2-GROUP BELYI MAPS

A Thesis

Submitted to the Faculty

in partial fulfillment of the requirements for the

degree of

Doctor of Philosophy

in

Mathematics

by Michael James Musty

Guarini School of Graduate and Advanced Studies

DARTMOUTH COLLEGE

Hanover, New Hampshire

July 9, 2019

Examining Committee:

John Voight, Chair

Thomas Shemanske

Carl Pomerance

David P. Roberts

F. Jon Kull, Ph.D.
Dean of the Guarini School of Graduate and Advanced Studies

Abstract

This thesis concerns the explicit computation of Galois Belyi maps $\phi\colon X \to \mathbb{P}^1$ with monodromy group a 2-group, which we call 2-group Belyi maps. The motivation behind computing these maps comes from Beckmann's theorem, which relates the primes of bad reduction of the algebraic curve X to the primes dividing the order of the monodromy group of ϕ. The computation has two parts. The first is a combinatorial computation to enumerate the isomorphism classes of 2-group Belyi maps. The second part is an explicit algorithm to compute equations for the curve X.

Contents

List of Figures

Chapter 1

Introduction

Section 1.1

Motivation

A broad goal of arithmetic geometry is to use tools from algebraic geometry to study questions that arise in number theory. An example of this connection is a theorem due to Belyi, which states that a nice algebraic curve X over the complex numbers can be defined by equations with coefficients in a number field if and only if X admits a Belyi map, a finite cover $\phi\colon X \to \mathbb{P}^1$ unramified outside $\{0, 1, \infty\}$. The way in which Belyi maps capture precisely when a transcendental object is also an algebraic object is just one of the remarkable properties of these covers. The goal of this thesis is to exploit these properties in the particular setting which we now describe.

We begin with a motivating example. Let E be an elliptic curve over $\mathbb{Q}$, let $\ell \in \mathbb{Z}$ be prime, and let $G_{\mathbb{Q}} := \mathrm{Gal}(\mathbb{Q}^{\mathrm{al}} \,|\, \mathbb{Q})$ be the absolute Galois group of $\mathbb{Q}$. There is an action of $G_{\mathbb{Q}}$ on the ℓ-torsion points $E[\ell](\mathbb{Q}^{\mathrm{al}}) \cong \mathbb{Z}/\ell\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z}$ of E, which determines a 2-dimensional mod-ℓ Galois representation

$$\rho\colon G_{\mathbb{Q}} \to \mathrm{Aut}(E[\ell]) \cong \mathrm{GL}_2(\mathbb{Z}/\ell\mathbb{Z}). \tag{1.1.1}$$

The kernel of this representation fixes the field $\mathbb{Q}(E[\ell])$, the ℓ-torsion field of E obtained by adjoining the coordinates of all ℓ-torsion points of E. The geometry of E and the arithmetic of ρ are intimately related. For example, if E has good reduction at a prime $p \neq \ell$, then p will be unramified in the field $\mathbb{Q}(E[\ell])$ by the criterion of Néron–Ogg-Shafarevich.

This relationship between curves and Galois representations extends to higher genus curves. Let X be an irreducible, smooth projective curve of genus $g \geq 1$ over a number field K. The Jacobian variety of X, $J := \mathrm{Jac}(X)$, is an abelian variety over K of dimension g. Again the ℓ-torsion points $J[\ell]$ of J define a mod-ℓ Galois representation and a number field $K(J[\ell])$. As was the case for elliptic curves, if X has good reduction at a prime $\mathfrak{p}$ in K, then $\mathfrak{p}$ is unramified in the ℓ-torsion field $K(J[\ell])$.

The application of Belyi maps to this situation comes from Beckmann's theorem. To state Beckmann's theorem requires a bit more terminology. Associated to every Belyi map $\phi\colon X \to \mathbb{P}^1$ over $\mathbb{C}$, or $\mathbb{Q}^{\mathrm{al}}$ is the monodromy group of the covering obtained by lifting paths around the ramification points on $\mathbb{P}^1$. We say that a Belyi map is Galois if the covering is Galois (equivalently if the degree of the cover equals the size of the monodromy group). We can now state Beckmann's theorem.

Theorem 1.1.2 (Beckmann [1])**.** *Let $\phi\colon X \to \mathbb{P}^1$ be a Galois Belyi map with monodromy group G and suppose p does not divide $\#G$. Then there exists a number field M with the following properties:*

- *p is unramified in M;*
- *the Belyi map ϕ is defined over M; and*
- *ϕ and X have good reduction at all primes $\mathfrak{p}$ of M above p.*

If we insist that G is a 2-group in Beckmann's theorem, then we can hope to find fields M and $M(\mathrm{Jac}(X)[2])$ unramified away from 2. The main interest in finding such a field comes from a conjecture (now theorem) of Gross.

Conjecture 1.1.3 (Gross)**.** *For every prime p, there exists a nonsolvable Galois number field ramified only at p.*

For $p \geq 11$, the existence of such fields is attributed to Serre in [36, 35] and explicit examples are given in [28]. For $p = 3, 5$, the existence of such fields is proved in [16] and an explicit example for $p = 5$ is given in [33]. Existence of such a field in the $p = 7$ case is attributed to [17] using techniques from [16] and some corrections by David P. Roberts. For $p = 2$, the existence of such a field is proven in [15]. A long-term application of the work in this thesis is to find an explicit nonsolvable number field ramified only at 2.

But first, to use Beckmann's theorem to construct interesting number fields, we must have explicit Galois Belyi maps with monodromy group a 2-group. The explicit construction of these objects is the focus of this thesis, and it is these objects we refer to as 2-group Belyi maps. We now summarize the results of this thesis concerning 2-group Belyi maps.

Section 1.2

Main results

Motivated by the discussion in Section 1.1, this work aims to address the task of explicitly computing 2-group Belyi maps up to isomorphism. This computation consists of two main parts:

- enumerating isomorphism classes,
- computing explicit equations for each isomorphism class.

Let $d \in \mathbb{Z}_{\geq 1}$. Isomorphism classes of degree d Belyi maps are in bijection with the set of transitive permutation triples up to simultaneous conjugation in the symmetric group S_d. A transitive permutation triple is a triple of permutations $\sigma := (\sigma_0, \sigma_1, \sigma_\infty) \in S_d^3$ that multiply to the identity and generate a transitive subgroup of S_d. We say that two permutation triples σ, σ' are simultaneously conjugate if there exists $\tau \in S_d$ such that

$$\sigma^\tau := (\tau^{-1}\sigma_0\tau, \tau^{-1}\sigma_1\tau, \tau^{-1}\sigma_\infty\tau) = (\sigma'_0, \sigma'_1, \sigma'_\infty) = \sigma'. \tag{1.2.1}$$

The first main result of this thesis is an explicit algorithm (see Algorithm 3.3.28 in Section 3.3) to compute permutation triples corresponding to all 2-group Belyi maps up to a given degree. We use this algorithm (implemented in Magma [10]) to enumerate all such permutation triples up to conjugation for 2-power degree up to 256. The algorithms in Section 3.3 are used to prove results such as the following theorem.

Theorem 1.2.2. *The following table lists the number of isomorphism classes of per-*

mutation triples corresponding to 2-group Belyi maps of degree d up to 256.

$$
\begin{array}{|c||c|c|c|c|c|c|c|c|c|}
\hline
d & 1 & 2 & 4 & 8 & 16 & 32 & 64 & 128 & 256 \\
\hline
\#\ \textit{permutation triples} & 1 & 3 & 7 & 19 & 55 & 151 & 503 & 1799 & 7175 \\
\hline
\end{array}
\tag{1.2.3}
$$

Other results of this type are detailed in Section 3.4. Having explicit permutation triples allows us to apply techniques from [31] to obtain information about the possible number fields that 2-group Belyi maps can be defined over. In particular, in Section 4.2, we prove the following theorem.

Theorem 1.2.4. *Every 2-group Belyi map of degree $d \leq 256$ is defined over a quadratic extension of an abelian number field ramified only at 2.*

The rest of the results of this thesis pertain to computing equations of 2-group Belyi maps. The first of these is an algorithm to compute 2-group Belyi maps over a finite field $\mathbb{F}_q$ where $q = p^k$ and $p \neq 2$ (see Section 5.4). This algorithm has been implemented in Magma and used to construct a database of all 2-group Belyi maps (up to isomorphism) over $\mathbb{F}_3^{\mathrm{al}}$ up to degree 32.

The other main result is an implementation (similar to the algorithm over $\mathbb{F}_q$) in characteristic zero (see Section 5.5). Although the characteristic zero implementation does not succeed in all cases, it often works well in practice. In particular, the Magma implementation succeeded in computing hundreds of 2-group Belyi maps over $\mathbb{Q}^{\mathrm{al}}$ up to degree 256.

The rest of Chapter 5 is devoted to describing the computations along with interesting examples encountered along the way.

Section 1.3

Navigation

Having motivated and stated the main results, we now provide some explanation of how this thesis is organized and where to find details pertaining to the main results.

Chapter 2 details some of the necessary backgroud material related to Belyi maps, permutation triples, and function fields.

Chapter 3 describes an algorithm to enumerate the isomorphism classes of 2-group Belyi maps using permutation triples (see Algorithm 3.3.11 and Algorithm 3.3.28). These algorithms have been used to enumerate all isomorphism classes of 2-group Belyi maps with degree up to 256. The results of these computations are detailed in Section 3.4.

Chapter 4 explains how the results of computations in Chapter 3 can be used to obtain information on the possible fields of definition of 2-group Belyi maps.

Chapter 5 discusses an algorithm to compute explicit equations for 2-group Belyi maps over finite fields with characteristic not 2 (see Algorithm 5.4.9, Algorithm 5.4.13, and Algorithm 5.4.15). Algorithm 5.5.1 describes the modifications to in characteristic zero.

In Chapter 6 we discuss the future direction of this work.

The source code for the implementation used in this thesis can be found at [30].

Chapter 2

Background on Belyi maps

Section 2.1

Belyi maps and Galois Belyi maps

We now set up the framework to discuss the main mathematical objects of interest in this work.

Definition 2.1.1. A Belyi map is a morphism $\phi\colon X \to \mathbb{P}^1$ of smooth projective algebraic curves over $\mathbb{C}$ that is unramified outside $\{0, 1, \infty\}$. We define the genus of ϕ to be the genus of X.

