index subgroups of $PSL_2(\mathbb{Z})$.

One of the conveniences of Farey symbols is that the index of the subgroup it represents can be read from the symbol relatively easily. Then, we can generate all subgroups of a given index by combinatorically generating all Farey symbols corresponding to a given index. Unfortunately, Farey symbols are in finite-to-one correspondence with subgroups of finite index. In [CLLT93], Chan, Lang, Lim and Tan give an algorithm for determining when two Farey symbols represent the same subgroup. This gives us a method of determining all subgroups of a given index without duplication. Although this method is very appealing and can be easily implemented, I will use a graph-theoretic method for finding the subgroups of a certain index.

Using the fact that $PSL_2(\mathbb{Z}) \simeq \mathbb{Z}_2 \ast \mathbb{Z}_3$, Kulkarni [Kul91] showed that the conjugacy classes of subgroups of $PSL_2(\mathbb{Z})$ are in one-to-one correspondence with bipartite cuboid graphs, bipartite graphs with vertices of degree at most 3 and with a counter-clockwise orientation around the vertices of degree 3. These are also called trivalent diagrams. The number of edges of a trivalent diagram is called the size of the diagram. Vidal [Vid06] proved that by traveling a trivalent diagram of size $d$ in a certain way one obtains subgroups of $PSL_2(\mathbb{Z})$ of index $d$. The subgroup one gets depends entirely on the choice of the first edge to travel and the
Thus, by marking the first edge to travel, we get a one-to-one relationship between subgroups of index $d$ and marked trivalent diagrams. After traveling along all diagrams of size $d$ in all admissible ways, we get all subgroups of $PSL_2(\mathbb{Z})$ of index $d$.

The two equivalences

\[
\left\{ \begin{array}{l}
\text{Conjugacy classes of subgroups} \\
\text{of } PSL_2(\mathbb{Z}) \text{ of index } d
\end{array} \right\} \leftrightarrow \left\{ \begin{array}{l}
\text{Trivalent diagrams} \\
\text{of size } d
\end{array} \right\}
\]

and

\[
\left\{ \begin{array}{l}
\text{Subgroups of } PSL_2(\mathbb{Z}) \\
\text{of index } d
\end{array} \right\} \leftrightarrow \left\{ \begin{array}{l}
\text{Marked trivalent diagrams} \\
\text{of size } d
\end{array} \right\}
\]

help us turn the problem of finding the subgroups of $PSL_2(\mathbb{Z})$ of a given index into a graph-theoretic problem: find all trivalent diagrams of a certain size and traverse them in all admissible ways. Every time a trivalent diagram is completely traveled along the edges we produce the Farey symbol of the corresponding subgroup. From the Farey symbol we can infer all generators, cusps, cusp widths and the genus of the subgroup and subsequently we can decide whether the subgroup is congruence or not.

We now recall some basic geometric and arithmetic definitions and some of the most important theorems related to fundamental domains of subgroups of finite index in $PSL_2(\mathbb{Z})$. We will denote $PSL_2(\mathbb{Z})$ by $\Gamma$. Most of the results come from [Kul91], [CLLT93], [Kob93] and [LLT95].

### 1.2 Linear Fractional Transformations

We investigate the linear fractional transformations induced by elements of $PSL_2(\mathbb{Z})$ on the upper half plane $\mathbb{H}$.

Let $\hat{\mathbb{C}}$ denote the Riemann sphere, (i.e. the complex plane with a point at infinity). If $z \in \mathbb{C}$ is a complex number and $g = (\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}) \in PSL_2(\mathbb{Z})$ we define

\[
g(z) \triangleq \frac{az + b}{cz + d}; \quad g(\infty) \triangleq \lim_{z \to \infty} \frac{az + b}{cz + d} = \frac{a}{c}
\]
If \( c = 0 \), then we put \( g(\infty) \triangleq \infty \). We note that any \( g = (a \ b \\
 c \ d) \in \text{PSL}_2(\mathbb{Z}) \) preserves the upper half plane \( \mathbb{H} \), i.e. \( \text{Im} \, z > 0 \) implies \( \text{Im} \, g(z) > 0 \). We have

\[
\text{Im} \, g(z) = \text{Im} \, \frac{az + b}{cz + d} = \text{Im} \, \frac{(az + b)(cz + d)}{|cz + d|^2} = \frac{\text{Im}(adz + bcz)}{|cz + d|^2} = \frac{(ad - bc)\text{Im} \, z}{|cz + d|^2}
\]

and because \( ad - bc = \det g = 1 \) we get

\[
\text{Im} \, g(z) = \frac{\text{Im} \, z}{|cz + d|^2} \quad \text{for} \quad g = (a \ b \\
 c \ d) \in \text{PSL}_2(\mathbb{Z}) \quad (1.1)
\]

Thus, \( \text{Im} \, g(z) = \frac{\text{Im} \, z}{|cz + d|^2} > 0 \) and it is easy to see that these linear fractional transformations, \( z \to gz \) define a group action on \( \tilde{\mathbb{C}} \).

A non-identity element \( g \in \text{PSL}_2(\mathbb{Z}) \) is called elliptic (parabolic or hyperbolic), if

\[
|\text{tr}(g)| < 2 \quad (\text{tr}(g) = \pm 2 \text{ or } |\text{tr}(g)| > 2),
\]

respectively.

Let us study the fixed points of the elements of \( \text{PSL}_2(\mathbb{Z}) \). Let \( g = (a \ b \\
 c \ d) \neq I \) be an element in \( \text{PSL}_2(\mathbb{Z}) \). A complex number \( z \) is a fixed point of \( g \) if it satisfies the equation

\[
g(z) = z, \quad \frac{az + b}{cz + d} = z
\]

or \( cz^2 + (d - a)z - b = 0 \). Notice that the discriminant of this equation is equal to \( \Delta = (d - a)^2 + 4bc = (a + d)^2 - 4(ad - bc) = \text{tr}(g)^2 - 4\det(g) = \text{tr}(g)^2 - 4 \). Let us find the fixed points of \( g \). First assume \( c = 0 \). Then \( \text{tr}(g)^2 - 4 = (a + d)^2 - 4(ad - bc) = (a + d)^2 - 4ad = (a - d)^2 \). Thus, \( g \) is parabolic or hyperbolic and \( \infty \) is a fixed point of \( g \). Remark that \( g \) is parabolic if and only if \( a = d \), and since \( g \) is non-identity we must have \( b \neq 0 \). Then, since \( z = \frac{b}{d-a} = \frac{b}{g} \), the unique fixed point of \( g \) is \( \infty \). We also see that if \( a \neq d \), then \( g \) is hyperbolic and it has two fixed points \( b/(d-a) \) and \( \infty \). Now we assume \( c \neq 0 \). Then the fixed points satisfy a quadratic equation, so they are conjugate complex numbers, a real number or two distinct real numbers if \( g \) is elliptic, parabolic or hyperbolic, respectively. The point \( \infty \) can not be a fixed point of \( g \) when \( c \neq 0 \). We also notice that if \( z \neq \infty \) is a parabolic element, then
is actually a rational number because $\Delta = 0$ implies $z = \frac{a-d}{2c}$ with $a$, $c$ and $d$
integers. We get the following theorem:

**Theorem 1.2.1.** An element $g \neq I$ of $\text{PSL}_2(\mathbb{Z})$ is characterized by its fixed points
as follows:

1) $g$ is elliptic if and only if $g$ has the fixed points $z_0$ and $z_0^\prime$, with $z_0 \in \mathbb{H}$
2) $g$ is parabolic if and only if $g$ has a unique fixed point on $\mathbb{Q} \cup \{\infty\}$
3) $g$ is hyperbolic if and only if $g$ has two distinct fixed points on $\mathbb{R} \cup \{\infty\}$.

Let us now fix a subgroup $\Gamma' \subseteq \text{PSL}_2(\mathbb{Z})$. When $z \in \mathbb{H} \cup \mathbb{R} \cup \{\infty\}$ is a fixed
point of an elliptic, parabolic or hyperbolic element of $\Gamma'$, we say that $z$ is an
elliptic point, a parabolic point or a hyperbolic point of $\Gamma'$, respectively. We also
call a parabolic point of $\Gamma'$ a cusp of $\Gamma'$. Let $g \in \Gamma'$ with $g(z) = z$ and $\gamma(z) = z'$,
where $\gamma \in \text{PSL}_2(\mathbb{Z})$. Then $z = \gamma^{-1}(z')$ and $g(z) = z$ implies $g\gamma^{-1}(z') = \gamma^{-1}(z')$ or
$\gamma g\gamma^{-1}(z') = (z')$. Since $tr(\gamma g\gamma^{-1}) = tr(g)$, $z$ elliptic (respectiv cusp or hyperbolic)
implies $\gamma(z)$ elliptic (respectiv cusp or hyperbolic). Thus, the property of a point
to be elliptic, parabolic or hyperbolic is invariant to the action of $\text{PSL}_2(\mathbb{Z})$.

Now we prove that the cusps of $\Gamma = \text{PSL}_2(\mathbb{Z})$ are precisely the points $\{\infty\} \cup \mathbb{Q}$.
As we saw above a cusp is either a rational number or it is infinity. Since $\infty$ is
fixed by $\left(\begin{smallmatrix}1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, it is a cusp. It remains to show that any rational number is a cusp
of $\text{PSL}_2(\mathbb{Z})$. But any fraction $\frac{a}{c}$ with $gcd(a,c) = 1$ can be completed to a matrix
$\left(\begin{smallmatrix}a & b \\ c & d \end{smallmatrix}\right) \in \Gamma$ by solving $ad - bc = 1$ for $b$ and $d$. This matrix takes $\infty$ to $\frac{a}{c}$. Hence all
rational numbers are in the same $\Gamma$-equivalence class as $\infty$, so they are cusps. $\Gamma$
permutes the cusps transitively.

If $\Gamma'$ is a subgroup of $\text{PSL}_2(\mathbb{Z})$ then $\Gamma'$ permutes the cusps, but in general not
transitively. Usually, there is more than one $\Gamma'$-equivalence class among the cusps
$\{\infty\} \cup \mathbb{Q}$. By an inequivalent cusp of $\Gamma'$ we mean a $\Gamma'$-equivalence class of cusps.

For $z \in \mathbb{H} \cup \mathbb{R} \cup \{\infty\}$ we put $\Gamma_z = \{g \in \Gamma \mid gz = z\}$, the isotropy subgroup of $z$
or the stabilizer of $z$ under the above action. For $z \in \mathbb{H}$, we put

$$\text{ord}_z = |\Gamma_z|$$
and call it the order of $z$ with respect to $\Gamma$. It is easy to see that for $z = \infty$ we have

$$\Gamma_\infty = \{(\begin{smallmatrix} 1 & b \\ 0 & 1 \end{smallmatrix}) | b \in \mathbb{Z}\} \cong \mathbb{Z}$$

Hence $\text{ord}_\infty = \infty$.

We now prove a decomposition theorem for $PSL_2(\mathbb{Z})$ and the corresponding linear fractional transformations.

**Theorem 1.2.2.** $PSL_2(\mathbb{Z})$ is generated by the matrices $T = (\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix})$ and $S = (\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix})$.

**Proof.** Let $G$ be the subgroup in $PSL_2(\mathbb{Z})$ generated by $T$ and $S$. Suppose that $G \neq PSL_2(\mathbb{Z})$. We have

$$STS^{-1} = (\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix})(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix})(\begin{smallmatrix} 0 & -1 \\ -1 & 0 \end{smallmatrix}) = (\begin{smallmatrix} 0 & -1 \\ 1 & -1 \end{smallmatrix})(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}) = (\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix})$$

and

$$ST^{-1}S^{-1} = (STS^{-1})^{-1} = (\begin{smallmatrix} 1 & 0 \\ -1 & 1 \end{smallmatrix}) .$$

Also

$$S^2 = (\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix})(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}) = (\begin{smallmatrix} -1 & 0 \\ 0 & -1 \end{smallmatrix}) = I.$$ 

Therefore, all elements of $PSL_2(\mathbb{Z})$ of the form $(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix})$ are contained in $G$. If we put $b_0 = \min\{|b| | (\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}) \in PSL_2(\mathbb{Z}) - G\}$, we have $b_0 \neq 0$.

Take an element $g_0 = (\begin{smallmatrix} a_0 & b_0 \\ c_0 & d_0 \end{smallmatrix}) \in PSL_2(\mathbb{Z}) - G$ and an integer $n$ such that $|a_0 - nb_0| < b_0$. If $b_0 = 1$ choose $n = a_0$, otherwise use coprimality of $a_0$ and $b_0$ to find $n$ satisfying $|a_0 - nb_0| < b_0$. Then

$$g_0S^{-1}T^n = (\begin{smallmatrix} a_0 & b_0 \\ c_0 & d_0 \end{smallmatrix})(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix})(\begin{smallmatrix} 1 & n \\ 0 & 1 \end{smallmatrix}) = (\begin{smallmatrix} -b_0a_0 \\ -d_0c_0 \end{smallmatrix})(\begin{smallmatrix} 1 & n \\ 0 & 1 \end{smallmatrix})$$

or

$$g_0S^{-1}T^n = (\begin{smallmatrix} -b_0 & a_0 - nb_0 \\ -d_0 & c_0 - nd_0 \end{smallmatrix})$$

is in $G$. But, then $g_0 \in G$ which contradicts the assumption on $g_0$.

Thus, $PSL_2(\mathbb{Z}) = \langle T, S \rangle$. □
The action corresponding to \( T \) and \( S \) are the translation \( z \rightarrow z + 1 \) and the inversion \( z \rightarrow \frac{-1}{z} \), respectively. Any linear fractional transformation \( g \) can be written as a word in \( S \) and \( T \) and their inverses: \( g = S^{a_1}T^{b_1}S^{a_2}T^{b_2}...S^{a_k}T^{b_k} \) where all of the \( a_t, b_t \) are integers. We can use the identities \( S^2 = I, (ST)^3 = (TS)^3 = I \) to further simplify the calculations.

1.3 Fundamental Domains and Special Polygons

We review much of the geometric terminology and the main results associated with fundamental domains for subgroups of \( PSL_2(\mathbb{Z}) \).

Whenever a group acts on a set, it divides the set into equivalence classes, where two points are said to be in the same equivalence class if there is an element of the group which takes one to the other. In particular, if \( G \) is a subgroup of \( \Gamma \), we say that two points \( z_1, z_2 \in \mathbb{H} \) are \( G \)-equivalent if there exists \( g \in G \) such that \( z_2 = gz_1 \).

Let \( F \) be a closed, simple connected region in \( \mathbb{H} \). We say that \( F \) is a fundamental domain for the subgroup \( G \) of \( PSL_2(\mathbb{Z}) \) if every \( z \in \mathbb{H} \) is \( G \)-equivalent to a point in \( F \), but no two distinct points \( z_1, z_2 \) in the interior of \( F \) are \( G \)-equivalent (two boundary points are permitted to be \( G \)-equivalent).

The most important example of a fundamental domain is shown in Fig. 1.1:

![Figure 1.1](image)

**FIGURE 1.1.** A fundamental domain of \( PSL_2(\mathbb{Z}) \).
Theorem 1.3.1. The region $F$ defined as

$$F = \left\{ z \in \mathbb{H} \mid -\frac{1}{2} \leq \text{Re} z \leq \frac{1}{2} \text{ and } |z| \geq 1 \right\}$$

is a fundamental domain for $PSL_2(\mathbb{Z})$.

Proof. First we prove that every $z \in \mathbb{H}$ is $PSL_2(\mathbb{Z})$-equivalent to a point in $F$. Idea is to use translations $T^{j_1}$ to move the point $z$ inside the strip $-\frac{1}{2} \leq \text{Re} z \leq \frac{1}{2}$. If it goes outside the unit circle, then it is in $F$. Otherwise use $S$ to move the point outside the unit circle and use a translation $T^{j_2}$ to bring it back inside the strip. We continue in this way until we get the point inside the strip and outside the unit circle. The rigorous proof is as follows.

Let $z \in \mathbb{H}$ be fixed. If $g = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in PSL_2(\mathbb{Z})$, then $\text{Im} \ g(z) = \frac{\text{Im} z}{|cz+d|^2}$. If $|cz+d| \geq 1$ then $\text{Im} \ g(z) \leq \text{Im} \ z$. Now, as $c$ and $d$ vary through all integers, the complex numbers $cz+d$ run through the lattice generated by 1 and $z$; but there are only finitely many lattice points inside the unit circle. Consequently, there are only finitely many complex numbers $z'$ of the form $z' = g(z)$ with $\text{Im} \ z' \geq \text{Im} \ z$. Thus, there is some $g = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in PSL_2(\mathbb{Z})$ such that $\text{Im} \ g(z)$ is maximal. Replacing $g$ by $T^j g$ for some suitable $j$, we can assume that $g(z)$ is inside the strip $-\frac{1}{2} \leq \text{Re} z \leq \frac{1}{2}$. Then, if $g(z)$ were not in $F$, i.e., if $|g(z)| < 1$, then, we would have $\text{Im} \ S(g(z)) = \frac{\text{Im} g(z)}{|g(z)|^2} > \text{Im} \ g(z)$, which contradicts the maximality of $\text{Im} \ g(z)$. Thus, there is a $g \in PSL_2(\mathbb{Z})$ such that $g(z) \in F$.