Definition 2.1.2. Two Belyi maps $\phi\colon X \to \mathbb{P}^1$ and $\phi'\colon X' \to \mathbb{P}^1$ are isomorphic if there exists an isomorphism of curves from X to X' such that the diagram

$$\begin{tikzcd} X \arrow[rr, "\sim"] \arrow[dr, "\phi"'] & & X' \arrow[dl, "\phi'"] \\ & \mathbb{P}^1 & \end{tikzcd} \tag{2.1.3}$$

commutes. If instead we only insist that the isomorphism makes a diagram

$$\begin{tikzcd} X \arrow[r, "\sim"] \arrow[d, "\phi"'] & X' \arrow[d, "\phi'"] \\ \mathbb{P}^1 \arrow[r, "\sim", "\beta"'] & \mathbb{P}^1 \end{tikzcd} \tag{2.1.4}$$

commute, with the bottom map β satisfying $\beta(\{0,1,\infty\}) = \{0,1,\infty\}$, then we say that ϕ and ϕ' are lax isomorphic. Note that a lax isomorphism between ϕ and ϕ' is an isomorphism if and only if β fixes the three points $\{0,1,\infty\}$.

Definition 2.1.5. The triple of partitions $(\lambda_0, \lambda_1, \lambda_\infty)$ encoding the ramification above 0, 1, and ∞ is called the ramification type of ϕ.

Let $\phi\colon X \to \mathbb{P}^1$ be a Belyi map of degree d. Once we label the sheets of the cover and pick a basepoint $\star \notin \{0,1,\infty\}$, we obtain a homomorphism

$$h\colon \pi_1(\mathbb{P}^1 \setminus \{0,1,\infty\}, \star) \to S_d \tag{2.1.6}$$

by lifting paths around the branch points of ϕ.

Definition 2.1.7. The image of h in (2.1.6) is the monodromy group of ϕ, denoted $\mathrm{Mon}(\phi)$.

Definition 2.1.8. A Belyi map $\phi\colon X \to \mathbb{P}^1$ is defined over a number field $K \subseteq \mathbb{C}$ if the defining equations of ϕ and X can be described by polynomial expressions with coefficients in K. We say that K is a field of definition for ϕ.

Theorem 2.1.9 (Belyi's theorem [2])**.** *An algebraic curve X over $\mathbb{C}$ can be defined over a number field if and only if X admits a Belyi map.*

Belyi's theorem implies that every Belyi map can be described by a morphism $\phi\colon X \to \mathbb{P}^1$ of algebraic curves defined over a number field $K \subseteq \mathbb{C}$ (instead of over $\mathbb{C}$). Since maps of curves correspond to function field extensions, we can consider a Belyi map $\phi\colon X \to \mathbb{P}^1$ (defined over K) as equivalently given by an extension of function fields $K(X) \supseteq K(\mathbb{P}^1)$. Note that $K(\mathbb{P}^1)$ is isomorphic the field of rational functions (referred to as the **rational function field** of K) in one variable, say $K(x)$, and $K(X)$ can be written as $K(x)(\alpha)$ for some primitive element α.

The degree of a Belyi map in this setting is the degree of the corresponding function field extension $K(X)$ over the rational function field. Ramification in this setting corresponds to the factorization of ideals (x), $(x-1)$, and $(1/x)$ in maximal orders of $K(X)$. The monodromy group in this setting corresponds to field automorphisms of the Galois closure of $K(X)$ fixing $K(x)$.

Let K^{al} denote an algebraic closure of K in $\mathbb{C}$.

Definition 2.1.10. A Belyi map $\phi\colon X \to \mathbb{P}^1$ defined over K is **(geometrically) Galois** if the corresponding function field extension $K^{\mathrm{al}}(X)$ is a Galois field extension over the rational function field $K^{\mathrm{al}}(x)$.

When ϕ is Galois, the ramification type of ϕ can be more simply encoded by a triple of integers $(a, b, c) \in \mathbb{Z}^3_{\geq 1}$. When ϕ is a Galois Belyi map, we can identify $\mathrm{Mon}(\phi)$ in Definition 2.1.7 as the Galois group $\mathrm{Gal}(K^{\mathrm{al}}(X) \mid K^{\mathrm{al}}(\mathbb{P}^1))$. For this reason, we may also write $\mathrm{Gal}(\phi)$ to denote $\mathrm{Mon}(\phi)$ when ϕ is Galois.

We can now define the main object of interest in this thesis.

Definition 2.1.11. A **2-group Belyi map** is a Galois Belyi map of degree d with monodromy group a 2-group of order d.

For a Galois Belyi map $\mathrm{Mon}(\phi) = \mathrm{Gal}(\phi) \subseteq S_d$ is the regular representation.

Section 2.2

Permutation triples and passports

Definition 2.2.1. A permutation triple of degree $d \in \mathbb{Z}_{\geq 1}$ is a tuple $\sigma = (\sigma_0, \sigma_1, \sigma_\infty) \in S_d^3$ such that $\sigma_\infty \sigma_1 \sigma_0 = 1$. A permutation triple is transitive if the subgroup $\langle \sigma \rangle \leq S_d$ generated by σ is transitive. We say that two permutation triples σ, σ' are simultaneously conjugate if there exists $\tau \in S_d$ such that

$$\sigma^\tau := (\tau^{-1}\sigma_0\tau, \tau^{-1}\sigma_1\tau, \tau^{-1}\sigma_\infty\tau) = (\sigma_0', \sigma_1', \sigma_\infty') = \sigma'. \tag{2.2.2}$$

An automorphism of a permutation triple σ is an element of S_d that simultaneously conjugates σ to itself, i.e., $\mathrm{Aut}(\sigma) = C_{S_d}(\langle \sigma \rangle)$, the centralizer inside S_d.

Lemma 2.2.3. *The set of transitive permutation triples of degree d up to simultaneous conjugation is in bijection with the set of Belyi maps of degree d up to isomorphism.*

Proof. The correspondence is via monodromy [25, Lemma 1.1]; in particular, the monodromy group of a Belyi map is (conjugate in S_d to) the group generated by σ. $\square$

The group $G_\mathbb{Q} := \mathrm{Gal}(\mathbb{Q}^{\mathrm{al}} \,|\, \mathbb{Q})$ acts on Belyi maps by acting on the coefficients of a set of defining equations; under the bijection of Lemma 2.2.3, it thereby acts on the set of transitive permutation triples, but this action is rather mysterious. We can cut this action down to size by identifying some basic invariants, as follows.

Definition 2.2.4. A passport consists of the data $\mathcal{P} = (g, G, \lambda)$ where $g \geq 0$ is an integer, $G \leq S_d$ is a transitive subgroup, and $\lambda = (\lambda_0, \lambda_1, \lambda_\infty)$ is a tuple of partitions λ_s of d for $s = 0, 1, \infty$. These partitions will be also be thought of as a tuple of conjugacy classes $C = (C_0, C_1, C_\infty)$ by cycle type, so we will also write passports as

(g, G, C). Two passports (g, G, C) and (g', G', C') are equal if $g = g'$, $C = C'$, and G is conjugate to G'.

Definition 2.2.5. The passport of a Belyi map $\phi\colon X \to \mathbb{P}^1$ is

$$\mathcal{P}(\phi) = (g(X), \mathrm{Mon}(\phi), (\lambda_0, \lambda_1, \lambda_\infty)) \tag{2.2.6}$$

where $g(X)$ is the genus of X and λ_s is the partition of d obtained by the ramification degrees above $s = 0, 1, \infty$, respectively.

Definition 2.2.7. The passport of a transitive permutation triple σ is

$$\mathcal{P}(\sigma) = (g(\sigma), \langle\sigma\rangle, \lambda(\sigma)) \tag{2.2.8}$$

where (following Riemann–Hurwitz)

$$g(\sigma) := 1 - d + (e(\sigma_0) + e(\sigma_1) + e(\sigma_\infty))/2 \tag{2.2.9}$$

and e is the index of a permutation (d minus the number of orbits), and $\lambda(\sigma)$ is the cycle type of σ_s for $s = 0, 1, \infty$.

Definition 2.2.10. The size of a passport $\mathcal{P}$ is the number of simultaneous conjugacy classes as in (2.2.2) of (necessarily transitive) permutation triples σ with passport $\mathcal{P}$.

The action of $G_{\mathbb{Q}}$ on Belyi maps preserves passports. Therefore, after computing equations for all Belyi maps with a given passport, we can try to identify the Galois orbits of this action.

Definition 2.2.11. We say a passport is irreducible if it has one $G_{\mathbb{Q}}$-orbit and reducible otherwise.

We finish this section with an observation about ramification and the Riemann-Hurwitz formula in the case where we have a Galois Belyi map.

Lemma 2.2.12. *Let σ be a degree d permutation triple corresponding to $\phi\colon X \to \mathbb{P}^1$, a Galois Belyi map with monodromy group G, and let a, b, c be the orders of $\sigma_0, \sigma_1, \sigma_\infty$ respectively. Then σ_0 consists of d/a many a-cycles, σ_1 consists of d/b many b-cycles, and σ_∞ consists of d/c many c-cycles. In particular, for a 2-group Belyi map, a, b, c, and $\#G$ are powers of 2.*

Proof. This follows from the condition that the field extension $K(X)$ is Galois over the rational function field $K(x)$. The Galois action is transitive on primes above any prime of $K(x)$ and in particular implies that the ramified primes all have the same ramification index if they lie above the same prime of $K(x)$. $\square$

Lemma 2.2.12 allows for a simplified version of the Riemann-Hurwitz formula for Galois Belyi maps.

Theorem 2.2.13 (Riemann-Hurwitz)**.** *Let σ be a degree d permutation triple corresponding to $\phi\colon X \to \mathbb{P}^1$, a Galois Belyi map with monodromy group G. Let a, b, c be the orders of $\sigma_0, \sigma_1, \sigma_\infty$ respectively. Then*

$$g(X) = 1 + \frac{\#G}{2}\left(1 - \frac{1}{a} - \frac{1}{b} - \frac{1}{c}\right). \tag{2.2.14}$$

Section 2.3

Triangle groups

Definition 2.3.1. Let $(a, b, c) \in \mathbb{Z}^3_{\geq 1}$. If $1 \in \{a, b, c\}$, then we say the triple is degenerate. Otherwise, we call the triple spherical, Euclidean, or hyperbolic according to whether the value of

$$\chi(a,b,c) = \frac{1}{a} + \frac{1}{b} + \frac{1}{c} - 1 \tag{2.3.2}$$

is positive, zero, or negative. We call this the **geometry type** of the triple. We associate the **geometry**

$$H = \begin{cases} \mathbb{P}^1, & \text{if } \chi(a,b,c) > 0 \\ \mathbb{C}, & \text{if } \chi(a,b,c) = 0 \\ \mathfrak{H}, & \text{if } \chi(a,b,c) < 0 \end{cases} \tag{2.3.3}$$

where $\mathfrak{H}$ denotes the complex upper half-plane.

Definition 2.3.4. For each triple (a,b,c) in Definition 2.3.1 we define the **triangle group**

$$\Delta(a,b,c) = \langle \delta_a, \delta_b, \delta_c \mid \delta_a^a = \delta_b^b = \delta_c^c = \delta_c\delta_b\delta_a = 1\rangle \tag{2.3.5}$$

The **geometry type** of a triangle group $\Delta(a,b,c)$ is the geometry type of the triple (a,b,c).

Definition 2.3.6. The **geometry type** of a Galois Belyi map with ramification type (a,b,c) is the geometry type of (a,b,c).

Definition 2.3.7. Let $\sigma = (\sigma_0, \sigma_1, \sigma_\infty)$ be a transitive permutation triple. Let a,b,c be the orders of $\sigma_0, \sigma_1, \sigma_\infty$ respectively. The **geometry type** of σ is the geometry type of (a,b,c).

The connection between Belyi maps and triangle groups of various geometry types is explained by Lemma 2.3.8.

Lemma 2.3.8. *The set of isomorphism classes of degree-d Belyi maps with ramification type (a,b,c) is in bijection with the set of index d subgroups $\Gamma \leq \Delta(a,b,c)$ up* *to conjugation.*

For a detailed explanation of this relationship see the first part of Section 1 in [25].

Section 2.4

Fields of moduli and fields of definition

We now discuss the background material necessary to describe the conjecture in Chapter 4. This section aims to define a canonical number field associated to a Belyi map which is well-defined on isomorphism classes, bound the degree of this number field, and discuss when a Belyi map can be defined over this field. To start let $\mathrm{Aut}(\mathbb{C})$ denote the field automorphisms of $\mathbb{C}$.

Definition 2.4.1. Let X be an algebraic curve over $\mathbb{C}$. The field of moduli of X, denoted $M(X)$, is the fixed field of $\mathbb{C}$ under the subgroup of field automorphisms

$$\{\tau \in \mathrm{Aut}(\mathbb{C}) : \tau(X) \simeq X\} \tag{2.4.2}$$

where $\tau \in \mathrm{Aut}(\mathbb{C})$ acts on a set of defining equations of X.

Definition 2.4.3. Let $\phi\colon X \to \mathbb{P}^1$ be a Belyi map. The field of moduli of ϕ, denoted $M(\phi)$, is the fixed field of $\mathbb{C}$ under the subgroup of field automorphisms

$$\{\tau \in \mathrm{Aut}(\mathbb{C}) : \tau(\phi) \simeq \phi\} \tag{2.4.4}$$

where $\tau \in \mathrm{Aut}(\mathbb{C})$ acts on a set of defining equations of ϕ and isomorphism is determined by Definition 2.1.2.

Theorem 2.4.5. *Let $\phi : X \to \mathbb{P}^1$ be a Belyi map with passport $\mathcal{P}(\phi)$. Then the degree of $M(\phi)$ is bounded by the size of $\mathcal{P}(\phi)$.*

Proof. Let $\tau \in G_{\mathbb{Q}}$ and consider the conjugated map $\tau(\phi)\colon \tau(X) \to \mathbb{P}^1$. By [24, Appendix] $\tau(\phi)$ is a Belyi map with $\mathcal{P}(\phi) = \mathcal{P}(\tau(\phi))$. Thus $G_{\mathbb{Q}}$ acts on the set of (isomorphism classes of) Belyi maps with a given passport. The degree of $M(\phi)$ is bounded by the index of the stabilizer of ϕ in $G_{\mathbb{Q}}$ under this action, and this index is bounded by the size of $\mathcal{P}(\phi)$. ☐

Recall from Definition 2.1.8 that a Belyi map $\phi\colon X \to \mathbb{P}^1$ is defined over a number field K if ϕ and X can be defined with equations over K. We say that K is a field of definition for ϕ. For a general Belyi map it may not be possible to define the Belyi map over its field of moduli. However, in the setting we are concerned with this is always possible.

Theorem 2.4.6. *A Galois Belyi map can always be defined over its field of moduli.*

Proof. See [13, Proposition 2.5] and [26, Theorem 2.2]. ☐

Chapter 3

Group theory

We begin this chapter with some background on 2-groups and group extensions which we use to explain the algorithms in Section 3.3 on computing explicit permutation triples corresponding to 2-group Belyi maps. We conclude the chapter with Section 3.4 where we summarize the computation of all permutation triples corresponding to 2-group Belyi maps up to degree 256. We also do some coarse data analysis of these results.

Section 3.1

2-groups

In this section we set up some notation and summarize some background material on 2-groups all of which can be found in [18, §6.1].

Let G be a finite group. Denote the centralizer and normalizer of a subet $S \subseteq G$ by $C_G(S)$ and $N_G(S)$ respectively. Let G act on a set X. For $x \in X$ denote the stabilizer of x by $\operatorname{stab}_x(G)$ and the orbit of x by $\operatorname{orb}_x(G)$.

Definition 3.1.1. Let $p \in \mathbb{Z}$ be prime. A finite group G is a p-group if the cardinality of G is a power of p.

Lemma 3.1.2. *The center of a nontrivial p-group is nontrivial.*

Lemma 3.1.3. *Let H be a normal subgroup of a p-group G. Let C be a conjugacy class of G. Then either $C \subseteq H$ or $C \cap H = \emptyset$.*

Lemma 3.1.4. *Let G be a p-group. Let H be a nontrivial normal subgroup of G. Then H intersects the center $Z(G)$ nontrivially.*

Corollary 3.1.5. *Let H be a normal subgroup of order p of a p-group G. Then H is central.*

Lemma 3.1.6. *Let H be a normal subgroup of a p-group G. Then for every divisor p^β of $\#H$, H contains a subgroup H_β, normal in G, of order p^β.*

Lemma 3.1.7. *Every maximal subgroup H of a p-group G has $[G : H] = p$ and $H \trianglelefteq G$.*

Definition 3.1.8. Let G be a finite group. We define a sequence of subgroups of G iteratively as follows. Let $Z_0(G) = \{1\}$ and let $Z_1(G) = Z(G)$. For $i \geq 2$ consider the map

$$\pi \colon G \to G/Z_i(G),$$

and define $Z_{i+1}(G)$ to be the preimage of the center of $G/Z_i(G)$ under π as follows.

$$Z_{i+1}(G) := \pi^{-1}\left(Z\left(\frac{G}{Z_i(G)}\right)\right)$$

Continuing this process produces a sequence of characteristic subgroups of G

$$Z_0(G) \trianglelefteq Z_1(G) \trianglelefteq \cdots \trianglelefteq Z_i(G) \trianglelefteq \cdots$$

called the upper central series of G.

Definition 3.1.9. For $x, y \in G$ a finite group, define the commutator of x and y by $[x, y] := x^{-1}y^{-1}xy$. For subgroups H, K of G define $[H, K] := \langle [h, k] : h \in H \text{ and } k \in K\rangle$. We define the lower central series of G iteratively as follows. Let $G_0 = G$, let $G_1 = [G, G]$, and for $i \geq 1$ define $G_{i+1} = [G, G_i]$.

Definition 3.1.10. A finite group G is nilpotent if the upper central series

$$Z_0(G) \trianglelefteq Z_1(G) \trianglelefteq \cdots \trianglelefteq Z_i(G) \trianglelefteq \cdots$$

has $Z_c(G) = G$ for some nonnegative integer c. The integer c is called the nilpotency class of the nilpotent group G.

Lemma 3.1.11. *A finite group G is nilpotent if and only if $G^c = \{1\}$ for some nonnegative integer c. Moreover, the smallest c such that $G^c = \{1\}$ is the nilpotency class of G and*

$$Z_i(G) \leq G^{c-i-1} \leq Z_{i+1}(G)$$

for all $i \in \{0, 1, \ldots, c-1\}$.

Lemma 3.1.12. *Every p-group is nilpotent.*

Section 3.2

Computing group extensions

In Section 3.3, we will be interested in constructing 2-groups as (central) extensions of other 2-groups. The computations we rely on are implemented in Magma and described in [9]. We now describe the broad strokes of this implementation emphasizing the particular setting we are interested in. The background material concerning group extensions in this section is summarized from [18, §17.4].

Definition 3.2.1. Let G be a finite group and A a finite abelian group. An extension of A by G is a group $\widetilde{G}$ such that the sequence

$$1 \longrightarrow A \xrightarrow{\iota} \widetilde{G} \xrightarrow{\pi} G \longrightarrow 1 \tag{3.2.2}$$

is exact. An extension (3.2.2) is central if $\iota(A)$ is contained in the center of $\widetilde{G}$.

Note that for a group extension (3.2.2) there is an action of G on $\iota(A)$ by conjugation. This action is obtained by choosing a lift in $\widetilde{G}$ and conjugating. Conjugating $\iota(A)$ by this lift is well-defined since A is abelian. From now on we identify A with its image $\iota(A)$ in $\widetilde{G}$ to ease notation. To keep track of the action of G on A we make the following definition.

Definition 3.2.3. Let G be a finite group. A G-module is a finite abelian group A and a group homomorphism $\phi\colon G \to \operatorname{Aut}(A)$.

Proposition 3.2.4. *The extension in (3.2.2) is central if and only if A (identified with its image $\iota(A)$ in $\widetilde{G}$) has trivial G-module structure.*

Proof. Let $a \in A$, let $g \in G$, and let $\widetilde{g} \in \pi^{-1}(g)$. Then g acts on a by

$$g \cdot a = \widetilde{g} a \widetilde{g}^{-1}, \tag{3.2.5}$$

so the action is trivial if and only if $a = \widetilde{g} a \widetilde{g}^{-1}$ for all $\widetilde{g} \in \widetilde{G}$ if and only if $a \in Z(\widetilde{G})$. $\square$

Definition 3.2.6. Two extensions of A by G are equivalent if there exists an isomorphism of groups ϕ making the diagram

$$\begin{tikzcd} 1 \arrow[r] & A \arrow[r] \arrow[d, "\mathrm{id}"] & \widetilde{G}_1 \arrow[r] \arrow[d, "\phi"] & G \arrow[r] \arrow[d, "\mathrm{id}"] & 1 \\ 1 \arrow[r] & A \arrow[r] & \widetilde{G}_2 \arrow[r] & G \arrow[r] & 1 \end{tikzcd} \tag{3.2.7}$$

commute.

Remark 3.2.8. The notion of equivalence from Definition 3.2.6 requires an isomorphism ϕ inducing the identity map on A and G. This definition comes from the G-module structure of A in the sense that equivalent extensions induce (by conjugation) the same G-module structure on A. A weaker notion of equivalence (where we only require ϕ to be any isomorphism from A to A) is useful to characterize the groups $\widetilde{G}$ up to group isomorphism, but will not be used in our situation.

We now look at a motivating example.