Now we prove that no two points in the interior of $F$ are $PSL_2(\mathbb{Z})$-equivalent. Let $z_1$, $z_2 \in F$ be $PSL_2(\mathbb{Z})$-equivalent and let $g = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in PSL_2(\mathbb{Z})$ be a matrix with $z_2 = g(z_1)$. We can assume $\text{Im} \ z_2 = \text{Im} \ g(z_1) \geq \text{Im} \ z_1$ by taking $g^{-1}$ instead of $g$, if necessary. Since $\text{Im} \ z_2 = \text{Im} \ g(z_1) = \frac{\text{Im} z_1}{|cz_1+d|^2} \geq \text{Im} \ z_1$, we must have $|cz_1+d| \leq 1$. If $z_1 = a_1 + b_1 i$, then $|cz_1+d| \leq 1$ implies $|ca_1 + d + cb_1 i| \leq 1$ or $(ca_1 + d)^2 + (cb_1)^2 \leq 1$. Thus, $|cb_1| \leq 1$ and since $b_1 = \text{Im} \ z_1 \geq \frac{\sqrt{3}}{2}$ we have $|c| \leq \frac{1}{b_1} \leq \frac{2}{\sqrt{3}} < 2$ that is $|c| \leq 1$. Adding $d$ to $cz_1$ just translates $cz_1$ to right or
left by $|d|$, so we need $|d| \leq 1$ for $cz_1 + d$ to be inside the unit circle. This leaves the cases:

i) $c = 0$, $d = \pm 1$;

ii) $c = \pm 1$, $d = 0$ with $z_1$ on the unit circle;

iii) $c = d = \pm 1$ and $z_1 = \frac{-1}{2} + \frac{\sqrt{3}}{2}i$;

iv) $c = -d = \pm 1$ and $z_1 = \frac{1}{2} + \frac{\sqrt{3}}{2}i$;

In case i) $g$ is a translation $g = T^j$, where $j \geq 0$. Such a $g$ can take a point in $F$ to another point in $F$ only if it is the identity or if $j = \pm 1$ and the points are on the two boundary lines $Re z = \pm \frac{1}{2}$. In case ii) $g$ must be of the form $g = T^m S$ with $m = 0$ and $z_1$ and $z_2$ on the unit circle, symmetrically located with respect to the imaginary axis ($z_2 = S z_1$) or $m = \pm 1$ and $z_1 = z_2 = \pm \frac{-1}{2} + \frac{\sqrt{3}}{2}i$. In case iii) we must have $|z_1 + 1| = |cz_1 + d| \leq 1$ Since adding 1 just translates a point in $F$ to the right by 1 the only possibility for $z_1$ is $z_1 = \frac{-1}{2} + \frac{\sqrt{3}}{2}i$. In case iv) we get $|z_1 - 1| = |cz_1 + d| \leq 1$ and the only possibility is $z_1 = \frac{1}{2} + \frac{\sqrt{3}}{2}i$. We conclude that we can never have $z_1$ and $z_2$ in the interior of $F$ unless $g = I$ and $z_1 = z_2$.

As a consequence of this proof we obtain the following useful fact.

**Theorem 1.3.2.** If $z \in F$, then $\Gamma_z = I$ except in the following cases:

1) $\Gamma_z = \{I, S\}$, if $z = i$;

2) $\Gamma_z = \{I, TS, (TS)^2\}$, if $z = e^{\frac{-2\pi i}{3}} = \frac{1}{2} + \frac{\sqrt{3}}{2}i$;

3) $\Gamma_z = \{I, ST, (ST)^2\}$, if $z = e^{\frac{2\pi i}{3}} = \frac{-1}{2} + \frac{\sqrt{3}}{2}i$;

Thus, in $PSL_2(\mathbb{Z})$ the elements $S$, $TS$, $ST$ generate cyclic subgroups of order 2, 3, and 3, respectively. They are the stabilizer subgroups of $i$, $\rho = e^{\frac{-2\pi i}{3}}$ and $\rho^2 = e^{\frac{2\pi i}{3}}$, respectively. The points in $H$, $PSL_2(\mathbb{Z})$-equivalent to $i$ and $\rho = e^{\frac{2\pi i}{3}}$ are the elliptic points of the upper half plane of order 2 and 3, respectively. They are the only points in $H$ having finite, nontrivial stabilizer subgroups.

We now look at fundamental domains for subgroups $\Gamma' \subset \Gamma$. Suppose that $[\Gamma : \Gamma'] = n < \infty$. Then $\Gamma$ can be written as a disjoint union of $n$ cosets $\Gamma = \bigcup_{i=1}^{n} g_i \Gamma'$. 

11
Then \( F' = \bigcup_{i=1}^{n} g_i^{-1} F \) will be a fundamental domain for \( \Gamma' \). Let \( z \in H \). Since \( F \) is a fundamental domain for \( \Gamma \), there is a \( g \in \Gamma \) such that \( g(z) \in F \). For some \( i \) we have \( g = g_i \gamma \) with \( \gamma \in \Gamma' \), and then \( \gamma(z) = g_i^{-1} g(z) \in g_i^{-1} F \subset F' \). It is easy to see that no two interior points of \( F' \) are \( \Gamma' \)-equivalent. Thus, \( F' \) is a fundamental domain for \( \Gamma' \). While there are many possible choices for \( g_i \), in practice the \( g_i' \)s are chosen so that the resulting \( F' \) is simply connected.

Let \( \Gamma^* = PSL_2^*(\mathbb{Z}) \) be the extended modular group, (i.e. the group of 2-by-2 integer matrices with determinant 1 or \(-1\), modulo \(< \pm I \> \)). We can extend the linear fractional transformations on the upper half plane \( H \) by defining

\[
    z \rightarrow g(z) = \frac{az + b}{cz + d} \quad \text{if } g = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in M_2(\mathbb{Z}) \quad \text{has } \det g = -1
\]

while as usual

\[
    g(z) = \frac{az + b}{cz + d} \quad \text{if } \det g = 1.
\]

\( PSL_2(\mathbb{Z}) \) has index 2 in \( \Gamma^* \). \( \Gamma^* = \Gamma \cup \gamma \Gamma \), where \( \gamma = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \) is the nontrivial coset representative with the corresponding action given by \( z \rightarrow -\bar{z} \).

Let \( D^* \) be the triangle with vertices \( i, \rho = e^{\pi i}, \) and \( \infty \). \( D^* \) is a fundamental domain for \( \Gamma^* \). Then, the fundamental domain \( F \) of \( PSL_2(\mathbb{Z}) \) defined in Theorem 1.3.1 consists of two copies of \( D^* \) sewn together along the sides. \( F \) is the union of \( D^* \) and the image of \( D^* \) under the action \( z \rightarrow -\bar{z} \).

The \( \Gamma^* \)-translates of \( D^* \) define the extended modular tessellation \( J^* \) of \( H \). The boundary of \( H \) consists of the real axis \( R \) and \( \infty \). Geodesics are semicircles orthogonal to the \( x \)-axis, including vertical lines. A complete geodesic is a geodesic that runs from a boundary point to another boundary point. The elements in the \( \Gamma^* \)-orbit of \( i \) are called the even vertices of \( J^* \) and those in the \( \Gamma^* \)-orbit of \( \rho \) are called the odd vertices of \( J^* \). As we saw above the \( \Gamma^* \)-orbit of \( \infty \) consists of the rational numbers, they are the cusps of \( J^* \). The elements in the \( \Gamma^* \)-orbit of the edge joining \( i \) to \( \infty \), respectively the edge joining \( \rho \) to \( \infty \) are called even edges, respectively odd edges. The edges in the \( \Gamma^* \)-orbit of the edge joining \( i \) to \( \rho \) are called \( f \)-edges.
The following proposition describes how the edges are paired together to form a complete geodesic. We will write $\infty$ as $\frac{1}{0}$ and all rational numbers $\frac{a}{b}$ will be in the reduced form, i.e. $\gcd(a, b) = 1$. Any integer $n$ will be written as $\frac{n}{1}$.

**Proposition 1.3.3.** [Kul91]  

**i)** The even edges come in pairs, each pair forming a complete hyperbolic geodesic. These geodesics are the ones with end-points $\frac{a}{c}$, $\frac{b}{d}$ satisfying $|ad - bc| = 1$. Each of these geodesics contains an even vertex. $\Gamma$ acts transitively on these geodesics and the stabilizer subgroup of $\Gamma$ preserving any one of these geodesics fixes the even vertex and is isomorphic to $\mathbb{Z}_2$.

**ii)** A pair of odd edges and a pair of $f$-edges form a complete hyperbolic geodesic. The geodesics obtained in this way are precisely the ones which have end-points $\frac{a}{c}$, $\frac{b}{d}$ satisfying $|ad - bc| = 2$. Each of these geodesics contains an even vertex and a pair of odd vertices. $\Gamma$ acts transitively on these geodesics and the stabilizer subgroup of $\Gamma$ preserving any one of these geodesics fixes the even vertex and is isomorphic to $\mathbb{Z}_2$.

**Proof.**  

**i)** It is sufficient to consider the geodesic joining $\infty$ to 0. It consists of two even edges ($\infty$, $i$) and $(i, 0)$ and contains the even vertex $i$. It also satisfy $|a_0d_0 - b_0c_0| = 1$, where $\frac{a_0}{c_0} = \infty \frac{1}{0}$ and $\frac{b_0}{d_0} = 0 = \frac{0}{1}$. If $g = (a b \ c d)$ is in $\Gamma$ or $\Gamma^*$ then the geodesic $(\frac{a}{c}, \frac{b}{d})$ is the translate of the geodesic $(\infty, 0)$, so $(\frac{a}{c}, \frac{b}{d})$ retains all properties of $(\infty, 0)$. Notice that $S = (\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix})$ is the matrix that sends the edge $(\infty, i)$ to $(0, i)$ and fixes the even vertex $i$. But $S^2 = I$ in $\Gamma^*$, so $< S > \simeq \mathbb{Z}_2$. Similarly, for the translated edge $(\frac{a}{c}, \frac{b}{d})$ we find that the stabilizer is isomorphic to $\mathbb{Z}_2$.

**ii)** This part is true for the geodesic joining $-1$ to 1. This geodesic, the upper half of the unit circle, consists of two paired odd edges $(1, \rho)$ and $(-1, \rho^2)$ and two paired $f$-edges $(i, \rho)$ and $(i, \rho^2)$. The side-pairing matrix is $A = (\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix})$. Since $A$ fixes the even vertex $i$ and $A^2 = I$ in $\Gamma^*$, we have $< A > \simeq \mathbb{Z}_2$. Then any $\Gamma$ or $\Gamma^*$ translate of $(-1, 1)$ will have the same properties.  

\[\Box\]
The complete geodesics that are unions of two even edges are called *even lines*, and the complete geodesics that are unions of two $f$-edges and two odd edges are called *odd lines*.

Let $P$ be a convex hyperbolic polygon with boundary $\partial P$ which is a union of even and odd edges. The following is assumed.

$P_1$) The even edges in $\partial P$ come in pairs, each pair forming an even line.

$P_2$) The odd edges in $\partial P$ come in pairs. The edges in each pair meet at an odd vertex making an internal angle of $\frac{2\pi}{3}$.

A *side-pairing* is an involution on edges so that no edge is carried into itself and the following rules hold.

$P_3$) An odd edge $a$ is paired to the odd edge $b$ which makes an internal angle of $\frac{2\pi}{3}$ with $a$.

$P_4$) Let $e, f$ be two even edges in $\partial P$ forming an even line. Then either $e$ is paired to $f$, or $e$ and $f$ form a free side of $P$ and this free side is paired to another free side of $P$.

The paired odd edges, the paired even edges and the free sides are simply called the *sides* of $P$. The points of intersection of the adjacent sides including those on $\partial \mathbb{H}$ are called the *vertices* of $P$.

$P_5$) 0 and $\infty$ are two of the vertices of $P$.

A convex hyperbolic polygon $P$ satisfying $P_1 - P_5$ is called a *special polygon*.

As an example, for $P_1$ and $P_4$ we have the even line joining 0 to $\infty$. We consider $i$ as a vertex of $P$ if and only if the two even edges meeting at in $i$, ($\infty$, $i$) and ($i$, 0) are paired. As an example of $P_2$ we have the odd edges ($\infty$, $\rho$) and ($\rho$, 0) making an angle of $\frac{2\pi}{3}$. Here $\rho$ will be a vertex of $P$. Neglecting the vertices of $P$ which
lie in $H$, we see that any two consecutive cusp-vertices $\frac{a}{c}, \frac{b}{d}$ of $P$ always satisfy $|ad - bc| = 1$.

If two sides $s_1$ and $s_2$ of $P$ are paired, then there is a unique element of $\Gamma$ which sends $s_1$ to $s_2$ in an orientation-reversing manner, where orientation of $P$ is the canonical counterclockwise orientation induced from that of $H$. The elements of $\Gamma$ obtained in this way are called the side-pairing transformations of $P$. The subgroup of $\Gamma$ generated by the side-pairing transformations of $P$ is denoted by $\Phi_P$.

A special polygon looks like the one in the figure:

![Figure 1.2. A special polygon.](image)

A fundamental domain $P$ for a subgroup $\phi \subseteq PSL_2(\mathbb{Z})$ is said to be admissible if the side pairings of $P$ form an independent set of generators for $\phi$. The following theorem is proved in [Kul91].

**Theorem 1.3.4.** Let $P$ be a special polygon, and $\Phi_P$ be the associated subgroup of $PSL_2(\mathbb{Z})$ generated by the side pairings of $P$. Then the elements formed by the side pairings give an independent system of generators for $\Phi_P$. Conversely, every subgroup $\Phi$ of finite index in $PSL_2(\mathbb{Z})$ admits an admissible fundamental domain which is a special polygon $P$, so that $\Phi = \Phi_P$.

### 1.4 Farey Symbols

Farey symbols are a useful way of representing special polygons and a key way of representing subgroups of the modular group, even if the representation is finite-to-one instead of one-to-one.

The $n$-th Farey sequence $F_n$ is the finite sequence of all rationals between 0 and 1, in increasing order, such that the denominators of the reduced form of the
rational are at most $n$. $F_n$ can be obtained recursively from $F_{n-1}$ as follows: if $\frac{a_i}{b_i}$ and $\frac{a_{i+1}}{b_{i+1}}$ are consecutive terms in $F_{n-1}$, then insert $\frac{a_i + a_{i+1}}{b_i + b_{i+1}}$ in between them if and only if $b_i + b_{i+1} = n$. Any two consecutive terms $\frac{a_i}{b_i}$ and $\frac{a_{i+1}}{b_{i+1}}$ of a Farey sequence satisfy the relation

$$|a_{i+1}b_i - a_ib_{i+1}| = 1.$$  

**Definition 1.4.1.** A generalized Farey sequence (gFS) is an expression of the form

$$\{\infty, x_1, x_2, \ldots, x_n, \infty\}$$

where

i) $x_1$ and $x_n$ are integers and some $x_i = 0$

ii) $x_i = \frac{a_i}{b_i}$ are rational numbers in their reduced form and ordered in increasing order, such that

$$|a_{i+1}b_i - a_ib_{i+1}| = 1, \quad i = 1, 2, \ldots, n - 1$$

It is convenient to set $x_0 = x_{n+1} = \infty$, and consider the $x'_i$s as forming a cyclic order. Recall that $\infty = \frac{1}{0}$. For calculation purposes the first term is sometimes denoted by $-\infty = \frac{-1}{0}$. In that case $a_{i+1}b_i - a_ib_{i+1} = 1$, for $i = 0, 1, \ldots, n$. The first and the last term are always identified, they are one and the same thing, $\infty$.

The importance of gFS is that the vertices of a special polygon, lying in $\mathcal{R} \cup \{\infty\}$, i.e. neglecting those in $\mathcal{H}$ form a gFS. On the other hand if we start with a gFS and take its convex hull in $\mathcal{H}$ we obtain a convex hyperbolic polygon which is a union of finitely many tiles of $J^*$. Thus, the gFS are in $1 - 1$ correspondence with such polygons which contain 0 and $\infty$ as their vertices.

We endow a gFS with an additional labelling structure on each consecutive pair of terms, an abstract analogue of encoding side-pairing information of a special polygon. First suppose $P$ is a special polygon and

$$\{\infty, x_1, x_2, \ldots, x_n, \infty\}$$
is the gFS formed by its vertices in $\mathbb{R} \cup \{\infty\}$.

If the complete hyperbolic geodesic joining $x_i$ to $x_{i+1}$ consists of two even edges which are paired we indicate this by

$$x_i \circlearrowleft x_{i+1}$$

and call it an even interval of the gFS.

If $x_i$ and $x_{i+1}$ are the endpoints of two paired odd edges we write this as

$$x_i \bullet x_{i+1}$$

and call it an odd interval of gFS.

If $x_i$ and $x_{i+1}$ are the endpoints of a free side $a$ of $P$ paired to the free side $b$ with endpoints $x_j$ and $x_{j+1}$ we indicate this by

$$x_i \xleftarrow{l} x_{i+1}, \quad x_j \xleftarrow{l} x_{j+1}$$

Here $l$ is a numerical symbol. If the $l$'s occur at all they will be numbered from 1 to some positive integer $r$ with different pairs of associated free sides having different numerical symbols. The specific numerical values for the labels have no significance. Each such pair

$$x_i \xleftarrow{l} x_{i+1}$$

is called a free interval of gFS.

A gFS without any reference to a special polygon $P$ endowed with an extra structure on each pair of consecutive terms of type (1.2)-(1.4) is called a Farey symbol. A typical example of Farey symbol may look like

$$\{ \infty , 0 \circlearrowleft \frac{1}{2} \circlearrowleft \frac{1}{2} \circlearrowleft 1 \circlearrowright \frac{1}{2} \circlearrowleft \frac{1}{2} \circlearrowleft \frac{1}{2} \circlearrowleft \infty \}$$

Conversely, given a Farey symbol whose underlying gFS is given by

$$\{ \infty, x_1, x_2, \ldots, x_n, \infty \}$$
we can construct a special polygon as follows. Let $P_0$ be the hyperbolic convex hull of the $x_i's$, $i = 0, 1, ..., n$. Suppose we have an odd interval

\[ x_i \rightarrow x_{i+1} \]

in our symbol. Then the complete hyperbolic geodesic joining $x_i$ to $x_{i+1}$ together with two odd edges situated inside $P_0$ form a hyperbolic triangle with angles $0$, $0$, $\frac{2\pi}{3}$. Adjoining such triangles for each odd interval in the Farey symbol we obtain a convex hyperbolic polygon $P$. The side-pairing is defined by reversing the process outlined above. Thus, we have

**Proposition 1.4.2.** [Kul91] The set of Farey symbols is in a natural 1-1 correspondence with the set of special polygons. In particular a Farey symbol determines a subgroup of finite index in $PSL_2(\mathbb{Z})$, and every subgroup of finite index in $\Gamma = PSL_2(\mathbb{Z})$ arises in this way. The map

\[
\{\text{Farey symbols}\} \longleftrightarrow \{\text{Subgroups of finite index in } PSL_2(\mathbb{Z})\}
\]

is finite-to-one.

### 1.5 Cusp Widths and Congruence Subgroups

We review some basic definitions regarding cusps, and also an important result due to Wohlfahrt relating the level of a congruence subgroup to the geometry of the cusps.

The even lines in $H$ give a tessellation of the upper half plane by *ideal triangles* (triangles having all angles equal to zero).

A *special triangle* is a $\Gamma$-translate of the triangle $D = (0, \rho, \infty)$. A special triangle is bounded by an even line and two odd edges, which make an internal angle of $\frac{2\pi}{3}$ with each other. Each special triangle also contains exactly one f-edge in its interior.

The tessellation of $H$ by ideal triangles is a sub-tessellation of $H$ by special triangles (three special triangles can be sewn together to form an ideal triangle).
Let \( \Phi \) be a subgroup of finite index in \( \Gamma \) and \( P \) a special polygon of \( \Phi \). The \textit{inequivalent cusps} of \( P \) are the \( \Phi \)-orbits of the free vertices (cusps) of \( P \). Let \( x \) be a cusp of \( \Phi \) and let \( \gamma \in \text{PSL}_2(\mathbb{Z}) \) such that \( \gamma(\infty) = x \). The smallest positive integer \( m \) such that \( \begin{pmatrix} 1 & m \\ 0 & 1 \end{pmatrix} \in \gamma^{-1}\Phi\gamma \) is called \textit{the width of a cusp} \( x \). Geometrically, this corresponds to the number of ideal triangles meeting at \( x \). The width of an inequivalent cusp of \( P \) is the sum of the widths of the cusps in the given \( \Phi \)-orbit plus half the number of special triangles intersecting the cusps in the \( \Phi \)-orbit. Geometrically, this corresponds to the number of even lines meeting at \( x \) in \( \Phi \setminus \mathbb{H} \).

Using Farey symbols we can easily calculate the width of a cusp. Let \( F \) be the Farey symbol of \( \Phi \)

\[
F = \{ -\infty, x_1, x_2, \ldots, x_n, \infty \}
\]

and \( P_\Phi \) be the associated special polygon.

As always we suppose that \( x_i = \frac{a_i}{b_i} \) are in reduced form, the \( x_i \)'s are in cyclic order and \( x_0 = -\infty = \frac{-1}{0} \) and \( x_{n+1} = \infty = \frac{1}{0} \) are identified. The cusp widths are given by

\[
d(x_i) = |a_{i-1}b_{i+1} - a_{i+1}b_{i-1}| \text{ for } i = 1, \ldots, n
\]

and

\[
d(x_0) = d(-\infty) = |a_1b_n - a_nb_1|.
\]

We define \( w(x_i) \) of \( x_i \) to be \( d(x_i) \), \( d(x_i) + \frac{1}{2} \), \( d(x_i) + 1 \), respectively, if \( x_i \) is incident to 0, 1, 2 odd intervals, respectively. If \( C \) denotes an inequivalent cusp of \( \Phi \), the width of \( C \) is given by the expression \( \text{w}(C) = \sum w(x_i) \), where \( x_i \) in the sum runs over the cusp vertices in the equivalence class of \( C \).

**Example 1.5.1.** Let \( \Phi \in \Gamma \) be the subgroup corresponding to the Farey symbol

\[
\{ -\infty \begin{array}{c} 0 \\ 1 \end{array} \begin{array}{c} 1 \\ 2 \end{array} \infty \}\}
\]

and special polygon \( P_\Phi \) shown in the figure 1.3.
FIGURE 1.3. Calculating the cusp widths.

We have

\[ d(\infty) = d(x_0) = |0 \cdot 1 - 2 \cdot 1| = 2, \quad d(0) = d(x_1) = |-1 \cdot 1 - 1 \cdot 0| = 1, \]

\[ d(1) = d(x_2) = |0 \cdot 1 - 2 \cdot 1| = 2 \quad \text{and} \quad d(2) = d(x_3) = |1 \cdot 0 - 1 \cdot 1| = 1. \]

Hence

\[ d(x_0) = 2, \quad d(x_1) = 1, \quad d(x_2) = 2, \quad d(x_3) = 1 \]

and

\[ w(x_0) = 2, \quad w(x_1) = 1 \frac{1}{2}, \quad w(x_2) = 3, \quad w(x_3) = 1 \frac{1}{2}. \]

Then 0, 1 and 2 or \( x_1, x_2 \) and \( x_3 \) are equivalent; this gives an inequivalent cusp of width

\[ 1 \frac{1}{2} + 3 + 1 \frac{1}{2} = 6, \]

and \( x_0 = \infty \) is equivalent to itself. So the two inequivalent cusps have widths 2 and 6.

The geometric level of \( \Phi \) is the least common multiple of the inequivalent cusp widths. For instance, in the example above the geometric level of \( \Phi \) is \( lcm(2, 6) = 6 \).

Recall that a subgroup \( \Phi \subseteq \Gamma \) is a congruence subgroup if it contains the principal congruence subgroup \( \Gamma(N) \) for some positive integer \( N \). The arithmetic level of a congruence subgroup \( \Phi \) is the least \( N \) such that \( \Gamma(N) \subseteq \Phi \). While the arithmetic level is only defined for congruence subgroups, the geometric level is defined for all subgroups. The following theorem connecting the two concepts is proved in [Woh64].
Theorem 1.5.2. If $\Phi \subseteq \Gamma$ is a congruence subgroup, then the arithmetic level of $\Phi$ is equal to the geometric level.

We can thus speak of the level of a subgroup. This equivalence is the key to the algorithm for congruence testing, given in [LLT95] and restated here in chapter 3.

1.6 The Side Pairing Matrices

We give here the matrices associated to the side-pairings of a special polygon.

Let $F$ be a Farey symbol and let $\Phi_F$ be the subgroup of finite index in $\Gamma$ determined by $F$. Let $x_i = \frac{a_i}{b_i}, x_{i+1} = \frac{a_{i+1}}{b_{i+1}}$ and $x_j = \frac{a_j}{b_j}, x_{j+1} = \frac{a_{j+1}}{b_{j+1}}$ be two different pairs of consecutive vertices of a special polygon $P$ corresponding to $\Phi_F$. We will determine the side-pairing matrices of $P$ when

is an even interval, a free interval

paired to the free interval

or an odd interval

If $(x_i, x_{i+1})$ is an even line, then the even line $(\infty, 0)$ is paired to $(x_i, x_{i+1})$ in an orientation-reversing manner by the matrix

$$A(x_i, x_{i+1}) = \begin{pmatrix} a_{i+1} & a_i \\ b_{i+1} & b_i \end{pmatrix}$$

which sends $\infty$ to $x_{i+1} = \frac{a_{i+1}}{b_{i+1}}$ and 0 to $x_i = \frac{a_i}{b_i}$:

$$(\infty, 0) \overset{A(x_i, x_{i+1})}{\rightarrow} \left( \frac{a_i}{b_i}, \frac{a_{i+1}}{b_{i+1}} \right)$$
Now the matrix $S = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ just reverses the orientation of $(\infty, 0)$ by sending $\infty$ to 0 and 0 to $\infty$:

$$S (\infty, 0) \rightarrow (\infty, 0)$$

For the first two cases, the side-pairing matrices can be constructed by sending the first side to $(0, \infty)$, then reversing the orientation via $S$, and then sending $(\infty, 0)$ to the second side. When

$$x_i \xrightarrow{g} x_{i+1}$$

is an even interval we get side-pairing matrix:

$$A(x_i, x_{i+1})^{-1} \xrightarrow{S} A(x_i, x_{i+1}) \xrightarrow{S} A(x_j, x_{j+1})^{-1}$$

The side-pairing matrix is given by

$$A_2(x_i) = A(x_i, x_{i+1})SA(x_i, x_{i+1})^{-1} = \left( \begin{array}{cc} a_{i+1} & a_i \\ b_{i+1} & b_i \end{array} \right) \left( \begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array} \right) \left( \begin{array}{cc} a_{i+1} & a_i \\ b_{i+1} & b_i \end{array} \right)^{-1}$$

or

$$A_2(x_i) = \left( \begin{array}{cc} a_{i+1}b_{i+1} + a_ib_i & -a_{i+1}^2 - a_i^2 \\ b_{i+1}^2 + b_i^2 & -a_{i+1}b_{i+1} - a_ib_i \end{array} \right)$$  \tag{1.5}$$

When we pair two free sides $(x_i, x_{i+1})$ and $(x_j, x_{j+1})$ we get the side-pairing matrix

$$A(x_i, x_j) = A(x_j, x_{j+1})SA(x_i, x_{i+1})^{-1} = \left( \begin{array}{cc} a_{j+1} & a_j \\ b_{j+1} & b_j \end{array} \right) \left( \begin{array}{cc} 0 & -1 \\ 1 & 0 \end{array} \right) \left( \begin{array}{cc} a_{j+1} & a_j \\ b_{j+1} & b_j \end{array} \right)^{-1}$$

or

$$A(x_i, x_j) = \left( \begin{array}{cc} a_{j+1}b_{j+1} + a_jb_j - a_{j+1}a_{i+1} - a_ja_i \\ b_{j+1}b_{i+1} + b_jb_i - a_{i+1}b_{j+1} - a_ib_j \end{array} \right)$$  \tag{1.6}$$

22
By reversing the orientation these side-pairing matrices just "switch" the endpoints: $A_2(x_i)$ sends $x_i$ to $x_{i+1}$ and $x_{i+1}$ to $x_i$ whereas $A(x_i, x_j)$ sends $x_i$ to $x_{j+1}$ and $x_{i+1}$ to $x_j$.

The case of an odd interval is more complicated. Instead of $S$ we will use $U = \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix}$, the matrix of order 3 in $\Gamma$ that fixes $\rho = e^{\pi i}$ and pairs the odd edges $(\infty, \rho)$ and $(\rho, 0)$. The side-pairing matrix of order 3 corresponding to the odd interval

\[\begin{array}{c}
\bullet \\
\overline{x_i x_{i+1}}
\end{array}\]

is given by

\[A_3(x_i) = A(x_i, x_{i+1})UA(x_i, x_{i+1})^{-1} = \begin{pmatrix} a_{i+1}+1 & a_i \\ b_{i+1}+1 & b_i \end{pmatrix} \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} \begin{pmatrix} a_{i+1}+1 & a_i \\ b_{i+1}+1 & b_i \end{pmatrix}^{-1}
\]

or

\[A_3(x_i) = \begin{pmatrix} a_{i+1}b_{i+1}+a_i b_{i+1}+a_i b_i & -a^2_{i+1}-a_{i+1}a_i-a_i-b_i^2-b_i^2 \\ b_{i+1}+b_i+b_i+b_i^2 & -a_{i+1}b_{i+1}-a_{i+1}b_i-a_i b_i \end{pmatrix}.
\]

(1.7)

**1.7 Geometric Invariants of a Subgroup and Their Relations**

For a subgroup of finite index $\Phi \in \Gamma$ we have the following geometric invariants:

i) $e_2 = \text{the number of branch points of } H \to \Phi \backslash H \text{ of order } 2$

ii) $e_3 = \text{the number of branch points of } H \to \Phi \backslash H \text{ of order } 3$

iii) $d = (\Gamma : \Phi)$ the degree of the branched covering $\Gamma \backslash H \to \Phi \backslash H$

iv) $g = \text{the genus of } \Phi \backslash H$

v) $t = \text{the number of inequivalent cusps of } \Phi \backslash H$

vi) $w(C_k) = \text{the width of the } k\text{-th inequivalent cusp}$

vii) $r = \text{the rank of } \pi_1(\Phi \backslash H)$
Most of these can be reformulated in the language of group theory as follows:

\[ e_2 = \text{the number of conjugacy classes of elements of order 2, or the number of generators of order 2 in any independent set of generators} \]

\[ e_3 = \text{similarly for order 3 elements} \]

\[ d = \text{the index of } \Phi \text{ in } \Gamma \]

\[ r = \text{the rank of the free factor of } \Phi, \text{ or the number of generators of infinite order in any independent set of generators} \]

Many of these invariants can be read directly from the labelling of a Farey symbol corresponding to \( \Phi \): \( e_2 \) is the number of even intervals \( \circ \), \( e_3 \) is the number of odd intervals \( \bullet \), \( r \) is the number of pairs of free sides which are glued together.

From the topology of surfaces, we know

\[ r = 2g + t - 1. \]

Since we can get \( r \) directly from the side pairings of a special polygon and calculate \( t \) relatively easily, we can find the genus of a subgroup given its special polygon.

The Riemann-Hurwitz formula relates these invariants by

\[ d = 3e_2 + 4e_3 + 6r - 6 = 3e_2 + 4e_3 + 12g + 6t - 12. \]

Also,

\[ d = \sum_{k=1}^{t} w(C_k) \]

where \( C_k's, k = 1, \ldots, t \) are the inequivalent cusps of \( \Phi \).

Another useful formula relating these invariants is given by the following proposition.

**Proposition 1.7.1.** Let \( F \) be a Farey symbol with underlying gFS given by \( \{\infty, x_1, x_2, \ldots, x_n, \infty\} \) and let \( \Phi_F \) be the corresponding subgroup. If \( F \) contains \( e_3 \) odd intervals, then the index \( d \) of \( \Phi_F \) in \( \Gamma \) is given by \( d = 3n + e_3 - 3 \).
Proof. The number of free intervals in $F$ is $n + 1 - e_2 - e_3$. So, $r = \frac{1}{2}(n+1 - e_2 - e_3)$ and by Riemann-Hurwitz formula we have

$$d = 3e_2 + 4e_3 + 6\left\{\frac{1}{2}(n + 1 - e_2 - e_3) - 1\right\} = 3n + e_3 - 3.$$
Chapter 2
Generating Subgroups of a Certain Index

2.1 Trivalent Diagrams and Tree Diagrams

We review the relationships between bipartite cuboid graphs, tree diagrams and special polygons. Then we will describe a graph-theoretic approach to finding the conjugacy classes of subgroups of $PSL_2(\mathbb{Z})$. We start by giving the definitions of bipartite cuboid graphs (also called trivalent diagrams), marked trivalent diagrams and tree diagrams.