Example 3.2.9. Let A be a G-module with $\phi \colon G \to \mathrm{Aut}(A)$ defining the action of G on A. Then we can construct the (external) semidirect product $A \rtimes G$ which is the set $A \times G$ equipped with multiplication defined by

$$(a_1, g_1)(a_2, g_2) := (a_1 + \phi(g_1)(a_2), g_1 g_2).$$

Then $A \rtimes G$ is an extension of A by G

$$1 \longrightarrow A \xrightarrow{\iota} A \rtimes G \xrightarrow{\pi} G \longrightarrow 1$$

where the conjugation action of $\pi^{-1}(G)$ on A (identified with $\iota(A)$) conincides with the original G-module action of A.

We now explain the bijection between equivalence classes of extensions (of A by G) and elements of the group $H^2(G, A)$. The latter can be efficiently computed in Magma [9], and is a crucial part of the algorithms in Section 3.3.

Definition 3.2.10. A function $s\colon G \to \widetilde{G}$ such that $\pi \circ s = \mathrm{id}_G$ is called a section of π. A section is normalized if it maps id_G to $\mathrm{id}_{\widetilde{G}}$. An extension is split if there exists a section s such that s is a homomorphism.

Proposition 3.2.11. *The extension in* (3.2.2) *is split if and only if it is equivalent to*

$$1 \longrightarrow A \xrightarrow{\iota'} A \rtimes G \xrightarrow{\pi'} G \longrightarrow 1$$

where $A \rtimes G$ is the semidirect product of G and A relative to the given action described in Example 3.2.9.

Proof. Suppose $\phi\colon \widetilde{G} \to A \rtimes G$ is an isomorphism inducing the identity maps on A and G. Let $s'\colon G \to A \rtimes G$ be the section $g \mapsto (\mathrm{id}_A, g)$. Then the section $s := \phi^{-1}s'$ is a group homomorphism $s\colon G \to \widetilde{G}$ showing the extension is split. Conversely, assume there exists a section $s\colon G \to \widetilde{G}$ which is a group homomorphism. Then the map $\phi\colon A \rtimes G \to \widetilde{G}$ defined by

$$(a, g) \mapsto \iota(a)s(g)$$

is a bijection. We now show that this map is a group isomorphism by analyzing the multiplication of two elements in the image of ϕ. Let $\iota(a)s(g)$ and $\iota(a')s(g')$ in the image of ϕ. Then from the G-module structure of A we have

$$s(g)\iota(a') = \iota(ga)s(g). \tag{3.2.12}$$

(3.2.12) then implies

$$\iota(a)s(g)\iota(a')s(g') = \iota(a)\iota(ga')s(g)s(g') = \iota(a + ga')s(gg')$$

which is precisely the semidirect product multiplication rule on $A \times G$. □

Proposition 3.2.11 completely describes split extensions. For nonsplit extensions, we must analyze sections that are not homomorphisms. To measure the failure of s to be a homomorphism, we make the following definition.

Definition 3.2.13. Let s be a section of an extension (3.2.2). Let $f\colon G \times G \to A$ be defined by the equation

$$s(g)s(h) = \iota(f(g, h))s(gh). \tag{3.2.14}$$

In other words, $\pi(s(gh)) = \pi(s(g)s(h)) = gh$, so we know that $s(gh)$ and $s(g)s(h)$ differ by an element of $\iota(A)$. We define $f(g, h)$ to be the element $a \in A$ such that (3.2.14) is satisfied. The function f is called the **factor set** for the extension and the section s. A factor set is **normalized** if s is normalized. A normalized factor set f satisfies

$$f(g, 1) = f(1, g) = 0$$

for all $g \in G$.

In Lemma 3.2.20 we will see that a factor set for an extension with a section is a special case of a 2-*cocycle* which we now define.

Definition 3.2.15. A 2-cocycle is a map $f\colon G \times G \to A$ satisfying

$$f(g,h) + f(gh,k) = gf(h,k) + f(g,hk) \tag{3.2.16}$$

for all $g,h,k \in G$. A 2-cocycle f is normalized if

$$f(g,1) = f(1,g) = 0$$

for all $g \in G$.

Definition 3.2.17. A 2-coboundary is a map $f\colon G \times G \to A$ such that there exists $f_1\colon G \to A$ satisfying

$$f(g,h) = gf_1(h) - f_1(gh) + f_1(g) \tag{3.2.18}$$

for all $g,h \in G$.

Definition 3.2.19. Let $Z^2(G,A)$ denote the set of 2-cocycles and $B^2(G,A)$ denote the set of all 2-coboundaries. The second cohohology group $H^2(G,A)$ is defined by the quotient $Z^2(G,A)/B^2(G,A)$.

Lemma 3.2.20. *The factor set f of an extension as in (3.2.2) and a section s is a 2-cocycle.*

Lemma 3.2.21. *Consider an extension as in (3.2.2). Let s and s' be sections of this extension with corresponding factor sets f and f' respectively. Then $f' - f$ is a 2-coboundary.*

Lemma 3.2.20 and Lemma 3.2.21 are explained on page 825 and 826 of [18].

Lemma 3.2.22. *An equivalence class of extensions of A by G determine a unique element of $H^2(G, A)$.*

Proof. Let f be the factor set for any section of the extension. Lemma 3.2.20 shows that $f \in Z^2(G, A)$. Lemma 3.2.21 shows that any other choice of f corresponding to another choice of section differs from f by an element of $B^2(G, A)$. Thus, any single extension of A by G determines a unique cohomology class in $H^2(G, A)$. It remains to show that equivalent extensions determine the same element of $H^2(G, A)$. Consider the equivalent extensions

$$\begin{tikzcd} 1 \arrow[r] & A \arrow[r] \arrow[d, "\mathrm{id}"] & \widetilde{G}_1 \arrow[r, "\pi_1"] \arrow[d, "\phi"] & G \arrow[r] \arrow[d, "\mathrm{id}"] & 1 \\ 1 \arrow[r] & A \arrow[r] & \widetilde{G}_2 \arrow[r, "\pi_2"] & G \arrow[r] & 1. \end{tikzcd} \tag{3.2.23}$$

and let $s_1 \colon G \to \widetilde{G}$ be a section of π_1. From (3.2.23) we have that $s_2 := \phi \circ s_1$ is a section of π_2. Let f_1 and f_2 be the factor sets corresponding to s_1 and s_2 respectively defined by

$$\begin{aligned} s_1(g)s_1(h) &= f_1(g, h)s_1(gh) \\ s_2(g)s_2(h) &= f_2(g, h)s_2(gh) \end{aligned} \tag{3.2.24}$$

for all $g, h \in G$. Chasing through the diagram in (3.2.23) we have

$$\begin{aligned} s_2(g)s_2(h) &= \phi(s_1(g))\phi(s_1(h)) \\ &= \phi(s_1(g)s_1(h)) \\ &= \phi(f_1(g,h)s_1(gh)) \\ &= \phi(f_1(g,h))\phi(s_1(gh)) \\ &= f_1(g,h)s_2(gh) \end{aligned} \tag{3.2.25}$$

where the last equality in (3.2.25) follows from chasing the diagram through the identity map $\mathrm{id}\colon A \to A$. This shows if two extensions are equivalent, then we can define sections for both extensions such that the corresponding factor sets are the same 2-cocycle. In particular, equivalent extensions define the same element of $H^2(G, A)$, which completes the proof. $\square$

Lemma 3.2.22 proves that any factor set for an extension of A by G defines a unique class in $H^2(G, A)$. We now discuss the reverse process of constructing an extension of A by G from a 2-cocycle.

Lemma 3.2.26. *Let $f \in H^2(G, A)$ for some finite group G and G-module A. Then there is an extension*

$$1 \longrightarrow A \xrightarrow{\iota} \widetilde{G} \xrightarrow{\pi} G \longrightarrow 1 \tag{3.2.27}$$

whose factor set is equivalent to f in $H^2(G, A)$.

Proof. Let $\widetilde{G}$ be defined by the set $A \times G$ equipped with the operation

$$(a_1, g_1)(a_2, g_2) = (a_1 + g_1a_2 + f(g_1, g_2), g_1g_2). \tag{3.2.28}$$

$A \times G$ with this operation is a group with identity element $(-f(1,1),1)$ and the inverse of $(a,g) \in A \times G$ given by

$$(a,g)^{-1} = (-g^{-1}a - f(g^{-1},g) - f(1,1), g^{-1}). \tag{3.2.29}$$

We now construct the rest of the extension. Let A^* be defined by

$$A^* := \{(a - f(1,1), 1) : a \in A\}. \tag{3.2.30}$$

A^* is a normal subgroup of $\widetilde{G}$ with the inverses given by

$$(a - f(1,1), 1)^{-1} = (-a - f(1,1), 1) \tag{3.2.31}$$

The isomorphism $\iota\colon A \to A^*$ is defined by

$$a \mapsto (a - f(1,1), 1). \tag{3.2.32}$$

Define $\pi\colon \widetilde{G} \to G$ by the projection $(a,g) \mapsto g$. Now A^*, the image of ι, is contained in $\ker \pi$ since the second coordinate is $1 \in G$ for every element of A^*. Thus (3.2.27) is an extension of A by G. Lastly, let $s\colon G \to \widetilde{G}$ be a section of π and let f_s be the factor set of the extension in (3.2.27). One can show that f_s and f are equal in $H^2(G,A)$.

For more of the details of this proof see page 827 of [18] and page 92 of [11]. $\square$

Remark 3.2.33. The construction in Lemma 3.2.26 generalizes the semidirect product construction in Example 3.2.9.

Theorem 3.2.34. *There is a bijection between equivalence classes of extensions of A by G as in (3.2.2) and elements of $H^2(G, A)$.*

Proof. The least technical proof is by reducing to the normalized setting and is described in detail on page 826 and page 827 in [18]. Below we give a summary of the proof.

Every 2-cocycle f has a normalized 2-cocycle in its cohomology class, so without loss of generality we can assume f is normalized. One then shows that extension constructed from f using Lemma 3.2.26 has normalized factor set equal to f. The last step is showing that this procedure does not depend on the choice of normalized 2-cocycle by showing that as long as the normalized 2-cocycles are in the same cohomology class then the corresponding extensions will be equivalent. □

Having established Theorem 3.2.34, we are interested in computing representatives of $H^2(G, A)$. To do this we use the implementation in Magma described in [9, Cohomology and group extensions]. Describing this implementation in detail is beyond the scope of this work. Instead, we provide Example 3.2.37 at the end of this section detailing how we use these implementations in practice. In our computation of permutation triples corresponding to 2-group Belyi maps in the next section, we will first be concerned with computing extensions of A by G where G is a finite 2-group and $A \simeq \mathbb{Z}/2\mathbb{Z}$. The first consideration in producing these extensions is the possible G-module structures on A. Fortunately, the only G-module structure on A is the trivial action corresponding to the only homomorphism

$$G \to \mathrm{Aut}(\mathbb{Z}/2\mathbb{Z}). \tag{3.2.35}$$

According to Theorem 3.2.34, the equivalent extensions of A by G correspond to elements of $H^2(G, A)$ which can be computed efficiently in Magma and explicitly converted to group extensions as in Example 3.2.37.