**Definition 2.1.1.** A **bipartite cuboid graph** is a finite connected graph whose vertex set is partitioned into subsets $V_0$ and $V_1$ such that:

1. Every vertex in $V_0$ has valence (degree) 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 $V_1$.

Note that a bipartite cuboid graph does not need to be simple, it can have multiple edges between two vertices. The cyclic orientation around the trivalent vertices is assumed to be the standard counterclockwise orientation of the plane. Also an isomorphism between two bipartite graphs is supposed to preserve the prescribed cyclic orders corresponding to the trivalent vertices.

**Definition 2.1.2.** A **marked trivalent diagram** is a trivalent diagram with a distinguished edge called the **marked edge**.

A morphism of two marked trivalent diagrams sends a marked edge to a marked edge.

26
Definition 2.1.3. A tree diagram is a tree $T$ containing at least one edge and satisfying:

1. Every internal vertex has degree 3.

2. There is a prescribed cyclic order on the edges incident at each internal vertex.

3. The terminal vertices are partitioned into two possible empty subsets $W$ and $B$ where the vertices in $W$ are called white vertices and those in $B$ are called black vertices.

4. There is an involution $\sigma$ on $W$.

$T$ can be embedded in the plane so that the cyclic order on the edges at each internal vertex coincides with the one induced by the orientation of the plane. Tree diagrams are thus best represented without explicitly indicating the cyclic order on the edges. An isomorphism of tree diagrams is an isomorphism of the underlying trees. We represent the white vertices as $\circ$ and black vertices as $\bullet$. The distinct white vertices related by $\sigma$ are given the same numerical label, being understood that the unlabelled white vertices are fixed by $\sigma$ and different pairs of distinct white vertices related by $\sigma$ carry different labels.

Now we study the relationship between special polygons and tree diagrams.

Given a special polygon $P$, we construct a tree diagram $T$. Let $T$ be the graph of $f$-edges in $P$. We do not count the even vertices in $\text{int} \ P$ as vertices, we just ignore them. The white vertices respectively the black vertices are the even vertices respectively the odd vertices on the boundary $\partial P$. The involution on the white vertices is given by the side-pairings of $P$. The cyclic order on the edges incident to the vertices of degree 3 is induced by the orientation of $P$. This turns $T$ into a tree diagram.

Conversely, let $T$ be a tree diagram. On all edges joining two internal vertices or joining an internal vertex with a black vertex we introduce a new divalent vertex.
$T$ must have at least one white vertex or one black vertex. Suppose it has a white vertex $v$. Then identify the edge incident to $v$ with the $f$-edge $(i, \rho)$ identifying $v$ with $i$ (if we have a black vertex, identify it with $\rho$). $T$ can then by developed into a tree of $f$-edges so that the cyclic orders on the edges incident at the vertices of valence 3 in $T$ match with the ones induced by the orientation of $H$. At the image of a white vertex $w$ in this development assign the even line passing through $w$. These even edges are paired if $w$ is fixed by the involution $\sigma$. Otherwise this complete geodesic will be considered a free side, which will be paired with the free side constructed at $\sigma(w)$. Similarly, at the image of a black vertex of degree 1 incident to the unique edge say $e$, assign those two odd edges which make an angle $\frac{\pi}{3}$ with the image of $e$. These odd edges are paired. Then these even sides, odd sides and free sides together with their pairing define a special polygon. It is clear that the special polygon associated to a tree diagram $T$ depends on the initial choice of the vertex $v$. So we have a well-defined, finite-to-one map between tree diagrams and special polygons.

Now we study the relationship between tree diagrams and trivalent diagrams.

To turn a tree diagram $T$ into a bipartite cuboid graph $G$, first for all white vertices $v$, identify $v$ with $\sigma(v)$. On all edges joining two internal vertices or an internal vertex to a black vertex introduce a new vertex of valence 2. These new vertices and the white vertices constitute $V_0$, while the vertices of valence 3 and the black vertices constitute $V_1$. The cyclic orders of the trivalent vertices in $V_1$ come from the cyclic orders of the internal vertices of $T$.

Conversely, let $G$ be a bipartite cuboid graph. If its cycle-rank (number of cycles) is $r$ we can choose $r$ vertices of valence 2 in $V_0$ such that cutting $G$ along these vertices we obtain a tree $T$. Corresponding to this $r$ cuts we have $2r$ terminal vertices in $T$. These $2r$ terminal vertices and the (terminal) vertices of valence 1 in $V_0$ correspond to the white vertices of $T$. We do not count the remaining vertices
of valence 2 in $V_0$ as vertices of $T$. Set up the involution $\sigma$ as fixing the terminal vertices of valence 1 in $V_0$ and interchanging the two vertices obtained at each of the $r$ cuts. The vertices of valence 1 in $V_1$ correspond to the black vertices. The cyclic order around the trivalent vertices corresponds directly to their cyclic order in $G$. It is clear that the tree diagram $T$ depends on the choice of the $r$ cuts. Thus, we have a well-defined, finite-to-one map from the isomorphism classes of tree diagrams onto those of bipartite cuboid graphs.

Now we describe a method for finding all trivalent diagrams of size $m$, where $m$ is given. The method assumes that all the diagrams of size 1 to $m - 1$ are given. We basically have two operations to apply to the diagrams. First one is just connecting a new edge $n_{\text{edge}}$ to an edge $e_{\text{diag}}$ of a diagram of size $m - 1$. The only requirement for the edge $e_{\text{diag}}$ is to have the white vertex of degree 1, because the connection is made there. We apply this procedure to all possible candidates $e_{\text{diag}}$ and to all diagrams of size $m - 1$. The second operation is adding three adjacent edges to diagrams of size $m - 3$. Since we add three adjacent edges we either add an edge joined to a 2-cycle or we add a tripod. When we add an edge joined to a 2-cycle, the only possibility for the connection point is a white vertex of degree 1. So, for each white vertex of degree 1 of each diagram of size $m - 3$ we get a new diagram of size $m$. When we add a tripod we have multiple choices for the connection points. We can have only a connection point, a white vertex $v$ of degree 1. In this case we just connect the tripod to $v$. Apply this to all white vertices of degree 1 and to all diagrams of size $m - 3$. We can have two connection points $v_1$ and $v_2$, both white vertices of degree 1. In this case we connect two white vertices of the tripod $t_1$ and $t_2$, one with $v_1$ and one with $v_2$, while the third white vertex of the tripod $t_3$ remains unconnected. Because of the cyclic orientation around the black vertex of the tripod $b_{tr}$, we can get two different diagrams. One will have the cycle around $b_{tr}$ as $(b_{tr}, t_1), (b_{tr}, t_3), (b_{tr}, t_2)$, the other will have the cycle around
$b_{tr}$ as $(b_{tr}, t_1), (b_{tr}, t_2), (b_{tr}, t_3)$. We apply these connections for all possible pairs $(v_1, v_2)$ and for all diagrams of size $m - 3$. When we have three connection points $(v_1, v_2, v_3)$, all white vertices of degree 1, we again get two different diagrams. One will have the cycle around $b_{tr}$ as $(b_{tr}, t_1), (b_{tr}, t_3), (b_{tr}, t_2)$, the other will have the cycle around $b_{tr}$ as $(b_{tr}, t_1), (b_{tr}, t_2), (b_{tr}, t_3)$. It is like in the situation of two connection points, the only difference is that now $t_3$ gets connected to a vertex $v_3$. We apply this type of connections to all possible triplets $(v_1, v_2, v_3)$ and to all diagrams of size $m - 3$.

To construct all diagrams of size $m$, we just start with the diagrams of size 1, 2, and 3. These are well-known. It is just an edge for the diagram of size 1, two adjacent edges for the diagram of size 2 and we have two diagrams of size 3, a tripod and an edge joined to a 2-cycle. Then we construct the diagrams of size 4 by adding an edge to the diagrams of size 3 and adding three edges to the diagram of size 1. If we get isomorphic diagrams, we just keep one representative from each isomorphism class, removing the rest. So, we get all diagrams of size 4. Then we find all diagrams of size 5. We add an edge to the diagrams of size 4 and add three edges to the diagrams of size 2. Again remove the duplicates. We continue until we get all diagrams of size $m$.

### 2.2 Vidal’s Classification Principle

Now we state without proof the following classification principle of Vidal:

**Theorem 2.2.1.** [Vid06] a) There is a bijective correspondence between isomorphism classes of bipartite cuboid graphs and conjugacy classes of subgroups of $\Gamma$.

b) There is a bijective correspondence between the marked trivalent diagrams and the subgroups of $\Gamma$.

While we do not prove the theorem, we will show how a subgroup of $PSL_2(\mathbb{Z})$ produces a bipartite cuboid graph and how a bipartite cuboid graph gives rise to a conjugacy class of subgroups of $\Gamma$. To get a trivalent diagram from a subgroup
\( \Phi \subseteq \Gamma \) is quite simple. Find a special polygon for \( \Phi \), then find its corresponding tree diagram and convert it to a bipartite cuboid graph. We do not deal with how to construct a special polygon for a given subgroup. We assume that the special polygon of \( \Phi \) is given by a Farey symbol and from the Farey symbol we will get the corresponding trivalent diagram. The example below show the process.

**Example 2.2.2.** Let \( \Phi \subseteq \Gamma \) be the subgroup with Farey symbol given by \( gFS \{ \infty, 0, \frac{1}{3}, \frac{1}{2}, 1, \infty \} \) and labels \{ 1, \bullet, 2, 2, 1 \}. To construct the special polygon of \( \Phi \) we use the \( \Gamma^* \)-translates of the triangle \( D \) with vertices 0, \( \rho \) and \( \infty \). Or equivalently, we take the even edges between any cusps \( x_i = \frac{a_i}{b_i} \) and \( x_j = \frac{a_j}{b_j} \) for which \( |a_i \cdot b_j - a_j \cdot b_i| = 1 \) and the odd edges inside the ideal triangle with vertices 0, 1 and \( \infty \). See the following figure. The even vertices are represented by \( \circ \) and the odd ones by \( \bullet \). We have marked the edge \( e_1 \), the edge joining \( i \) to \( \rho \), and we have put the Farey symbol labels.

![FIGURE 2.1. The tessellation inside the polygon.](image-url)
Ignoring the cusp values we get the tree diagram

![Tree Diagram](image)

**FIGURE 2.2.** The tree diagram.

or

![Rotated Tree Diagram](image)

**FIGURE 2.3.** The rotated tree diagram.

Now to get the corresponding bipartite cuboid graph, we just identify the vertices with the same numerical label (we are making cycles).

![Bipartite Cuboid Graph](image)

**FIGURE 2.4.** The bipartite cuboid graph.

Now let us interpret Vidal’s theorem. Let us denote the bipartite cuboid graph we have got in the above example by $T_{\Phi}$. Here we ignore that one edge is marked.
$T_\Phi$ corresponds to the conjugacy class of subgroups of $\Gamma$ containing $\Phi$. Let us denote this conjugacy class by $\Gamma_\Phi$. The index $[\Gamma : \Phi]$ is given by the size (number of edges) of $T_\Phi$. In our case $[\Gamma : \Phi] = 10$. Marking an edge of the graph produces a certain subgroup from $\Gamma_\Phi$. If we mark a different edge of $T_\Phi$ and identify it with the edge joining $i$ to $\rho$, then by reversing the above process we get a subgroup $\Phi'$ conjugate to $\Phi$. Cutting the cycles at different points gives rise to the same subgroup $\Phi$, but a different Farey symbol. The following graph corresponds to a different Farey symbol (since the cut of the cycle between $C$ and $B$ is at a different place), and a different subgroup $\Phi'$ (since the marked edge $e_1$ is in a different place). Nonetheless $\Phi$ and $\Phi'$ are conjugate because their corresponding bipartite cuboid graph is the same, the graph $T_\Phi$.

![FIGURE 2.5. Changing the marked edge and the cycle cuts.](image)

Therefore, to find the conjugacy classes of subgroups of $\Gamma$ having a certain index $d$, it suffices to find all bipartite cuboid graphs of size $d$. Once we find a bipartite cuboid graph of size $d$ we mark an edge and by reversing the process given in the example above we can find the special polygon of the corresponding subgroup. We try to simplify this process, though. All we need to find is the Farey symbol of the corresponding subgroup. To find the Farey symbol labels we just need the labels of the degree-1 vertices. We start with the degree-1 vertex of the marked edge $e_1$ and read the labels counterclockwise. If we look again at the figure 2.3 we begin with label 1 corresponding to the edge $e_1$ and then we get • corresponding to vertex $D$, then get the labels 2 and 2 for the vertices lower right to $C$ and lower left to $D$.
respectively, and then finally we get label 1.

\[
\begin{array}{cc}
D & C \\
2 & 2 \\
B & A
\end{array}
\]

FIGURE 2.6. Reading the Farey labels.

This retrieves the sequence of Farey labels \{1, \bullet, 2, 2, 1\}.

Now we need a method for finding the cusps corresponding to each vertex of degree 1. The method is a little bit more laborious. It is based on the fact that any \(f\)-edge \(e\) is the image of the edge \(e_1 = (i, \rho)\):

\[
e = g(e_1) \text{ for some } g = \left(\begin{array}{cc} a & b \\ c & d \end{array}\right) \in PSL_2(\mathbb{Z}).
\]

We start with the triangle \(D\) with vertices \(0, \rho\) and \(\infty\) and then we translate it by \(g \in \Gamma\)

\[
\begin{array}{cc}
\bullet & e_1 \\
0 & 1/2
\end{array}
\]

FIGURE 2.7. The \(D\) triangle.

\[
g = \left(\begin{array}{cc} a & b \\ c & d \end{array}\right)
\]

\[
e = ge_1
\]

\[
\begin{array}{cc}
\bullet & a \\
\bullet & e \rightarrow \bullet
\end{array}
\]

FIGURE 2.8. A generic \(f\)-edge and the associated cusps.
Edge $e$ has cusps $\frac{b}{d}$ and $\frac{c}{e}$ and these should be considered as added to the even (open) vertex:

![Figure 2.9. The rotated $f$-edge.](image)

For instance, the edge $e_1$ has cusps $\infty$ and 0:

![Figure 2.10. The edge $e_1$.](image)

If we know the cusps at an even vertex $\circ$ we can compute the cusps at nearby even vertices. Suppose that the cusps of the edge $e$ are $\frac{a}{b}$ and $\frac{c}{d}$ with $\frac{a}{b} < \frac{c}{d}$. Then the part of the extended modular tesselation around the edge $e$ and the part of the corresponding bipartite cuboid graph can have one of the following three configurations:

![Figure 2.11. Adding a cusp between two given cusps.](image)
Since \( \frac{a-c}{b-d} = \frac{c-a}{d-b} \), we see that the white vertices in the last two cases will have the same cusps. We get the rules for calculating the cusps of the nearby white vertices.

**Rule 1** If \( \frac{a}{b} < \frac{c}{d} \) or \( a \cdot d < b \cdot c \) then the cusps of the nearby white vertices are as follows:

![Figure 2.14. Rule 1.](image)
Rule 2  If \( \frac{a}{b} > \frac{c}{d} \) or \( a \cdot d > b \cdot c \) then

![Diagram](image)

FIGURE 2.15. Rule 2.

We will always assume that the edges incident at a black vertex of valence 3 are arranged counterclockwise in a cyclic order. We illustrate the entire process of finding the Farey symbol from a marked trivalent diagram with another example.

Example 2.2.3. Let us consider the marked trivalent diagram we drew to emphasize the way a subgroup is changing when we change the marked edge:

![Diagram](image)

FIGURE 2.16. Finding the cusps.

To get the Farey labels, we just read the labels of valence-1 vertices, beginning with the vertex corresponding to the edge \( e_1 \) and then going counterclockwise around the diagram. We get the Farey labels \( \{1, 1, \circ, 2, 2\} \).

Now, to get the cusp corresponding to each vertex, we start with the edge \( e_1 \):

![Diagram](image)

FIGURE 2.17. The cusps of \( e_1 \).
Since $0 < \infty$ or $\frac{0}{1} < \frac{1}{0}$, we have $0 \cdot 0 < 1 \cdot 1$. We apply rule 1 to get the cusps at nearby vertices. The new cusp will be \(\frac{0+1}{1+0} = \frac{1}{1}\)

![Diagram of cusps around B.](image)

FIGURE 2.18. The cusps around B.

Similarly, we calculate the cusps corresponding to the vertices adjacent to the vertex A. Since $1 < \infty$, $1 \cdot 0 < 1 \cdot 1$ and we apply rule 1. The new cusp will be \(\frac{1+1}{0+1} = \frac{2}{1}\).

Likewise, we find the cusps of the vertices adjacent to C. Again $0 \cdot 1 < 1 \cdot 1$ and when we apply rule 1 we get the new cusp \(\frac{0+1}{1+1} = \frac{1}{2}\). Finally, vertex D is a black vertex of valence 1. To find its cusps, we just copy the cusps of the adjacent white vertex. We get the following diagram:

![Diagram of all cusps.](image)

FIGURE 2.19. All the cusps.

To get the Farey sequence corresponding to the diagram we just read the cusps counter-clockwise beginning with the cusps at the white vertex of \(e_1\). Every pair of cusps at a valence-1 vertex gives an interval of the Farey symbol or an edge of the special polygon.
We get the Farey sequence \( \{ \infty, \frac{0}{1}, \frac{1}{2}, \frac{1}{1}, \frac{2}{1}, \infty \} \) or \( \{ \infty, 0, \frac{1}{2}, 1, 2, \infty \} \). When we put together the Farey labels and the Farey sequence we get the Farey symbol:

\[
\{ \infty \ 0 \ \frac{1}{2} \ 1 \ 2 \ 2 \ \infty \}
\]

Using the Farey symbol, we can calculate the generators of the corresponding subgroup. We just use the side-pairing matrices corresponding to a free side, even side or odd side, matrices given by (1.5)-(1.7). For our Farey symbol

\[
\{ \infty \ 0 \ \frac{1}{2} \ 1 \ 2 \ 2 \ \infty \}
\]

we have three generators, one corresponding to free sides labeled 1, one to the odd side labeled \( \bullet \), and one to the free sides labeled 2.

Using (1.6) with \( x_i = -\frac{1}{6}, x_{i+1} = 0 \) and \( x_j = \frac{0}{1}, x_{j+1} = \frac{1}{2} \) we get \( g_1 = \left( \begin{array}{cc} 1 & 0 \\ 2 & 1 \end{array} \right) \).

For the odd side between \( \frac{1}{2} \) and \( \frac{1}{1} \) we use (1.7) to get \( g_2 = \left( \begin{array}{cc} 4 & -3 \\ -7 & -5 \end{array} \right) \). And for the free sides labeled 2, using (1.6) with \( x_i = \frac{1}{1}, x_{i+1} = \frac{2}{1} \) and \( x_j = \frac{2}{1}, x_{j+1} = \frac{1}{0} \) we get \( g_3 = \left( \begin{array}{cc} 3 & -4 \\ 1 & -1 \end{array} \right) \).

So, the subgroup \( G \) corresponding to the trivalent diagram is the subgroup generated by \( g_1, g_2 \) and \( g_3 \):

\[
G = \langle \left( \begin{array}{cc} 1 & 0 \\ 2 & 1 \end{array} \right), \left( \begin{array}{cc} 4 & -3 \\ -7 & -5 \end{array} \right), \left( \begin{array}{cc} 3 & -4 \\ 1 & -1 \end{array} \right) \rangle
\]
Chapter 3
Membership Tests for Matrices in $PSL_2(\mathbb{Z})$

3.1 Lang, Lim and Tan’s Algorithm

In this chapter we give two algorithms for determining the membership of a matrix to a subgroup of $\Gamma$ given by a Farey symbol. As an application we use the algorithms to find small-index noncongruence subgroups of $\Gamma$. By small-index we refer to subgroups of index less than 10. We start with the Lang, Lim and Tan’s algorithm [LLT95]. We saw in the previous chapter that to any subgroup $\Phi \subseteq \Gamma$ corresponds a tree diagram $T_\Phi$ given by the $f$-edges inside a special polygon $P_\Phi$ of $\Phi$. We turn any such diagram into a $\mathbb{Z}$-tree by specifying the distance between any two adjacent vertices to be 1. We have the following proposition:

**Proposition 3.1.1.** [LLT95] If $P$ is a special polygon in $H$ and $l$ is an even line then either $l \subseteq P$ or $l \cap P = \emptyset$.

*Proof.* Let $A$ be the even vertex of $l$. $A$ is an elliptic point of order 2 and in the tessellation of $H$, $A$ is a point of valence 2. If $l \cap P \neq \emptyset$ then $A \in P$ and then $A$ belongs to an even line $l' \subseteq P$. Since the degree of $A$ is 2, the end points of $l$ must coincide with the end points of $l'$. These are just the vertices adjacent to $A$ in the tessellation of $H$. Hence $l = l' \subseteq P$. 

**Definition 3.1.2.** Let $l$ be an even line, $P$ a special polygon and $T$ the $\mathbb{Z}$-tree of all $f$-edges in $H$. We define the distance $d(l, P)$ between $P$ and $l$ to be the distance along the tree $T$ between the sub-tree $P \cap T$ and the vertex $l \cap T$.

**Example 3.1.3.** Let us consider again the tree diagram from the example in chapter 2 in figure 3.1 and let $l$ be the even line $l = (\frac{2}{3}, 1)$

When we add the line $l = (\frac{2}{3}, 1)$ we get the picture in figure 3.2. Hence $d(l, P) = d(G, E) = 2$. 

40
Now we describe the Lang, Lim and Tan’s algorithm. Let $\Phi \subseteq \Gamma$ be a subgroup of the modular group and let $P_\Phi$ be a special polygon of $\Phi$ with Farey symbol having the generalized Farey sequence

$$\{-\infty, x_1, x_2, \ldots, x_n, \infty\}.$$
As always $x_0 = -\infty$ and $x_{n+1} = \infty$ are identified and $x_t = 0$ for some $t$ with $1 \leq t \leq n$. Let $\{g_i\}_{i \in I}$ be the set of generators of $\Phi$ corresponding to the side pairings of $P_\Phi$ and let $g = \left( \begin{smallmatrix} a_0 & b_0 \\ c_0 & d_0 \end{smallmatrix} \right) \in PSL_2(\mathbb{Z})$. The algorithm decides whether $g$ belongs to the subgroup $\Phi = \langle \{g_i\}_{i \in I} \rangle$ or not and if yes, then it gives a decomposition of $g$ as a reduced word in $g_i$'s.

Notice that $g = \left( \begin{smallmatrix} a_0 & b_0 \\ c_0 & d_0 \end{smallmatrix} \right)$ sends the oriented even line $(0, \infty)$ to the even line $l_0 = \left( \begin{smallmatrix} b_0 \\ d_0 \end{smallmatrix} \right)$. The algorithm does the following: if $d(l_0, P_\Phi) > 0$, then it finds an element $g_{i_1} \in \{g_i\}_{i \in I}$ such that $g_{i_1} \cdot g = \left( \begin{smallmatrix} a_1 & b_1 \\ c_1 & d_1 \end{smallmatrix} \right)$ (where $p_1 = \pm 1$) and the even line $l_1 = \left( \begin{smallmatrix} b_1 \\ d_1 \end{smallmatrix} \right)$ has the property $d(l_1, P_\Phi) < d(l_0, P_\Phi)$.

After a finite number of steps we get an even line $l_k$ such that $d(l_k, P_\Phi) = 0$ and then we can determine if the corresponding element $g_{i_k}^{p_k} \cdot \ldots \cdot g_{i_1}^{p_1} \cdot g$ is in $\Phi$ or not.

Now we describe how we choose the matrices $g_{i_t}^{p_t}$, $t = 1 \ldots, k$.

Suppose that $d(l_0, P_\Phi) > 0$ and $l_0 = \left( \begin{smallmatrix} b_0 \\ d_0 \end{smallmatrix} \right)$ is such that $l_0 \cap P_\Phi = \emptyset$.

We can assume that

$$\frac{b_0}{d_0} < \frac{a_0}{c_0}.$$

Since $l_0 \cap P_\Phi = \emptyset$ we must have

$$x_i \leq \frac{b_0}{d_0} < \frac{a_0}{c_0} \leq x_{i+1}$$

for some consecutive terms $x_i$ and $x_{i+1}$ of the generalized Farey sequence of $P_\Phi$.

Depending on the Farey label of the side $(x_i, x_{i+1})$ we have three cases:

**Case 1.** $(x_i, x_{i+1})$ is a free side paired to the free side $(x_j, x_{j+1})$:
In this case let $g_i$ be the side-pairing matrix that sends the free side $(x_i, x_{i+1})$ to its paired side $(x_j, x_{j+1})$. That is, $g_i = A(x_i, x_j)$ where $A(x_i, x_j)$ is given by (1.6). Then $g \in \Phi \iff g_i \cdot g \in \Phi$. So, we can consider the matrix

$$g_i \cdot g = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix}$$

and the even line

$$l_1 = \left( \begin{pmatrix} b_1 \\ a_1 \\ d_1 \\ c_1 \end{pmatrix} \right), \text{ where } l_1 = g_i \cdot l_0.$$

To get the shortest path from $l_0$ to $P_\Phi$ along $\mathbb{T}$ one has to go through $g_i^{-1}P_\Phi$. Hence, $d(l_0, P_\Phi) > d(l_0, g_i^{-1}P_\Phi)$. But $d(l_0, g_i^{-1}P_\Phi) = d(g_i \cdot l_0, g_i \cdot g_i^{-1}P_\Phi) = d(l_1, P_\Phi)$. Thus, $d(l_0, P_\Phi) > d(l_1, P_\Phi)$. We have got an even line with a shorter distance to $P_\Phi$. We can proceed inductively using $g_i \cdot g$ instead of $g$ and $l_1$ instead of $l_0$.

**Case 2.** $(x_i, x_{i+1})$ is an even side

$$x_i \xrightarrow{o} x_{i+1},$$

so $(x_i, x_{i+1})$ consists of two even edges which are paired. Let $g_i$ be the corresponding side-pairing transformation, $g_i = A_2(x_i)$, where $A_2(x_i)$ is given by (1.5). Then $g \in \Phi \iff g_i \cdot g \in \Phi$. As in Case 1

$$g_i \cdot g = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix}$$

has the property that

$$l_1 = \left( \begin{pmatrix} b_1 \\ a_1 \\ d_1 \\ c_1 \end{pmatrix} \right)$$

satisfies the inequality $d(l_0, P_\Phi) > d(l_1, P_\Phi)$. Then we can proceed inductively.

**Case 3.** $(x_i, x_{i+1})$ is an odd side

$$x_i \xrightarrow{\cdot} x_{i+1},$$

$x_i = \frac{a_i}{b_i}$ and $x_{i+1} = \frac{a_{i+1}}{b_{i+1}}$ are the endpoints of two odd sides which are paired.

Let $y = \frac{a_i + a_{i+1}}{b_i + b_{i+1}}$ and let $z$ be the common end-point of the two paired odd sides.
Let $g_i = A_3(x_i)$ be the side-pairing transformation taking $(x_i, z)$ to $(z, x_{i+1})$. We have the even lines $(x_i, y)$, $(y, x_{i+1})$ and $\left(\frac{b_0}{d_0}, \frac{a_0}{c_0}\right)$. Then either

$$x_i \leq \frac{b_0}{d_0} < \frac{a_0}{c_0} \leq y \text{ or } y \leq \frac{b_0}{d_0} < \frac{a_0}{c_0} \leq x_{i+1}.$$  

In the former case

$$g_i \cdot g = \left(\frac{a_1}{c_1}, \frac{b_1}{d_1}\right)$$

satisfies $d(l_0, P_\Phi) > d(l_1, P_\Phi)$, while in the latter case

$$g_{i-1}^{-1} \cdot g = \left(\frac{a_1}{c_1}, \frac{b_1}{d_1}\right)$$

satisfies $d(l_0, P_\Phi) > d(l_1, P_\Phi)$. Here $l_1 = (\frac{b_1}{d_1}, \frac{a_1}{c_1})$. We can then proceed inductively.

Thus, there is a finite number of generators $g_{i_t}, t = 1, 2, \ldots, k$ of $\Phi$ such that

$$g_{p_k} \cdot \ldots \cdot g_{p_1} \cdot g = \left(\frac{a_k}{c_k}, \frac{b_k}{d_k}\right), \ (p_t = \pm 1)$$

has the property $d(l_k, P_\Phi) = 0$, where $l_k = \left(\frac{b_k}{d_k}, \frac{a_k}{c_k}\right)$. That means $l_k \subseteq P_\Phi$.

The matrix $A_k = \left(\frac{a_k}{c_k}, \frac{b_k}{d_k}\right)$ maps the even line $(0, \infty)$ to the even line $l_k = \left(\frac{b_k}{d_k}, \frac{a_k}{c_k}\right)$ and both of these even lines are contained in $P_\Phi$. Then, $\left(\frac{a_k}{c_k}, \frac{b_k}{d_k}\right)$ is either identity or one of the side-pairing transformations $g_i^{\pm 1}$ of the special polygon $P_\Phi$. The following proposition describes the situation.

**Proposition 3.1.4.** [LLT95] Let $A_k = \left(\frac{a_k}{c_k}, \frac{b_k}{d_k}\right) \in \Gamma$, and $\Phi \subseteq \Gamma$ with special polygon $P_\Phi$ such that the even line $l_k = \left(\frac{b_k}{d_k}, \frac{a_k}{c_k}\right) \subseteq P_\Phi$. Then $A_k \in \Phi$ if and only if exactly one of the following holds:

1) $\left(\frac{a_k}{c_k}, \frac{b_k}{d_k}\right) = \pm I$,
2) $\left(\frac{b_k}{d_k}, \frac{a_k}{c_k}\right)$ is a free side of $P_\Phi$ paired to $(0, \infty)$,
3) $\left(\frac{a_k}{c_k}, \frac{b_k}{d_k}\right) = \pm \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ and $(0, \infty)$ consists of two even edges which are paired.

We illustrate the algorithm by an example.

**Example 3.1.5.** Let us test if the matrices $g = \left(\begin{smallmatrix} 1 & 5 \\ 2 & 11 \end{smallmatrix}\right)$ and $g' = \left(\begin{smallmatrix} 1 & -1 \\ 9 & -8 \end{smallmatrix}\right)$ belong to the subgroup $\Phi = \Gamma_0(9)$ with the following special polygon:
FIGURE 3.3. The fundamental domain of $\Gamma_0(9)$.

The generators corresponding to the free sides are:

$$g_1 = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$$

sending the edge $(0, \infty)$ to $(1, \infty)$,

$$g_2 = \begin{pmatrix} -4 & 1 \\ -9 & 2 \end{pmatrix}$$

sending the edge $(0, \frac{1}{3})$ to the edge $(\frac{1}{3}, \frac{1}{2})$ and

$$g_3 = \begin{pmatrix} -7 & 4 \\ -9 & 5 \end{pmatrix}$$

sending the edge $(\frac{1}{3}, \frac{2}{3})$ to the edge $(\frac{2}{3}, 1)$.

For $g = \begin{pmatrix} 1 & 5 \\ 2 & 11 \end{pmatrix}$, we have $g \cdot (0, \infty) = (\frac{5}{11}, \frac{1}{2})$ and since the edge $l_1 = (\frac{5}{11}, \frac{1}{2})$ lies under the edge $(\frac{1}{3}, \frac{1}{2})$, we have to multiply $g$ to the left by $g_2^{-1}$. Because the generator that sends the edge $(\frac{1}{3}, \frac{1}{2})$ to its corresponding edge in the special polygon of $\Gamma_0(9)$ is $g_2^{-1}$.

We get

$$g_2^{-1} \cdot g = \begin{pmatrix} 2 & -1 \\ 3 & -4 \end{pmatrix} \cdot \begin{pmatrix} 1 & 5 \\ 2 & 11 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix}.$$

If we apply this matrix to the edge $(0, \infty)$, we get the even line $l_2 = (-1, 0) = (-1, 0)$. 

45
Since this even line lies under the edge $(-\infty, 0)$ we apply $g_1$.

We get the matrix

$$g_1 \cdot g_2^{-1} \cdot g = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & -1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}. $$

When we apply this matrix to the line $(0, \infty)$, we get the even line $l_3 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = (0, 1)$. This even line is not a side of the polygon paired to $(0, \infty)$. It just overlaps the edges inside the polygon. We conclude that $g$ does not belong to $\Phi$.

\[ \text{FIGURE 3.4. The images of the even lines.} \]

For $g' = \begin{pmatrix} \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{5} & -\frac{8}{5} \end{pmatrix}$, we have $l_1' = g' \cdot (0, \infty) = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} \\ \frac{1}{5} & \frac{8}{5} \end{pmatrix}$. Since this edge lies under $(0, \frac{1}{3})$ we multiply $g'$ by $g_2$. We get

$$g_2 \cdot g' = \begin{pmatrix} -\frac{4}{9} & \frac{1}{3} \\ -\frac{9}{5} & -\frac{8}{5} \end{pmatrix} \begin{pmatrix} \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{5} & -\frac{8}{5} \end{pmatrix} = \begin{pmatrix} \frac{5}{9} & -\frac{4}{7} \\ \frac{5}{9} & -\frac{4}{7} \end{pmatrix}$$

The even line corresponding to this matrix is $l_2' = \begin{pmatrix} \frac{5}{9} & \frac{4}{7} \\ \frac{5}{9} & \frac{4}{7} \end{pmatrix}$. $l_2'$ is under the edge $(\frac{1}{2}, \frac{2}{3})$.

Thus, we need to multiply by $g_3$.