Remark 3.2.36. Modifications are required to compute extensions when A is cyclic of prime order p. All possible homomorphisms $G \to \mathrm{Aut}(\mathbb{Z}/p\mathbb{Z}) \cong (\mathbb{Z}/p\mathbb{Z})^\times$ must be computed, and for each G-module A, the corresponding group $H^2(G, A)$ must also be computed. When A has more than one cyclic factor, the situation becomes more complicated. For example, the possible G-module structures on $A \cong \underbrace{\mathbb{Z}/p\mathbb{Z} \times \cdots \times \mathbb{Z}/p\mathbb{Z}}_{d \text{ times}}$ correspond to irreducible $\mathbb{F}_p[G]$-modules of dimension d. Although Magma is capable of computing these modules, we do not require this level of generality for the computations in the next section.

We conclude this section with an example of how we compute group extensions in Magma.

Example 3.2.37. A file with the source code for this example can be found in the repository [30] and can be run from a shell in the repository as follows.

Shell

```
magma thesis_examples/group_extensions.m
```

Let σ be the permutation triple representing the size 1 passport $(41, G, (16, 2, 8))$ where G is the permutation group generated by σ of order 256 and small group database label $(256, 100)$. Suppressing the permutations in σ, the source code is as follows.

Magma

```
...
G := sub<Sym(256)|sigma>;
```

```
assert IsTransitive(G);
assert #G eq 256;
A := TrivialModule(G, GF(2));
CM := CohomologyModule(G, A);
H2 := CohomologyGroup(CM, 2);
extensions := [* *];
for h in H2 do
  E_fp, pi_fp, iota_fp := Extension(CM, h);
  iso, E, K := CosetAction(E_fp, sub<E_fp|Id(E_fp)>);
  iotaE := iota_fp*iso;
  piE := (iso^-1)*pi_fp;
  assert Image(iotaE) eq Kernel(piE);
  assert Image(iotaE).1 in Center(E);
  Append(~extensions, [* E, iotaE, piE , h *]);
end for;
```

We first construct A as a G-module and construct $H^2(G,A)$ using the *Cohomology module* functionality in Magma. In this example $\#H^2(G,A) = 32$. For each cohomology class, we compute the corresponding extension (as a finitely presented group) along with mappings defining the extension. Lastly, we act on the identity coset to obtain the extension as a permutation group along with the appropriate mappings.

Section 3.3

An iterative algorithm to produce generating triples

The aim of this section is to use the group cohomology algorithms, discussed in Section 3.2, to iteratively compute *p-group permutation triples* which we define below.

Definition 3.3.1. Let p be prime. Let $d \in \mathbb{Z}_{\geq 1}$. A p-group permutation triple of degree d is a triple of permutations $\sigma := (\sigma_0, \sigma_1, \sigma_\infty) \in S_d^3$ satisfying

- $\sigma_\infty \sigma_1 \sigma_0 = 1$;
- $G := \langle \sigma \rangle$ is a transitive subgroup of S_d; and
- G is a p-group of order d embedded in S_d via its left regular representation.

The group G is called the monodromy group of σ. We say that two p-group permutation triples σ, σ' are simultaneously conjugate if there exists $\tau \in S_d$ such that

$$\sigma^\tau := (\tau^{-1}\sigma_0\tau, \tau^{-1}\sigma_1\tau, \tau^{-1}\sigma_\infty\tau) = (\sigma_0', \sigma_1', \sigma_\infty') = \sigma'. \tag{3.3.2}$$

Remark 3.3.3. In the process of computing extensions of monodromy groups of p-group Belyi maps we must pass back and forth between permutation groups and abstract groups given by a presentation. Insisting that G embeds into S_d via its regular representation eliminates the ambiguity in embedding a finitely presented group into S_d. This explains the last property in Definition 3.3.1.

Example 3.3.4. When $d = 1$ we define the triple $(\mathrm{id}, \mathrm{id}, \mathrm{id}) \in S_1^3$ to be a p-group permutation triple for every p. This is the unique p-group permutation triple of degree 1.

Example 3.3.5. Let $d = p$ and let σ_s be any p-cycle in S_p. Then we can write 3 distinct p-group permutation triples of degree p:

$$\left(\sigma_s, \sigma_s^{-1}, \mathrm{id}\right), \left(\sigma_s, \mathrm{id}, \sigma_s^{-1}\right), \left(\mathrm{id}, \sigma_s, \sigma_s^{-1}\right). \tag{3.3.6}$$

These are the only p-group permutation triples of degree p up to simultaneous conjugation.

We will describe the algorithms in this section in this slightly more general setting even though the $p = 2$ case is our primary concern.

Notation 3.3.7. Let σ be a p-group permutation triple with monodromy group G and let $A \cong \mathbb{Z}/p\mathbb{Z}$ cyclic of prime order. Let $\widetilde{G}$ be an extension of A by G sitting in the exact sequence

$$1 \longrightarrow A \xrightarrow{\iota} \widetilde{G} \xrightarrow{\pi} G \longrightarrow 1. \tag{3.3.8}$$

By Corollary 3.1.5 the image of ι is a central subgroup of $\widetilde{G}$. The algorithm discussed in this section is iterative, and the base case for this iteration is described in Example 3.3.4.

Definition 3.3.9. Let σ be a p-group permutation triple of degree d with monodromy group G. We say that another p-group permutation triple $\widetilde{\sigma}$ is a p-lift (or simply a lift) of σ if $\widetilde{\sigma}$ is a p-group permutation triple of degree pd with monodromy group $\widetilde{G}$ sitting in the exact sequence in (3.3.8) with $A \cong \mathbb{Z}/p\mathbb{Z}$.

Notation 3.3.10. In Algorithm 3.3.11, the objective will be to lift a p-group permutation triple σ of degree d to p-group permutation triples $\widetilde{\sigma}$ of degree pd. We will denote the set of lifts of σ by $\mathrm{Lifts}(\sigma)$ and write $\mathrm{Lifts}(\sigma)/\!\sim$ to denote the equivalence classes of lifts up to simultaneous conjugation in S_{pd}.

Once we can compute $\mathrm{Lifts}(\sigma)$, the next objective is to enumerate all p-group permutation triples up to a given degree along with the bipartite graph structure determined by lifting triples. More precisely, let $\mathscr{G}_{p^i}$ denote the bipartite graph with the following node sets.

- $\mathscr{G}_{p^i}^{\text{above}}$: the set of isomorphism classes of p-group permutation triples of degree p^i indexed by permutation triples $\widetilde{\sigma}$ up to simultaneous conjugation in S_{p^i}
- $\mathscr{G}_{p^i}^{\text{below}}$: the set of isomorphism classes of p-group permutation triples of degree p^{i-1} indexed by permutation triples σ up to simultaneous conjugation in $S_{p^{i-1}}$

The edge set of $\mathscr{G}_{p^i}$ is defined as follows. For every pair of nodes $(\widetilde{\sigma}, \sigma) \in \mathscr{G}_{p^i}^{\text{above}} \times \mathscr{G}_{p^i}^{\text{below}}$ there is an edge between $\widetilde{\sigma}$ and σ if and only if $\widetilde{\sigma}$ is simultaneously conjugate to a lift of σ.

Now that we have set up some notation and definitions, we now describe the algorithms.

Algorithm 3.3.11. Let p be prime and let $d \in \mathbb{Z}_{\geq 1}$.

Input:

- $\sigma = (\sigma_0, \sigma_1, \sigma_\infty) \in S_d^3$ a p-group permutation triple with monodromy group G
- A a G-module

Output:

All degree p lifts $\widetilde{\sigma}$ of σ up to simultaneous conjugation in S_{pd} where the induced G-module structure on A from the extension in (3.3.8) matches the G-module structure of A given as input.

1. Let $G = \langle \sigma \rangle$ and compute representatives of $H^2(G, A)$.

2. For each $f \in H^2(G, A)$ compute the corresponding extension

$$1 \longrightarrow A \xrightarrow{\iota_f} \widetilde{G}_f \xrightarrow{\pi_f} G \longrightarrow 1 \tag{3.3.12}$$

3. For each extension $\widetilde{G}_f$ in (3.3.12) compute the set

$$\mathrm{Lifts}(\sigma, f) := \left\{ \widetilde{\sigma} : \widetilde{\sigma}_s \in \pi_f^{-1}(\sigma_s) \text{ for } s \in \{0, 1, \infty\},\ \widetilde{\sigma}_\infty \widetilde{\sigma}_1 \widetilde{\sigma}_0 = 1,\ \langle \widetilde{\sigma} \rangle = \widetilde{G}_f \right\} \tag{3.3.13}$$

4. Let

$$\mathrm{Lifts}(\sigma) := \bigcup_{f \in H^2(G,A)} \mathrm{Lifts}(\sigma, f) \tag{3.3.14}$$

5. Quotient $\mathrm{Lifts}(\sigma)$ by the equivalence relation $\sim$ identifying triples in $\mathrm{Lifts}(\sigma)$ that are simultaneously conjugate, as in (3.3.2), to obtain representatives of $\mathrm{Lifts}(\sigma)/\sim$.

Proof of correctness. The computation of $H^2(G, A)$ is described in [9] and implemented in [10]. Theorem 3.2.34 in Section 3.2 implies the following.

- The elements of $H^2(G, A)$ are in bijection with extensions $\widetilde{G}_f$ as in (3.3.12).
- Any lift of σ inducing the G-module structure of A on $\mathbb{Z}/p\mathbb{Z}$ must have monodromy group sitting in an exact sequence obtained in Step 2.

In Step 3 all possible lifts of σ for a single extension $\widetilde{G}_f$ are computed. This is done by computing all $(\#A)^3$ triples mapping to σ under π_f and checking which satisfy the conditions to be a lift of σ. After collecting all the lifts together in Step 4 it is possible there are simultaneously conjugate p-group permutation triples in $\mathrm{Lifts}(\sigma)$. In Step 5 we quotient by simultaneous conjugation to obtain the desired set of lifts as output. $\square$

Algorithm 3.3.11 reduces the problem of finding all lifts of a given p-group permutation triple σ to determining all possible $\langle\sigma\rangle$-module structures on $\mathbb{Z}/p\mathbb{Z}$. Although computations of this sort are implemented in [10], it is especially easy to do when $p = 2$.