We get

$$g_3 \cdot g_2 \cdot g' = \begin{pmatrix} -\frac{7}{9} & \frac{4}{5} \\ -\frac{9}{5} & \frac{4}{5} \end{pmatrix} \begin{pmatrix} \frac{5}{9} & -\frac{4}{7} \\ \frac{5}{9} & -\frac{4}{7} \end{pmatrix} = \begin{pmatrix} \frac{1}{1} & 0 \\ 0 & 1 \end{pmatrix} = I. $$

The corresponding even line is $l_3' = (0, \infty)$.

We conclude that $g' \in \Phi$ and from $g_3 \cdot g_2 \cdot g' = I$ we get

$$g' = g_2^{-1} \cdot g_3^{-1}. $$
3.2 A New Algorithm

The Lang, Lim and Tan’s algorithm uses the line \((0, \infty)\) and its images under different matrices to test the membership of a matrix \(g\) to a subgroup \(\Phi \subseteq \Gamma\). A variant of this algorithm would be to use the entire fundamental domain \(P_\Phi\).

Example 3.2.1. Consider again the example with \(\Phi = \Gamma_0(9)\) and \(g = (\frac{1}{2} \frac{5}{11})\). Then the even line \(l_3\) is the image of \((0, \infty)\) under the matrix \(m = g_1 \cdot g_2^{-1} \cdot g = (\frac{1}{1} \frac{0}{1})\).

So, just consider \(m = (\frac{1}{1} \frac{0}{1})\) and the image \(m \cdot P_\Phi\) of \(P_\Phi\) under \(m\). The domain \(m \cdot P_\Phi\) has vertices \(\{m \cdot \infty, m \cdot 0, m \cdot \frac{1}{3}, m \cdot \frac{1}{2}, m \cdot \frac{2}{3}, m \cdot 1\} = \{0, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{1}{1}\}\).

The images of \(P_\Phi\), \(m \cdot P_\Phi\) and \(P_\Phi\) together with \(m \cdot P_\Phi\) are shown in figure 3.5-3.7.

![FIGURE 3.5. The fundamental domain \(P_\Phi\) of \(\Phi\).]

![FIGURE 3.6. The image \(m \cdot P_\Phi\) of \(P_\Phi\) under \(m\).]
FIGURE 3.7. Overlapping domains

The new algorithm is based on the way a fundamental domain $F$ and its $\Gamma$-translates are tiling the plane. We prove some facts about the tiling of $\mathbb{H}$ by fundamental domains.

**Proposition 3.2.2.** Let $F$ be a special polygon with side-pairing matrices $G_l = \{g_1, g_2, \ldots, g_n\}$ and let $m_1$ and $m_2$ be elements in $\Gamma$.

a) If $m_1F = m_2F$, then $m_1 = m_2$.

b) $m_1F$ is adjacent to $m_2F$ if and only if $m_1 = m_2 \cdot g_i$, for some $1 \leq i \leq n$.

**Proof.** a) Suppose $m_1F = m_2F$. Suppose $y \in m_1F$, and assume that $y$ is in the interior of $m_1F$. Since $y \in m_1F$, we have $y = m_1z_1$ for some $z_1 \in F$. Similarly, $y = m_2z_2$ for some $z_2 \in F$. So $m_1z_1 = m_2z_2$ or $z_1 = m_1^{-1}m_2z_2$. Since there is a unique point $z$ in $F$ with $z_1 = gz$ for some $g \in \Gamma$, and we have $z_1 = Iz_1$ we must have $z_1 = z_2$, so $m_1^{-1}m_2$ must fix $z_1$. Points strictly in the upper half plane are fixed by non trivial matrices in $SL_2(\mathbb{Z})$ only if they are elliptic of order 2 or 3, but these are isolated points and since $F$ is open, we can choose $y$ to be some point in $m_1F$ which is not elliptic, and so $z_1$ is not elliptic. Then $z_1$ is fixed only by the identity, so $m_1^{-1}m_2 = I$, i.e., $m_1 = m_2$.

b) When are $m_1F$ and $m_2F$ adjacent? By definition, $F$ is a fundamental domain, so either $m_1F = m_2F$, or these regions have no intersection (so the images of the domain tile $\mathbb{H}$ without overlapping). Suppose $m_1F$ and $m_2F$ are adjacent. This means that there is a point $x$ in the intersection of the closure of these regions;
this means that $x$ must be on the boundary of $m_1 F$ and $m_2 F$. Apply the map $m_1^{-1}$ to these domains. This is a continuous map on $\mathbf{H}$, so $F = m_1^{-1}m_1 F$ and $m_1^{-1}m_2 F$ also meet in a boundary point, $y = m_1^{-1}x$.

Now, $F$ is a fundamental domain, so that means that some bounding edges should be included, and some not. For instance, suppose $g_1$ maps edge $e_1$ to edge $e_2$, then we can include edge $e_1$ in the domain, in which case we should not include edge $e_2$ (this will be an open edge of the domain, whereas edge $e_1$ will be “closed”).

We can’t include both edges, since we can only include a unique point equivalent to a point under the action of $\Gamma$.

Suppose $y$ is a point on an edge $e_2$ that is not included in the domain $F$ (but is a boundary edge). Then for some $g_i$ and some edge $e_1$, we have $g_i$ mapping a point $w$ of the edge $e_1$ to $y$. That is, $g_i w = y$. On the other hand, $y$ is also on an edge of $m_1^{-1}m_2 F$. Since $y$ must be contained in some image of $F$, if it’s not contained in $F$, it must be contained in $m_1^{-1}m_2 F$, since the space between these domains along the edge where they meet can’t be contained in a third domain.

So $y = m_1^{-1}m_2 p$ for some point in $F$. Now we have $y = g_i w = m_1^{-1}m_2 p$, so by a uniqueness argument as before, and making sure not to take $y$ an elliptic point, we have $w = p$ and $m_1^{-1}m_2 = g_i$, so $m_2 = m_2 g_i$.

Now suppose that two domains have the form $m F$ and $mg F$ with $g$ a side-pairing matrix. If edge $e_1$ is joined to $e_2$ by $g$, then the region $F$ has an adjoining region $g F$, and the place where these two regions join is the edge $e_2$. The regions $F$ and $g^{-1} F$ meet each other along the edge $e_1$ (see figure 3.8).

Now take some matrix $m$ and apply to these regions. Since $F$ and $g F$ join along $e_2$, then $m F$ and $mg F$ join along $me_2$ (see figure 3.9).

We should notice that any region of the form $m F$ with $F$ a fundamental domain and $m$ a matrix in $PSL_2(\mathbb{Z})$ is surrounded by regions $mg_i F$ and $mg_i^{-1} F$, where
FIGURE 3.8. A side-pairing matrix and the corresponding domains.

FIGURE 3.9. The image of the domains under a matrix.

$g_i$ runs through the generators corresponding to the Farey symbol. All of these regions lie under $mF$, except one of them, which borders the largest edge of $mF$. Let $g$ be the matrix corresponding to this exceptional case, so $mgF$ lies over $mF$.

The matrix $g$ is the only side-pairing matrix that enlarges the domain $mF$. If $g$ sends the edge $e_1$ to $e_2$ then $gF$ lies under the edge $e_2$, $F$ and $gF$ having the edge $e_2$ in common (see figure 3.8). Then $mF$ and $mgF$ will have the edge $me_2$ in common, so they are joined along $me_2$.

The new algorithm is based on the existence of such a side-pairing matrix $g_{enl}$, which at each step widens the domain $mF$. Given a matrix $m$ and a subgroup $\Phi \subseteq \Gamma$ with the special polygon $F$ we start by finding $mF$. Then we find the largest edge of $mF$, let us say it is $me_2$. Let $e_2$ be the preimage of $me_2$ in $F$. Then $g_{enl} = g_1$ is the side-pairing matrix that has as destination $e_2$, that is $g_{enl}$ is
FIGURE 3.10. The domains $mF$ and $mgF$.

the matrix that pairs some edge $e_1$ to the edge $e_2$. Then we proceed inductively with $mg_{en}F$ instead of $mF$ and $mg_1$ instead of $m$. At each step we widen the domain till we either overlap with $F$ or we get exactly $F$. If at some step we have overlapping, we conclude that $m$ does not belong to $\Phi$. If we get exactly $F$, then $m \cdot g_1 \cdot g_2 \cdot \ldots \cdot g_k F = F$ for some side-pairing matrices $g_1, g_2, \ldots, g_k$. From the above proposition we have $m \cdot g_1 \cdot g_2 \cdot \ldots \cdot g_k = I$, so $m = g_k^{-1} \cdot \ldots \cdot g_2^{-1} \cdot g_1^{-1}$, which gives the decomposition of $m$ in terms of the generators of $\Phi$.

Let us illustrate the algorithm with an example

**Example 3.2.3.** Let us consider the subgroup $\Phi = \Gamma_0(6)$ with the special polygon $F$ as in the figure 3.11 and let $m = (\begin{array}{cc} 1 & 0 \\ -12 & 1 \end{array})$.

![Fundamental domain of $\Gamma_0(6)$](image)

Each edge $e$ is joined to another edge by some matrix $g_e$. We write these matrices next to each edge. Note that if $e_i$ is joined to $e_j$ by $g_e$, then $e_j$ is joined to $e_i$ by
$g_e^{-1}$. Corresponding to this picture, we have the Farey symbol

$$FS = \{-\infty, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{2}{1}, 1, \infty\}$$

and the list of side-pairing matrices:

$$\text{Glue}(FS) = \{(1 \ 1), (-\frac{5}{6} \ 1), (-\frac{7}{12} \ 3), (\frac{5}{12} \ -\frac{3}{7}), (\frac{1}{6} \ -\frac{1}{5}), (1 \ 0)\}$$

Now we use the algorithm to see if $m$ is in $\Phi$. The image of $mF$ is as follows.

![Figure 3.12. The image of $mF$.](image)

![Figure 3.13. $F$ and $mF$ together](image)

We see that the largest edge of $mF$ is $(-\frac{1}{9}, 0)$. We find the preimage of $(-\frac{1}{9}, 0)$ in $F$. Because $m(0) = 0$ and $m(\frac{1}{3}) = -\frac{1}{9}$, the preimage is the edge $(0, \frac{1}{3})$. Now we find what side-pairing matrix has as destination the edge $(0, \frac{1}{3})$. We see that $g_1 = (\frac{1}{6} \ -\frac{5}{6})$ sends $(\frac{2}{3}, 1)$ to $(0, \frac{1}{3})$. We continue with the matrix $mg_1 = (\frac{1}{6} \ -\frac{1}{7})$ and the domain $mg_1F$. 

52
The domain $mg_1F$ has vertices

$$\left\{ mg_1 \cdot \infty, mg_1 \cdot 0, mg_1 \cdot \frac{1}{3}, mg_1 \cdot \frac{1}{2}, mg_1 \cdot \frac{2}{3}, mg_1 \cdot 1 \right\}$$

or

$$\left\{ -\frac{1}{6}, -\frac{7}{1}, -\frac{2}{15}, -\frac{1}{8}, -\frac{9}{1}, 0 \right\}.$$

$F$ and $mg_1F$ are shown below:

![Figure 3.14. F and mg1F](image)

The largest edge of $mg_1F$ is $(-\frac{1}{6}, 0)$ and it has as preimage the edge $(\infty, 1)$.

$g_2 = (\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix})$ has as destination $(\infty, 1)$. Then we apply $g_2$ to $mg_1F$ to get the domain $mg_1g_2F$. The new matrix is $mg_1 \cdot g_2 = (\begin{smallmatrix} 1 & 0 \\ -6 & 1 \end{smallmatrix})$. The domains $F$ and $mg_1g_2F$ are shown below:

![Figure 3.15. F and mg1g2F](image)

The largest edge of $mg_1g_2F$ is $(-\frac{1}{3}, 0)$ and it has as preimage the edge $(0, \frac{1}{3})$.

$g_3 = (\begin{smallmatrix} 1 & -1 \\ 6 & 5 \end{smallmatrix})$ has as destination $(0, \frac{1}{3})$. Then we apply $g_3$ to $mg_1g_2F$ to get the
domain $mg_1g_2g_3F$. The new matrix is $mg_1g_2g_3 = (\frac{1}{0} - \frac{1}{1})$. The domains $F$ and $mg_1g_2g_3F$ are shown:

![Diagram](image)

FIGURE 3.16. $F$ and $mg_1g_2g_3F$.

Now the largest edge of $mg_1g_2g_3F$ is $(0, \infty)$ which has as preimage the edge $(1, \infty)$. So we need to multiply by the matrix $g_4 = (\frac{1}{0} \frac{1}{1})$. Then we get $mg_1g_2g_3g_4F = F$. We conclude that $m \in \Phi$ and the decomposition is

$$m = g_4^{-1}g_3^{-1}g_2^{-1}g_1^{-1} = \left(\frac{-5}{6} \frac{1}{1}\right) \left(\frac{1}{0} - \frac{1}{1}\right) \left(\frac{-5}{6} \frac{1}{1}\right) \left(\frac{1}{0} - \frac{1}{1}\right).$$

All we need to prove about the algorithm is that it ends after a finite number of steps.

Let $m$ be the initial matrix and let $F$ be the fundamental domain. If $mF$ has an infinite edge, then it is clear that the algorithm stops. Just check if $F$ and $mF$ overlap or if $mF$ can be brought over $F$ by some side-pairing matrix of the form $(\frac{1}{n} \frac{n}{1})$. In the former case $m$ does not belong to the subgroup, in the latter case $m$ is of the form $(\frac{1}{n} \frac{n}{1})$. So, we can assume that the largest edge of $mF$, let us say $(a_1, b_1)$ has finite length. It is enough to prove that there are only finitely many lengths of the form $|g(a_1) - g(b_1)|$ such that $|g(a_1) - g(b_1)| > |a_1 - b_1|$, where $g = (\frac{a}{c} \frac{b}{d}) \in \Gamma$.

But

$$g(a_1) - g(b_1) = \frac{aa_1 + b}{ca_1 + d} - \frac{ab_1 + b}{cb_1 + d} = \frac{(ad - bc)(a_1 - b_1)}{(ca_1 + d)(cb_1 + d)} = \frac{a_1 - b_1}{(ca_1 + d)(cb_1 + d)}.$$ 

So, $|g(a_1) - g(b_1)| > |a_1 - b_1|$ means

$$\frac{1}{|ca_1 + d||cb_1 + d|} > 1$$

(3.1)

With $a_1$ and $b_1$ fixed, it is enough to show that there are only finitely many $c's$ and only finitely many $d's$ such that (3.1) holds. Since there are only finitely many
lattice points of the form $cb_1 + d$ inside the unit circle, we infer that

$$M_{b_1} = \max_{c,d \in \mathbb{Z}} \left\{ \frac{1}{|cb_1 + d|}, 1 \right\}$$

is a finite number. Then, there are only finitely many lattice points of the form $ca_1 + d$ with the property $|ca_1 + d| < M_{b_1}$. That means $|ca_1 + d||cb_1 + d| < 1$ only for finitely many $c's$ and $d's$, which is exactly what we needed.

### 3.3 Small-Index Noncongruence Subgroups

Even if a lot of effort have been put into studying both the congruence and the noncongruence subgroups, little is known about them. In [Den75], Dennin showed that there are only finitely many congruence subgroups of a given genus. In [Seb01], Sebbar classified all torsion-free (elliptic elements free) genus zero congruence subgroups. There are only 33 such groups. Chua, Lang, and Yang gave explicit descriptions for all the genus zero congruence subgroups [CLY04]. Cummins and Pauli [CP03] classified all congruence subgroups with genus less than 25.

On the other hand, Newman showed that there are infinitely many genus zero noncongruence subgroups [New65]. In [ASD71], Atkin and Swinnerton-Dyer showed that noncongruence subgroups predominate congruence subgroups. Jones proved that there are infinitely many noncongruence subgroups of a given genus. Let $N_c(d)$ (resp. $N_{nc}(d)$) be the number of congruence (resp. noncongruence) subgroups of index $d$ in $PSL_2(\mathbb{Z})$. Stothers gave an upper bound for $N_c(d)$ and showed that $N_c(d)/N_{nc}(d) \to 0$ when $n \to \infty$ [Sto84].

We will find the noncongruence subgroups of $PSL_2(\mathbb{Z})$ of index less than 10. Since congruence is invariant under conjugation, it’s enough to work with representatives of conjugacy classes. According to Vidal’s classification principle, to any conjugacy class corresponds a trivalent diagram. We mark the diagram and find the corresponding Farey symbol. From the Farey symbol we calculate an independent set of generators and the geometric level $N$ of the corresponding subgroup. Then using Lang, Lim and Tan’s algorithm we see if $\Gamma(N)$ is included in the group
or not. We just take any generator of $\Gamma(N)$ and test its membership to the group.

The table below describes the noncongruence subgroups of index less than 10. The table indicates the following characteristics of a subgroup: the index, the genus, the geometric level, the list of cusp widths with the first one being the width of $\infty$, the cusp vertices without $-\infty$ and $\infty$ and the Farey labels. A Farey label equal to $-3$ corresponds to an odd interval, while a $-2$ label corresponds to an even interval. A positive label is associated to a free interval.

Notice that all subgroups of index less than or equal to 6 are congruence, which confirms a claim of Wohlfahrt [Woh64]. The appendix provides the Magma code for testing the congruence of a subgroup given by a Farey symbol.

TABLE 3.1. Noncongruence Subgroups of Index Less Than 10

<table>
<thead>
<tr>
<th>index</th>
<th>genus</th>
<th>level</th>
<th>cusp widths</th>
<th>vertices</th>
<th>Farey labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>0</td>
<td>12</td>
<td>4, 3</td>
<td>0, 1/2, 1</td>
<td>-3, 1, -2, 1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>10</td>
<td>5, 2</td>
<td>0, 1/2, 1</td>
<td>-3, -2, 1, 1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>6</td>
<td>6, 1</td>
<td>0, 1/2, 1</td>
<td>-2, 1, 1, -3</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>6</td>
<td>6, 1</td>
<td>0, 1, 2</td>
<td>-2, -3, 1, 1</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0, 1/2, 1</td>
<td>-2, -2, -3, -3</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>8</td>
<td>8</td>
<td>0, 1/2, 1</td>
<td>-2, -3, -3, -2</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>0, 1/2, 1</td>
<td>-2, -3, -3, -3</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>9</td>
<td>9</td>
<td>0, 1, 2</td>
<td>-2, -3, -3, -3</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>9</td>
<td>9</td>
<td>0, 1/2, 2/3, 1</td>
<td>-2, 1, 2, 1, 2</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>14</td>
<td>7, 2</td>
<td>0, 1, 2, 3</td>
<td>-2, 1, 1, -2, -2</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>6</td>
<td>3, 6</td>
<td>0, 1/3, 1/2, 1</td>
<td>1, -2, -2, 1, -2</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>6</td>
<td>6, 2, 1</td>
<td>0, 1, 2, 3</td>
<td>-2, 1, 1, 2, 2</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>15</td>
<td>3, 5, 1</td>
<td>0, 1/3, 1/2, 1</td>
<td>1, 2, 2, 1, -2</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>8</td>
<td>8, 1</td>
<td>0, 1, 2, 3</td>
<td>-2, -2, 1, 1, 1</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>7</td>
<td>7, 1, 1</td>
<td>0, 1, 2, 3</td>
<td>1, 1, -2, 2, 2</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>8</td>
<td>8, 1</td>
<td>0, 1, 2, 3</td>
<td>1, 1, -2, -2, -2</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>12</td>
<td>4, 2, 3</td>
<td>0, 1/2, 2/3, 1</td>
<td>-2, 1, 1, 2, 2</td>
</tr>
</tbody>
</table>

In [Vid06], Vidal calculated the generating series for the number of trivalent diagrams and the one for the marked trivalent diagrams. For the trivalent diagrams the generating series also indicates the number of conjugacy classes of subgroups of $\Gamma$ of a certain index. The coefficients of the series indicate the number of trivalent
diagrams of a certain size.

\[ \overline{D}_3(t) = t + t^2 + 2t^4 + t^5 + 8t^6 + 6t^7 + 7t^8 + 14t^9 \]

\[ + 27t^{10} + 26t^{11} + 80t^{12} + 133t^{13} + 170t^{14} + 348t^{15} + 765t^{16} + 1002t^{17} + \ldots \]

Thus, 4 out of 6 trivalent diagrams of size 7 give classes of noncongruence subgroups of index 7, 2 out of 7 trivalent diagrams of size 8 give classes of noncongruence subgroups of index 8 and 11 out of 14 trivalent diagrams of size 9 correspond to classes of noncongruence subgroups of index 9.

3.4 Conclusions

The noncongruence subgroups of $PSL_2(\mathbb{Z})$ play an important role in many branches of mathematics. Even if several authors tried to classify them according to their geometric characteristics like genus or the list of cusp widths, only a partial description of these groups have been achieved. In this work, I tried to classify these groups based on their index. The algorithms I presented can determine all noncongruence subgroups of a certain index. Such a method will run into computational problems at high indices, because of the big number of noncongruence subgroups and the time required to calculate the generators of $\Gamma(N)$ for big values of $N$.

For a big index the method is more suitable to testing if a subgroup given by a Farey symbol or a marked trivalent diagram is congruence or not. The method is also applicable to subgroups given by a system of generators, but involves more computation since one will have to work with all marked trivalent diagrams of a given size.
References


[Fri86] R. Fricke, Über die Substitutionsgruppen, welche zu den aus den Legendre’schen Integralmodul $k2(w)$ gezogenen Wurzeln gehören. (Mit einer Figurentafel), Math. Ann. 28(1886), 99-118


[New65] M. Newman, Normal subgroups of the modular group which are not congruence subgroups, Proc. Amer. Math. Soc. 16(1965), 831-832


Appendix : Magma Programs

The first program is used for generating trivalent diagrams. To generate all trivalent diagrams of size 1 to \(m\) just use the function gentrivdiag(m). The second program is used to determine the Farey symbol associated to a marked trivalent diagram. Just use travdiag(G, cyclesG, VsatG, ein, d), where G is the diagram, cyclesG is a list of its cycles and VsatG is a list of white vertices of degree 2. The third program is used for testing the congruence of a subgroup given by a Farey symbol. Just use the function resneed(V, FS, n), where V is the set of vertices without \(-\infty\) and \(\infty\), FS is the sequence of Farey labels, \(-2\) means an even interval, \(-3\) means an odd interval, a positive label corresponds to a free interval.