Lemma 3.3.15. *Let G be a finite group. The only G-module structure on $\mathbb{Z}/2\mathbb{Z}$ is trivial.*

Proof. A G-module structure on $\mathbb{Z}/2\mathbb{Z}$ is a homomorphism from G to $\mathrm{Aut}(\mathbb{Z}/2\mathbb{Z})$. But $\mathrm{Aut}(\mathbb{Z}/2\mathbb{Z}) \cong (\mathbb{Z}/2\mathbb{Z})^\times$ which is the trivial group, so there is only one such homomorphism. $\square$

For the rest of this section we suppose that $p = 2$. In this special case, Algorithm 3.3.11 does not require a G-module as input since (by Lemma 3.3.15) the trivial G-module structure on $\mathbb{Z}/2\mathbb{Z}$ can be assumed.

Remark 3.3.16. Suppose $p = 2$ using Notation 3.3.7. Then $\iota(A)$ is an order 2 normal subgroup of $\widetilde{G}$. Let α denote the generator of $\iota(A)$. From the perspective of branched covers, α is identifying $2d$ sheets in a degree $2d$ cover down to d sheets in a degree d cover. To relate the degree $2d$ cover corresponding to $\widetilde{G}$ with the degree d cover corresponding to G it is convenient to choose α to be the following product of d

transpositions.

$$\alpha := (1\ d+1)(2\ d+2)\dots(d-1\ 2d-1)(d\ 2d) \tag{3.3.17}$$

The benefit of following this convention can be seen in Example 3.3.18 where we illustrate Algorithm 3.3.11.

Example 3.3.18. In this example we carry out Algorithm 3.3.11 for the degree 2 permutation triple $\sigma = ((1\,2), \mathrm{id}, (1\,2))$. Here $G = \langle\sigma\rangle \cong \mathbb{Z}/2\mathbb{Z}$. In Algorithm 3.3.11 Step 2, we obtain two group extensions $\widetilde{G}_1 \cong \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}/2\mathbb{Z}$ and $\widetilde{G}_2 \cong \mathbb{Z}/4\mathbb{Z}$ sitting in the following exact sequences.

$$\begin{array}{ccccccccc} 1 & \longrightarrow & \mathbb{Z}/2\mathbb{Z} & \xrightarrow{\iota_1} & \widetilde{G}_1 & \xrightarrow{\pi_1} & G & \longrightarrow & 1 \\ 1 & \longrightarrow & \mathbb{Z}/2\mathbb{Z} & \xrightarrow{\iota_2} & \widetilde{G}_2 & \xrightarrow{\pi_2} & G & \longrightarrow & 1 \end{array} \tag{3.3.19}$$

We will consider the two extensions separately.

- For $\widetilde{G}_1$, we can look at preimages of σ_s under the map π_1 to obtain 4 triples that multiply to the identity:

$$\begin{gathered} \Big\{((1\,2)(3\,4), \mathrm{id}, (1\,2)(3\,4)), ((1\,2)(3\,4), (1\,3)(2\,4), (1\,4)(2\,3)), \\ ((1\,4)(2\,3), \mathrm{id}, (1\,4)(2\,3)), ((1\,4)(2\,3), (1\,3)(2\,4), (1\,2)(3\,4))\Big\} \end{gathered} \tag{3.3.20}$$

  Before we continue with the algorithm, let us take a moment to analyze these triples more closely. The generator α of $\iota(\mathbb{Z}/2\mathbb{Z})$ in $\widetilde{G}_1$ is $(1\,3)(2\,4)$. Each triple in (3.3.20) must act on the blocks $\left\{\boxed{1\,3}, \boxed{2\,4}\right\}$ so that the induced permutations of these blocks is the same as the corresponding permutation in σ. For

$$(\widetilde{\sigma}_0, \widetilde{\sigma}_1, \widetilde{\sigma}_\infty) = ((1\,2)(3\,4), (1\,3)(2\,4), (1\,4)(2\,4)) \tag{3.3.21}$$

we have $\widetilde{\sigma}_0\left(\boxed{1\,3}\right) = \boxed{2\,4}$ and $\widetilde{\sigma}_0\left(\boxed{2\,4}\right) = \boxed{1\,3}$ so that the induced permutation of blocks is

$$\left(\boxed{1\,3}, \boxed{2\,4}\right) \tag{3.3.22}$$

which is the same as the permutation $\sigma_0 = (1\,2)$ (as long as we identity $\boxed{1\,3}$ with 1 and $\boxed{2\,4}$ with 2). Insisting α has the form in Remark 3.3.16 allows us to label blocks by reducing modulo d as in (3.3.22). The last requirement for a triple $\widetilde{\sigma}$ in Equaiton 3.3.20 to be in $\mathrm{Lifts}(\sigma, \widetilde{G}_1)$ is that $\widetilde{\sigma}$ generates $\widetilde{G}_1$. We obtain $\mathrm{Lifts}(\sigma, \widetilde{G}_1)$ to be

$$\Big\{((1\,2)(3\,4), (1\,3)(2\,4), (1\,4)(2\,3)), ((1\,4)(2\,3), (1\,3)(2\,4), (1\,2)(3\,4))\Big\} \tag{3.3.23}$$

- For $\widetilde{G}_2$, we obtain $\mathrm{Lifts}(\sigma, \widetilde{G}_2)$ to be

$$\begin{aligned}&\Big\{((1\,4\,3\,2), \mathrm{id}, (1\,2\,3\,4)), ((1\,2\,3\,4), (1\,3)(2\,4), (1\,2\,3\,4)),\\ &((1\,2\,3\,4), \mathrm{id}, (1\,4\,3\,2)), ((1\,4\,3\,2), (1\,3)(2\,4), (1\,4\,3\,2))\Big\}\end{aligned} \tag{3.3.24}$$

At the end of Step 4 we have that $\mathrm{Lifts}(\sigma)$ contains the 2 triples in (3.3.23) and the 4 triples in (3.3.24). Lastly, in Step 5 we quotient by simultaneous conjugation to obtain the 3 triples

$$\begin{aligned}\mathrm{Lifts}(\sigma)/\!\sim\, = \Big\{&((1\,2)(3\,4), (1\,3)(2\,4), (1\,4)(2\,3)),\\ &((1\,4\,3\,2), \mathrm{id}, (1\,2\,3\,4)),\\ &((1\,2\,3\,4), (1\,3)(2\,4), (1\,2\,3\,4))\Big\}\end{aligned} \tag{3.3.25}$$

as output.

Now that we have an algorithm to find all lifts of a single permutation triple, we now describe how to use this to compute all isomorpism classes of 2-group permutation triples up to a given degree. In the algorithms to follow, we are concerned with constructing the bipartite graphs $\mathscr{G}_{2^i}$ defined in Notation 3.3.10.

Algorithm 3.3.26. Let $p = 2$ and the notation be as in 3.3.7 and 3.3.10. Then we can construct $\mathscr{G}_2$ as follows.

- The set of nodes $\mathscr{G}_2^{\text{below}}$ consists of a single triple $(\mathrm{id}, \mathrm{id}, \mathrm{id}) \in S_1^3$
- The set of nodes $\mathscr{G}_2^{\text{above}}$ consists of 3 triples described in Example 3.3.5.
- The edge set of $\mathscr{G}_2$ consists of 3 edges (i.e. it is the complete bipartite graph for the sets $\mathscr{G}_2^{\text{below}}$ and $\mathscr{G}_2^{\text{above}}$)

Proof of correctness. By definition, the 3 degree 2 permutation triples from Example 3.3.5 are the only 2-group permutation triples of degree 2. These are all lifts of the unique 2-group permutation triple (in Example 3.3.4) via the extension

$$1 \longrightarrow \mathbb{Z}/2\mathbb{Z} \xrightarrow{\mathrm{id}} \mathbb{Z}/2\mathbb{Z} \xrightarrow{\pi} \{\mathrm{id}\} \longrightarrow 1 \tag{3.3.27}$$

$\square$

Having constructed $\mathscr{G}_2$, we now describe the iterative process to compute $\mathscr{G}_{2^i}$ from $\mathscr{G}_{2^{i-1}}$.

Algorithm 3.3.28. Let $p = 2$ and the notation be as in 3.3.7 and 3.3.10. This algorithm describes the process of computing $\mathscr{G}_{2^i}$ given $\mathscr{G}_{2^{i-1}}$.

Input: The bipartite graph $\mathscr{G}_{2^{i-1}}$

Output: The bipartite graph $\mathscr{G}_{2^i}$

1. For every $\sigma \in \mathscr{G}^{\text{above}}_{2^{i-1}}$ apply Algorithm 3.3.11 to obtain the set $\text{Lifts}(\sigma)/\!\sim$ for each σ. Combine these lifts into a single set

$$\text{Lifts}(\mathscr{G}_{2^{i-1}}) := \bigcup_{\sigma \in \mathscr{G}^{\text{above}}_{2^{i-1}}} \text{Lifts}(\sigma) \tag{3.3.29}$$

2. Compute $\text{Lifts}(\mathscr{G}_{2^{i-1}})/\!\sim$ which we define to be the equivalence classes of $\text{Lifts}(\mathscr{G}_{2^{i-1}})$ where two triples $\widetilde{\sigma}$ and $\widetilde{\sigma}'$ in $\text{Lifts}(\mathscr{G}_{2^{i-1}})$ are equivalent if and only if they are simultaneously conjugate in S_{2^i}. Denote the equivalence class of $\widetilde{\sigma} \in \text{Lifts}(\mathscr{G}_{2^{i-1}})$ by $[\widetilde{\sigma}] \in \text{Lifts}(\mathscr{G}_{2^{i-1}})/\!\sim$.

3. Define $\mathscr{G}^{\text{below}}_{2^i} := \mathscr{G}^{\text{above}}_{2^{i-1}}$. Define $\mathscr{G}^{\text{above}}_{2^i}$ by choosing a single representative for each equivalence class of $\text{Lifts}(\mathscr{G}_{2^{i-1}})/\!\sim$. This defines the nodes of $\mathscr{G}_{2^i}$.

4. For every pair $(\widetilde{\sigma}, \sigma) \in \mathscr{G}^{\text{above}}_{2^i} \times \mathscr{G}^{\text{below}}_{2^i}$ place an edge between $\widetilde{\sigma}$ and σ if and only if there is a triple in the equivalence class $[\widetilde{\sigma}] \in \text{Lifts}(\mathscr{G}_{2^{i-1}})/\!\sim$ that is a lift of σ.