The first program generates all trivalent diagrams of a given size:

```magma
function bfirst1(G, d, v);
//adds an edge to the edge with terminal vertex(of index) v,
//of the graph G of size d, v=white vertex of degree 1
Gco:=G;
V:=Vertices(Gco);
E:=Edges(Gco);
n:=#V;
AddVertex(~Gco, "b");
Gco+=[n+1, v];
return Gco;
end function;

function bfirst3(G, cyclesG, VsatG, d, v);
// adds 3 edges to G in all possible ways, v=white vertex
//of degree 1
V:=Vertices(G);
n:=#V;
V1:=[x: x in [1..n]|(InDegree(V!x) eq 1) and (x ne v) and (x notin VsatG)];

// 1-point connection

Gc:=G;
Vc:=Vertices(Gc);
Ec:=Edges(Gc);
AddVertices(~Gc, 3, ["b", "w", "w"]);```
AddEdges(~Gc, [[n+1, v], [n+1, n+2], [n+1, n+3]]);
cyclesGc:=cyclesG cat [[n+1, v, n+2, n+3]];
cyclesGc2:=cyclesG cat [[n+1, v, n+2, n+3, n+1]];
Vsat2:=VsatG cat [n+2, n+3];
r1:=[Gc, Gc];
r2:=[cyclesGc, cyclesGc2];
r3:=[VsatG, Vsat2];

// 2-point connection

if #V1 ge 1 then
    for i in V1 do
        Gco1:=G; Gco2:=G;
        Vco1:=Vertices(Gco1); Eco1:=Edges(Gco1);
        Vco2:=Vertices(Gco2); Eco1:=Edges(Gco2);
        AddVertices(~Gco1, 2, ["b", "w"]);
        AddEdges(~Gco1, [[n+1, i], [n+1, v], [n+1, n+2]]);
        cyclesGco1:=cyclesG cat [[n+1, i, v, n+2]];
        AddVertices(~Gco2, 2, ["b", "w"]);
        AddEdges(~Gco2, [[n+1, i], [n+1, v], [n+1, n+2]]);
        cyclesGco2:=cyclesG cat [[n+1, i, n+2, v]];
        r1 cat:=[Gco1, Gco2];
r2 cat:=[cyclesGco1, cyclesGco2];
r3 cat:=[VsatG, Vsat2];
    end for;
end if;

// 3-point connection

if #V1 ge 2 then
    for t:=1 to (#V1-1) do
        a:=V1[t];
        for u:=t+1 to #V1 do
            b:=V1[u];
            Gcop1:=G;
            Gcop2:=G;
            Vcop1:=Vertices(Gcop1); Ecop1:=Edges(Gcop1);
            Vcop2:=Vertices(Gcop2); Ecop1:=Edges(Gcop2);
            AddVertex(~Gcop1, ["b"]);
            AddEdges(~Gcop1, [[n+1, a], [n+1, b], [n+1, v]]);
            cyclesGcop1:=cyclesG cat [[n+1, a, b, v]];
            AddVertex(~Gcop2, ["b"]);
            AddEdges(~Gcop2, [[n+1, a], [n+1, v], [n+1, b]]);
        end for;
    end for;
end if;
cyclesGcop2 := cyclesG cat [[n+1, a, v, b]];

r1 cat := [Gcop1, Gcop2];
r2 cat := [cyclesGcop1, cyclesGcop2];
r3 cat := [VsatG, VsatG];
end for;
end for;
end if;

return r1, r2, r3;
end function;

function equivtrivdiagr(G1, cyclesG1, G2, cyclesG2);
// determines if two trivalent diagrams G1 and G2 are
// equivalent (isomorphic as colored graphs)
a, f := IsIsomorphic(G1, G2);
if a eq true then
    Vs1 := Vertices(G1);
    Vs2 := Vertices(G2);
    B := [x: x in [1..#Vs1]|Degree(Vs1!x) eq 3];
    for bl in B do
        ci1 := [cyclesG1[t]: t in [1..#cyclesG1]|cyclesG1[t][1] eq bl];
        ci2 := [cyclesG2[u]: u in [1..#cyclesG2]|cyclesG2[u][1] eq f(bl)];
        c1 := ci1[1];  c2 := ci2[1];
        if #c1 ne #c2 then
            return false;
        else
            i := 1;
            while (i le #c1) do
                if f(Vs1!c1[i]) ne Vs2!c2[i] then
                    return false;
                else
                    i := i + 1;
                end if;
            end while;
        end if;
    end for;
    return true;
else
    return false;
end if;
end function;
function gentrivdiag(m);

//generate all trivalent diagrams of size m
//all graphs will be stored in listofgr, their cycles in
//listofcycles, and the white bivalent vertices in
//listofsatvert initialize these lists with the graphs
//of size 1, 2 and 3

G1:=Digraph<2|[1,2]>;
AssignLabels(G1, [1,2], ["b", "w"]);
G2:=Digraph<3|[1,2], [3, 2]>;
AssignLabels(G2, [1,2,3], ["b", "w", "b"]);
G31:=Digraph<4|[2, 1], [2, 3], [2,4]>;
AssignLabels(G31, [1,2,3,4], ["w", "b", "w", "w"]);
listofgr:= [[G1], [G2], [G31, G31]];
listofcycles:= [[], [], [[[2, 1, 3, 4]], [[2, 1, 3, 4, 2]]]];
listofsatvert:= [[], [], [[], [3, 4]]];

if m gt 3 then
    for j:=4 to m do
        //generate the graphs of size j
        agr:=[];
        acyc:=[];
        asat:=[];

        //apply bfirst1 to all graphs of size j-1
        M:=#listofgr[j-1];
        for c:=1 to M do
            G:=listofgr[j-1][c];
            cyclesG:=listofcycles[j-1][c];
            VsatG:=listofsatvert[j-1][c];
            Vs:=Vertices(G);
            V:=[x: x in [1..#Vs]|(InDegree(Vs!x) eq 1) and (x notin VsatG)];
            for v in V do
                agr:=agr cat [bfirst1(G, j-1, v)];
                acyc:=acyc cat [cyclesG];
                asat:=asat cat [VsatG];
            end for;
        end for;

        //apply bfirst3 to all graphs of size j-3
        N:=#listofgr[j-3];
        for r:=1 to N do
            G:=listofgr[j-3][r];

        end for;
    end for;
end if;
cyclesG:=listofcycles[j-3][r];
VsatG:=listofsatvert[j-3][r];
Vs:=Vertices(G);
Vn:=[x: x in [1..#Vs]|(InDegree(Vs!x) eq 1) and (x notin VsatG)];
for nv in Vn do
  a1, a2, a3:= bfirst3(G, cyclesG, VsatG, j-3, nv);
  agr:=agr cat a1;
  acyc:=acyc cat a2;
  asat:=asat cat a3;
end for;
end for;

// Remove the equivalent diagrams

Nd:=#agr;
if Nd gt 2 then
  i:=2;
  while i le Nd do
    if equivtrivdiagr(agr[1],acyc[1],agr[i],acyc[i]) then
      Remove(~agr, i);
      Remove(~acyc, i);
      Remove(~asat, i);
    else
      i:=i+1;
    end if;
  end while;
end if;
listofgr cat:=agr;
listofcycles cat:=acyc;
listofsatvert cat:=asa;
end for;
end if;
return listofgr, listofcycles, listofsatvert;
end function;

The second program finds the Farey symbol corresponding to a marked trivalent diagram

function oncycle(G, ed);
//returns true if the edge ed belongs to a cycle of the graph G
E:=Edges(G);
Dualg:=Graph{u, v|u, v in E| u adj v}
V:=Vertices(Dualg);
i:=Index(ed);
A:=AdjacencyMatrix(Dualg);
if #E ge 4 then
  j:=4;
  while j le #E do
    B:=A^j;
    if B[i, i] ge 1 then
      return true;
    else
      j:=j+2;
    end if;
  end while;
end if;
return false;
end function;

function consum(G, cyclesG, vbl);
//determines whether all edges coming out of the
//black vertex vbl are labelled (degree vbl=3)
i:=Index([cyclesG[y][1]: y in [1..#cyclesG]], vbl);
Edc:=Edges(G);
if IsLabelled(Edc[vbl, cyclesG[i][2]]) and
  IsLabelled(Edc[vbl, cyclesG[i][3]]) and
  IsLabelled(Edc[vbl, cyclesG[i][4]]) then
  return true;
else
  return false;
end if;
end function;

function travdiag(G, cyclesG, VsatG, ein, d);
// produces the Farey symbol associated to trivalent diagram G
// of size d with ein the marked edge
// r means vertex labels 1..r
// ledge is the edge label of the current edge
// vdeg1stuff contain all information w.r.t. degree 1 vertices:
// the index and the cusps c1, c2 c1 on top(counterclockwise)
r:=0;
ledge:=0;
llab:=[Integers()|0; x in [1..#Vertices(G)]]
vdeg1stuff:=[ ];

//make copies
Gn:=G;
cyclesGn:=cyclesG;
VsatGn:=VsatG;
i:=Index(Edges(G), ein);
E:=Edges(Gn); V:=Vertices(Gn);
edin:=E!i;
edge:=edin;
while ledge le d do
v1:=Index(edge[1]);
v2:=Index(edge[2]);
v2n:=v2;
if oncycle(Gn, edge) then
    // cut cycle
    Gn-:=[v1, v2];
    AddVertex(~Gn, "w");
    v2n:=#Vertices(Gn);
    Gn+:=[v1, v2n];

    // put vertex labels
    r:=r+1;
    llab cat:=[r];
    llab[v2]:=r;

    // modify the cycle
    a:=Index([cyclesGn[t][1]: t in [1..#cyclesGn]], v1);
    cyclesGn[a][Index(cyclesGn[a], v2)]:=v2n;
    edge:=E![v1, v2n];
elseif (v2 in VsatGn) then
    b:=Index([cyclesGn[t][1]: t in [1..#cyclesGn]], v1);
    Remove(~cyclesGn[b], 5);

    // put vertex labels
    r:=r+1;
    llab[v2]:=r;
    oth:=[u: u in (Seqset(VsatGn) meet Seqset(cyclesGn[b]))|u ne v2];
    llab[oth[1]]:=r;
end if;
// label the current edge
ledge:=ledge+1;
AssignLabel(cedge, ledge);

// this is the sequence of black vertices encountered so far
lblack3:=[ ];

if Label(cedge) eq 1 then
    // put the initial cusps
    vdeg1stuff:=[[v2n, 1,0,0,1]];
    lofdeg1or2:=[v2n];
end if;

m:=Index([vdeg1stuff[q][1]: q in [1..#vdeg1stuff]], v2n);

if Degree(V!v1) eq 1 then
    vdeg1stuff cat:=[[v1, vdeg1stuff[m][2], vdeg1stuff[m][3],
                     vdeg1stuff[m][4], vdeg1stuff[m][5]];
    lofdeg1or2 cat:=[v1];
else
    lblack3:=lblack3 cat [v1];
    w:=Index([cyclesGn[t][1]: t in [1..#cyclesGn]], v1);
    z:=Index(cyclesGn[w], v2n);

    // determine the next edges
    if z eq 2 then
        n1:=cyclesGn[w][3];
        n2:=cyclesGn[w][4];
    else if z eq 3 then
        n1:=cyclesGn[w][4];
        n2:=cyclesGn[w][2];
    else
        n1:=cyclesGn[w][2];
        n2:=cyclesGn[w][3];
    end if;

    if !IsLabelled(E![v1, n1]) and !IsLabelled(E![v1, n2]) then
        if Degree(V!v2n) eq 2 then
            eda:=Setseq(IncidentEdges(V!v2n) diff {cedge});
            eadj:=eda[1];
            if IsLabelled(eadj) then
                // change the order of cusps
                p1:=vdeg1stuff[m][2];
                p2:= vdeg1stuff[m][3];
                vdeg1stuff[m][2]:= vdeg1stuff[m][4];
            end if;
        end if;
    end if;
end if;
vdeg1stuff[m][3]:=vdeg1stuff[m][5];
vdeg1stuff[m][4]:=p1;
vdeg1stuff[m][5]:=p2;
end if;
end if;

// add the new cusps
if vdeg1stuff[m][3]*vdeg1stuff[m][4] < vdeg1stuff[m][2]*vdeg1stuff[m][5] then
    r1:= vdeg1stuff[m][2]+ vdeg1stuff[m][4];
r2:= vdeg1stuff[m][3]+vdeg1stuff[m][5];
else
    if vdeg1stuff[m][5] > vdeg1stuff[m][3] then
        r1:= vdeg1stuff[m][4]- vdeg1stuff[m][2];
r2:= vdeg1stuff[m][5]-vdeg1stuff[m][3];
    else
        r1:= vdeg1stuff[m][2]- vdeg1stuff[m][4];
r2:= vdeg1stuff[m][3]-vdeg1stuff[m][5];
    end if;
end if;

vdeg1stuff cat:=[[n1, vdeg1stuff[m][4], vdeg1stuff[m][5], r1, r2]];
vdeg1stuff cat:=[[n2, r1, r2, vdeg1stuff[m][2], vdeg1stuff[m][3]]];
lofdeg1or2 cat:=[n1, n2];
end if;  // !IsLab(v1, n1) !IsLab(v1, n2)
end if;

// find the next edge
if ledge < d then
    if Degree(V!v2n) eq 2 then
        eda:=Setseq(IncidentEdges(V!v2n) diff {cedge});
eadj:=eda[1];
    end if;
    if (Degree(V!v2n) eq 2) and (!IsLabelled(eadj)) then
        nedge:=eadj;
    else
        lc:=#lblack3;
        while consum(Gn, cyclesGn, lblack3[lc]) do
            lc:=lc-1;
        end while;
        q1:= Index([cyclesGn[t][1]: t in [1..#cyclesGn]], lblack3[lc]);
e1:=E[lc, cyclesGn[q1][2]];
e2:=E[lc, cyclesGn[q1][3]];
e3:=E[lc, cyclesGn[q1][4]];
end if;
if IsLabelled(e1) then
    if IsLabelled(e2) then
        nedge:=e3;
    else
        nedge:=e2;
    end if;
else
    if IsLabelled(e2) then
        if IsLabelled(e3) then
            nedge:=e1;
        else
            nedge:=e3;
        end if;
    else
        nedge:=e1;
    end if;
end if;
cedge:=nedge;
end while;

// take the degree 2 vertices out of vdeg1stuff
// and from lofdeg1or2
//update llab

for s:=1 to #llab do
    if llab[s] eq 0 then
        if Label(V![s]) eq "w" then
            llab[s]:=-2;
        else
            llab:=-3;
        end if;
    end if;
end for;

//eliminate the degree 2 vertices

i:=1;
while i le #vdeg1stuff do
    if Degree(V![vdeg1stuff[i][1]]) eq 2 then
        Remove(~vdeg1stuff, i);
        Exclude(~lofdeg1or2, vdeg1stuff[i][1]);
    else
        i:=i+1;
    end if;
end while;
end while;

//find the Farey symbol

posit;=[];
cus:=[];
for j:=1 to #vdeg1stuff do
  if vdeg1stuff[j][3] eq 0 then
    posit:=posit cat [ vdeg1stuff[j][1]]; 
cus:=cus cat [ vdeg1stuff[j][4]/vdeg1stuff[j][5]]; 
    Remove(~vdeg1stuff, i);
  end if;
  if vdeg1stuff[j][5] eq 0 then
    posit:=posit cat [ vdeg1stuff[j][1]]; 
cus:=cus cat [ vdeg1stuff[j][2]/vdeg1stuff[j][3]]; 
    Remove(~vdeg1stuff, i);
  end if;
end for;

  dum1:=cus[1];
cus[1]:=cus[2];
cus[2]:=dum1;
dum2:=posit[1];
posit[1]:=posit[2];
posit[2]:=dum2;
end if;

intc:=[vdeg1stuff[1][2]/vdeg1stuff[1][3], vdeg1stuff[1][4]/vdeg1stuff[1][5]]; 
if #vdeg1stuff ge 2 then
  for j:=2 to #vdeg1stuff do
    if vdeg1st[j][2]/vdeg1stuff[j][3] notin intc then
      intc cat:=vdeg1st[j][2]/vdeg1stuff[j][3]; 
    end if;
    if vdeg1st[j][4]/vdeg1stuff[j][5] notin intc then
      intc cat:=vdeg1st[j][4]/vdeg1stuff[j][5]; 
    end if;
  end for;
end if;

lcusps:=Sort(cus cat intc);
fareylli:=[llab[pos[1]]];
fareylli cat:=[0: x in [1..#vdeg1stuff]];
for t:=1 to #vdeg1stuff do
  b, pb:=Max([vdeg1stuff[t][2]/vdeg1stuff[t][3], vdeg1stuff[t][4]/vdeg1stuff[t][5]]);
m:=Index(lcusps, b);
fareylli[m]:=llab[vdeg1stuff[t][1]];
end for;
fareylli cat:=[llab[pos[2]]];
return lcusps, fareylli;

//lcusps is the list of cusps without infinity,
//fareylli is the list of Farey symbols(labels)
end function;

The third program tests a group for congruence:

H:=UpperHalfPlaneWithCusps();

mlp:=function (g, a1, a2);

r1:=g[1,1]*a1+g[1,2]*a2;
r2:=g[2,1]*a1+g[2,2]*a2;
if r2 ne 0 then
    return H!(r1/r2);
else
    return H!Infinity();
end if;

end function;

function mi(a,b);
// returns minimum(a, b) and the position of
// the minimum (1 for a, 2 for b)
// a, b are cusps not both of them H!Infinity()

if (a ne H!Infinity()) and (b ne H!Infinity()) then
    c:=Rationals()!Eltseq(ExactValue(a)) ;
d:=Rationals()!Eltseq(ExactValue(b)) ;
if c le d then
    return c, 1;
else
    return d, 2;
end if;
elseif a eq H!Infinity() then
    return Rationals()!Eltseq(ExactValue(b)), 2;
else
    return Rationals()!Eltseq(ExactValue(a)), 1;
end if;

end function;
Num:=function (r);
    return Numerator(r);
end function;

Den:=function (r);
    return Denominator(r);
end function;

mgen1:=function (a1, b1, a2, b2, a3, b3, a4, b4);
    //output the gluing matrix joining a1/b1 and a2/b2
    //to a3/b3 and a4/b4
    M1:=Matrix(Rationals() ,2,2, 
    [a4*b2+a3*b1, -a4*a2-a1*a3,
    b4*b2+b3*b1, -b4*a2-a1*b3]);
    return M1;
end function;

mgen2:=function (a1, b1, a2, b2);
    //output the gluing matrix of order 2
    //joining a1/b1 to a2/b2
    M:=Matrix(Rationals() ,2,2, 
    [a1*b1+a2*b2, -a2*a2-a1*a1,
    b1*b1+b2*b2, -a2*b2-a1*b1]);
    return M;
end function;

mgen3:=function (a1, b1, a2, b2);
    //output the gluing matrix of order 3 joining a1/b1 to a2/b2
    M:=Matrix(Rationals() ,2,2, 
    [a1*b1+a1*b2+a2*b2, -a1*a1-a2*a2-a1*a2, b1*b1+b2*b2+b1*b2, 
    -b1*a1-a2*b2-a2*b1]);
    return M;
end function;

function stuff(V, FS, n);

    //given a sequence of n+1 vertices V and a farey
    //symbol FS of length n+2,
    //outputs the generators, the inequivalent cusp widths,
    //the geometric level

72
Vcomp:=[[-1,0]]; Vcomp cat:=[[\text{Num}(xa), \text{Den}(xa)] : xa \in V]; Vcomp cat:=[[1,0]]; lofeqc:=[[H!*\infty()]]; lofeqc cat:=[[H!(xb)] : xb \in V]; copylofeq:=[[H!*\infty()]] \text{ cat } [H!(xc): xc \in V]; lofeqc cat:=[[H!*\infty()]]; gen:=[\text{MatrixRing}(\text{Rationals}(), 2)];

r:=0;

for i:=1 to n+2 do
  if FS[i] eq -3 then
    gen := gen cat \text{mgen3}(Vcomp[i][1], Vcomp[i][2], Vcomp[i+1][1], Vcomp[i+1][2]);
    lofeqc[i] := Setseq(Seqset(lofeqc[i]) join Seqset(lofeqc[i+1]));
  elif FS[i] eq -2 then
    gen := gen cat \text{mgen2}(Vcomp[i][1], Vcomp[i][2], Vcomp[i+1][1], Vcomp[i+1][2]);
    lofeqc[i] := Setseq(Seqset(lofeqc[i]) join Seqset(lofeqc[i+1]));
  else
    FS[i]:=FS[i]+n+2;
    j:=Index(FS, FS[i]-n-2);
    gen := gen cat \text{mgen1}(Vcomp[i][1], Vcomp[i][2], Vcomp[i+1][1], Vcomp[i+1][2], Vcomp[j][1], Vcomp[j][2], Vcomp[j+1][1], Vcomp[j+1][2]);
    lofeqc[i] := Setseq(Seqset(lofeqc[i]) join Seqset(lofeqc[j+1]));
    lofeqc[i+1] := Setseq(Seqset(lofeqc[i+1]) join Seqset(lofeqc[j]));
    FS[i]:=FS[i]-n-2;
    r:=r+1;
  end if;
end for;

r:=r div 2;

// find the inequivalent cusps

i:=1;
while i lt #lofeqc do
  u:=i+1;
  while (u le #lofeqc) do
    if not (IsEmpty(Seqset(lofeqc[i]) meet Seqset(lofeqc[u]))) then
      lofeqc[i]:=Setseq(Seqset(lofeqc[i]) join Seqset(lofeqc[u]));
    end if;
  end while;
  i:=i+1;
end while;

73
Seqset(lofeqc[u]));
   Remove(~lofeqc, u);
else
   u:=u+1;
end if;
end while;
i:=i+1;
end while;

//calculate the list of cusp widths
s1:= Abs(Vcomp[2][1]*Vcomp[n+2][2]-
         Vcomp[2][2]*Vcomp[n+2][1]);
if (FS[1] eq -3) then
   s1:=s1+1/2;
end if;
if FS[n+2] eq -3 then
   s1:=s1+1/2;
end if;
lofwidths :=[s1];
for i:=2 to n+2 do
   s:= Abs(Vcomp[i-1][1]*Vcomp[i+1][2]-
            Vcomp[i-1][2]*Vcomp[i+1][1]);
   if (FS[i-1] eq -3) then
      s:=s+1/2;
   end if;
   if (FS[i] eq -3) then
      s:=s+1/2;
   end if;
   lofwidths cat:=[s];
end for;

    t:=#lofeqc;
lofineqwids:=[];
for j:=1 to t do
   w:=0;
   for m:=1 to #lofeqc[j] do
      w:=w+lofwidths[Index(copyloeq, lofeqc[j][m])];
   end for;
   lofineqwids:=lofineqwids cat [w];
end for;

lofinwids:=[Integers()| x: x in lofineqwids];
winf:=lofinwids[1];
g:=(r-t+1) div 2;
N := Lcm(lofinwids);
dinde := 0;
for u := 1 to t do
    dinde += lofinwids[u];
end for;

return dinde, g, N, winf, lofinwids, V, FS, gen;
end function;

function belons(matg, winf, V, FS, gen);
// determine whether the matrix matg belongs to the subgroup
// with vertices (cusps) V, Farey symbol FS, generators gen and
// width at infinity winf; if yes it provides the decomposition
// here we are using Lang, Lim and Tan's algorithm

Vcom := [-1, 0];
Vcom cat := [[Num(x), Den(x)] : x in V];
Vcom cat := [1, 0];
F := FreeGroup(#gen + 1);
x := [F.k : k in [1 .. #gen + 1]];
dword := Identity(F);

// x[1] corresponds to the generator [1, winf, 0, 1],
// x[last] corresponds to Identity [1, 0, 0, 1]

lofgedges := [[H!(0), H!Infinity()]];

if FS[1] gt 0 then
    lofgedges cat := [[mlp(gen[1], 0, 1), mlp(gen[1], 1, 0)]];
end if;

if FS[1] eq -2 then
    lofgedges cat := [[H!Infinity(), H!(0)]];
end if;

en := [mlp(matg, 0, 1), mlp(matg, 1, 0)];

while (en not in Seqset(lofgedges)) do
    lend, t := mi(en[1], en[2]);
    while (lend gt V[#V]) or (lend lt V[1]) do
        if lend gt V[#V] then
            matg := gen[#gen]*matg;
            dword := dword*x[#gen]^(-1);
        end if;
        lend, t := mi(en[1], en[2]);
    end while;
end while;
if lend lt V[1] then
    matg:=gen[1]*matg;
    dword:=dword*x[1]^(-1);
end if;
en:=[mlp(matg, 0, 1), mlp(matg, 1, 0)];
lend, t:=mi(en[1], en[2]);
end while;

if (en[3-t] ne H!Infinity()) then
    rend:=Rationals()!Eltseq(ExactValue(en[3-t]));
    ninterf:=#[h: h in V | (h lt lend) and (h gt lend)];
    if ninterf gt 0 then
        return false, Identity(F);
    else
        i:=#[w: w in V | w le lend] +1;
        if (FS[i] gt -3) then
            matg:=gen[i]*matg;
            dword:=dword*x[i]^(-1);
        else
            y:=(Vcom[i][1]+Vcom[i+1][1])/(Vcom[i][2]+Vcom[i+1][2]);
            if ((i eq 1) and (lend eq y) and (rend eq V[1])) or
                ((i eq (#V+1)) and (lend eq V[#V]) and
                    (rend eq y)) then
                return false, Identity(F);
            end if;
            if (i ge 2) and (i le #V) and ((lend eq V[i-1] and
                rend eq y) or (lend eq y and rend eq V[i])) then
                return false, Identity(F);
            end if;
            if (rend le y) then
                matg:=gen[i]^(2)*matg;
                dword:=dword*x[i];
            else
                matg:=gen[i]*matg;
                dword:=dword*x[i]^(-1);
            end if;
        end if;
        end if;
    end if;
else
    if (en notin Seqset(lofgedges)) then
        return false, Identity(F);
    end if;
en:=[mlp(matg, 0, 1), mlp(matg, 1, 0)];
end if;
end while;

m:=Index(lofgedges, en);
if (m ne #gen +1) then
    dword:=dword*x[m];
end if;
return true, dword;
end function;

function establcongsin(N, winf, V, FS, gen);

// establish if the subgroup with vertices V, Farey symbol FS,
// width of infinity winf and generators gen is congruence or
// not using LLT’s algorithm

dindex:=3*(#V-1)+[#x: x in FS| x eq -3];
Bgr:=CongruenceSubgroup(N);
if ((Index(Bgr) mod dindex) ne 0) then
    return false;
else
    geng:=Generators(Bgr);
    i:=1;
    while i le #geng do
        pr, dr:=belons(Matrix(geng[i]), winf, V, FS, gen);
        if pr then
            i:=i+1;
        else
            return false;
        end if;
    end while;
    return true;
end if;
end function;

function resneed(V, FS, n);
//returns "congruence" if the subgroup given by V and FS is
//congruence, otherwise returns "nomcongruence" and the
//index, genus, the geometric level, the width of infinity,
//the list of cusp widths, V, FS and the generators. n=#V-1

a1, a2, a3, a4, a5, a6, a7, a8:=stuff(V, FS, n);

77
if establcongsin(a3, a4, a6, a7, a8) then
    return "congruence";
else
    return "noncongruence", a1, a2, a3, a4, a5, a6, a7, a8;
end if;

end function;
Vita

Constantin C. Caranica was born in April 1972, in Barlad, Romania. He finished his undergraduate studies at Bucharest University in June 1997, with a major in mathematics. He earned a master of science degree in mathematics from Bucharest University in February 2000. In August 2002 he came to Louisiana State University to pursue graduate studies in mathematics where he earned a Master of Science in mathematics in May 2005. He is currently a candidate for the degree of Doctor of Philosophy in mathematics, which will be awarded in August 2009.