5. Return $\mathscr{G}_{2^i}$ as output.

Proof of correctness. Since 2-groups are nilpotent, every 2-group permutation triple of degree 2^i is the lift of at least one 2-group permutation triple of degree 2^{i-1}. Let $\widetilde{\sigma} \in \mathscr{G}^{\text{above}}_{2^i}$ be an arbitrary representative of an isomorphism class of 2-group permutation triples of degree 2^i contained in $\text{Lifts}(\sigma)$ for some degree 2^{i-1} triple σ. Let σ' denote the representative in $\mathscr{G}^{\text{above}}_{2^{i-1}}$ that is simultaneously conjugate to σ. Algorithm 3.3.11 ensures that there is a 2-group permutation triple $\widetilde{\sigma}'$ of degree 2^i in $\text{Lifts}(\sigma')$ that is simultaneously conjugate to $\widetilde{\sigma}$. Thus, $\text{Lifts}(\mathscr{G}_{2^{i-1}})$ computed in Step 1 contains at least one triple for every isomorphism class of 2-group permutation triples of degree 2^i. It is, however, possible for $\mathrm{Lifts}(\mathscr{G}_{2^{i-1}})$ to contain simultaneously conjugate triples arising as lifts of different triples in $\mathscr{G}_{2^{i-1}}^{\mathrm{above}}$. Step 2 quotients $\mathrm{Lifts}(\mathscr{G}_{2^{i-1}})$ by simultaneous conjugation and Steps 3 and 4 define the desired graph $\mathscr{G}_{2^i}$ in such a way that the edge structure of the lifts is preserved. □

Algorithm 3.3.26 combined with Algorithm 3.3.28 allows us to compute

$$\mathscr{G}_2, \mathscr{G}_4, \dots, \mathscr{G}_{2^i}, \dots, \mathscr{G}_{2^m} \tag{3.3.30}$$

up to any degree $d = 2^m$. A Magma implementation of Algorithms 3.3.11, 3.3.26, and 3.3.28 can be found at [30]. In the next section we discuss the results of these computations.

Section 3.4

Results of computations

In this section we discuss the Magma implementation of Algorithms 3.3.11, 3.3.26, and 3.3.28 available at [30] where the techniques of this chapter are used to tabulate a database of 2-group permutation triples up to degree 256. This computation took roughly 50 CPU hours on a standard desktop. The majority of this time is spent checking conjugacy of degree 256 permutation triples. This database consists of roughly 340MB worth of text files. We devote the rest of this section to summarizing the results of these computations.

Theorem 3.4.1. *The following table lists the number of isomorphism classes of* 2-

group permutation triples of degree d up to 256.

d	1	2	4	8	16	32	64	128	256
# permutation triples	1	3	7	19	55	151	503	1799	7175

(3.4.2)

Theorem 3.4.3. *The following table lists the number of passports of* 2*-group permutation triples of degree d up to* 256.

d	1	2	4	8	16	32	64	128	256
# passports	1	3	7	16	41	96	267	834	2893

(3.4.4)

Theorem 3.4.5. *The following table lists the number of lax passports of* 2*-group permutation triples of degree d up to* 256.

d	1	2	4	8	16	32	64	128	256
# lax passports	1	1	3	6	14	31	85	257	882

(3.4.6)

Theorem 3.4.7. *The following table lists the number of* 2*-group permutation triples up to degree* 256 *with* $\{\text{order}(\sigma_s) : s \in \{0, 1, \infty\}\}$ *equal to* $\{a, b, c\}$ *as sets.*

(a, b, c)	*# permutation triples*
$(1, 1, 1)$	1
$(1, 2, 2)$	3
$(1, 4, 4)$	3
$(1, 8, 8)$	3
$(1, 16, 16)$	3
$(1, 32, 32)$	3

$(1, 64, 64)$	3
$(1, 128, 128)$	3
$(1, 256, 256)$	3
$(2, 2, 2)$	1
$(2, 2, 4)$	24
$(2, 2, 8)$	132
$(2, 2, 16)$	144
$(2, 2, 32)$	60
$(2, 2, 64)$	24
$(2, 2, 128)$	12
$(2, 4, 4)$	24
$(2, 4, 8)$	78
$(2, 4, 16)$	78
$(2, 4, 32)$	30
$(2, 4, 64)$	18
$(2, 4, 128)$	6
$(2, 8, 8)$	132
$(2, 8, 16)$	156
$(2, 8, 32)$	60
$(2, 8, 64)$	12
$(2, 16, 16)$	144
$(2, 16, 32)$	36
$(2, 32, 32)$	60

$(2, 64, 64)$	24
$(2, 128, 128)$	12
$(2, 256, 256)$	3
$(4, 4, 4)$	65
$(4, 4, 8)$	1581
$(4, 4, 16)$	969
$(4, 4, 32)$	225
$(4, 4, 64)$	69
$(4, 4, 128)$	15
$(4, 8, 8)$	1581
$(4, 8, 16)$	960
$(4, 8, 32)$	168
$(4, 8, 64)$	24
$(4, 16, 16)$	969
$(4, 16, 32)$	84
$(4, 32, 32)$	225
$(4, 64, 64)$	69
$(4, 128, 128)$	15
$(4, 256, 256)$	6
$(8, 8, 8)$	726
$(8, 8, 16)$	1542
$(8, 8, 32)$	378
$(8, 8, 64)$	78

$(8, 16, 16)$	1542
$(8, 16, 32)$	72
$(8, 32, 32)$	378
$(8, 64, 64)$	78
$(8, 128, 128)$	24
$(8, 256, 256)$	12
$(16, 16, 16)$	136
$(16, 16, 32)$	552
$(16, 32, 32)$	552
$(16, 64, 64)$	144
$(16, 128, 128)$	48
$(16, 256, 256)$	24
$(32, 64, 64)$	288
$(32, 128, 128)$	96
$(32, 256, 256)$	48
$(64, 128, 128)$	192
$(64, 256, 256)$	96
$(128, 256, 256)$	192

Remark 3.4.8. The above table in Theorem 3.4.7 has a pattern. The table entries corresponding to $(8, 8, 16)$ and $(8, 16, 16)$ are equal and the table entries corresponding to $(16, 16, 32)$ and $(16, 32, 32)$ are equal. This appears to be more than just a coincidence, but at this point we do not have an explanation of this phenomenon.

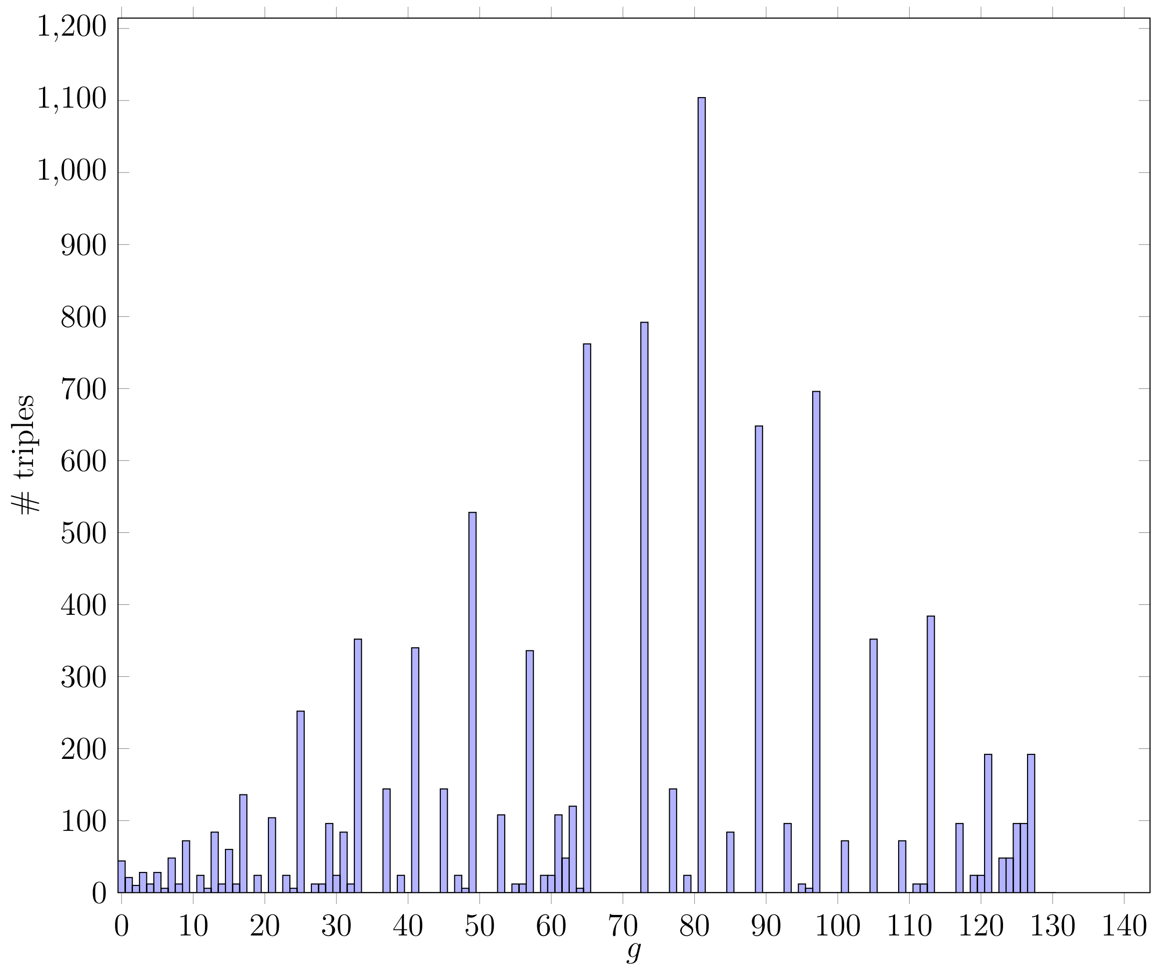

Figure 3.4.8: Distribution of genera up to degree 256

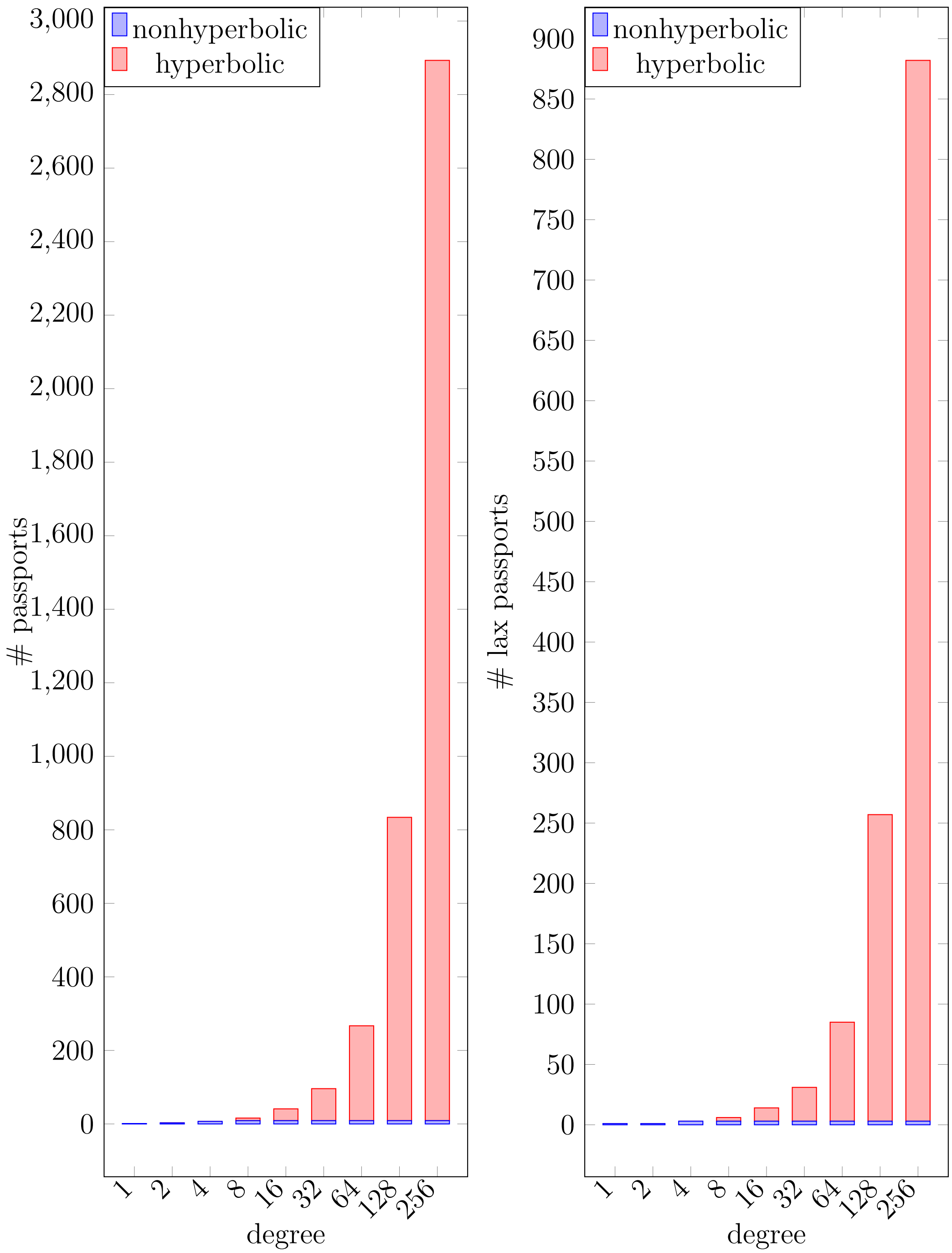

Figure 3.4.8: # nonhyperbolic and hyperbolic passports by degree (left), and # nonhyperbolic and hyperbolic lax passports by degree (right).

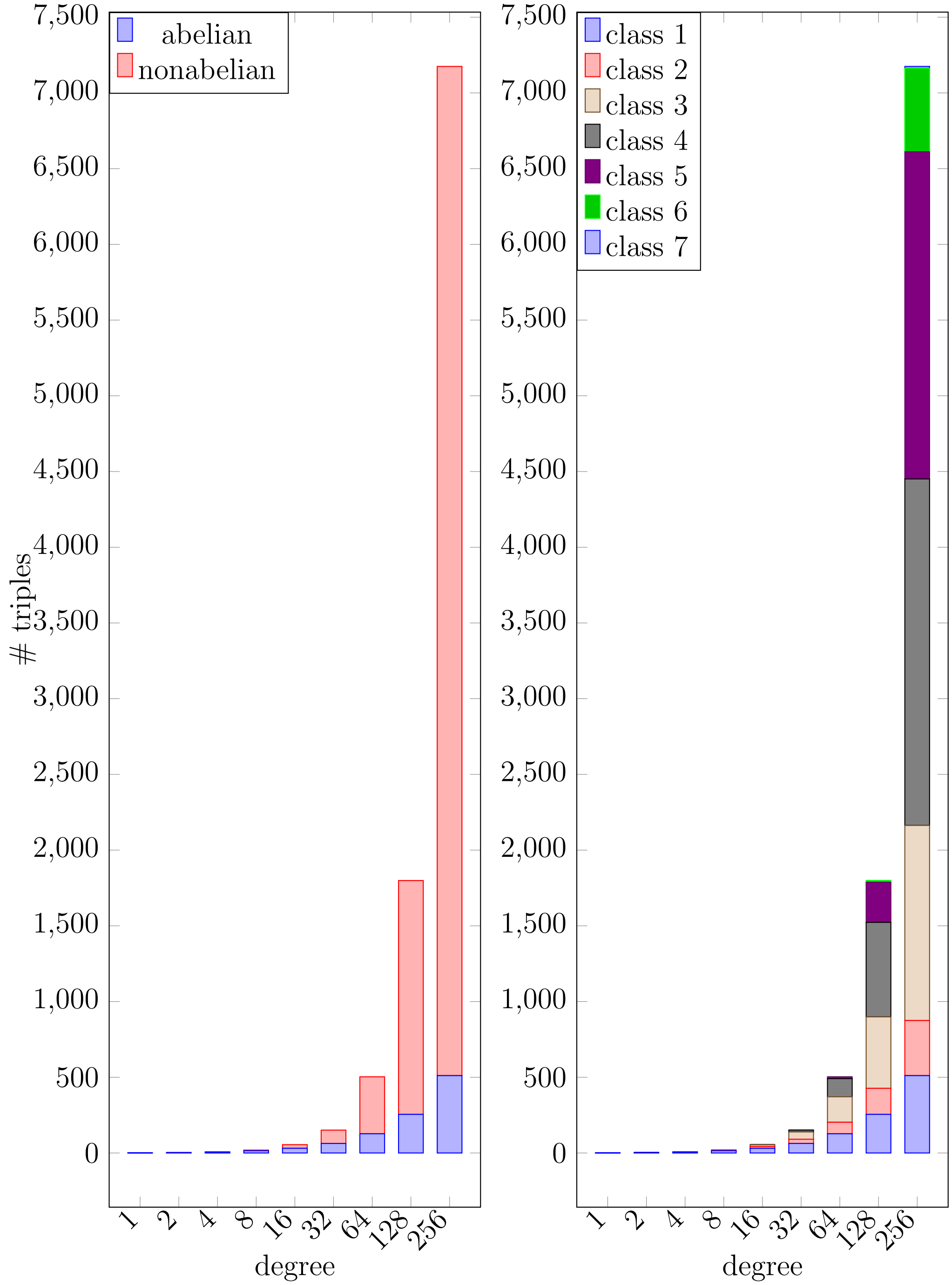

Figure 3.4.8: # permutation triples by degree with abelian and nonabelian monodromy groups (left) and # permutation triples by degree with monodromy groups of various nilpotency classes (right).

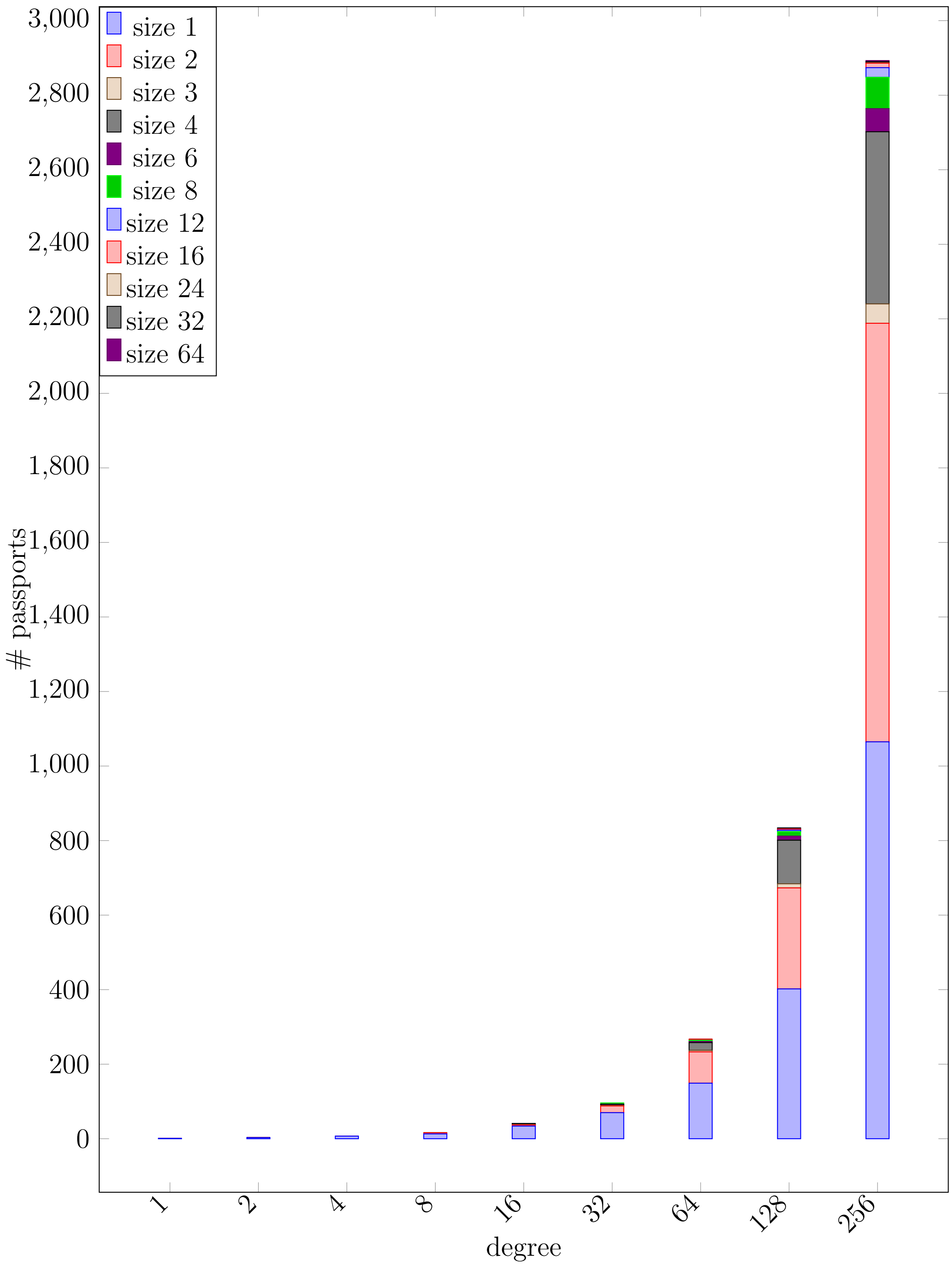

Figure 3.4.8: # passports of various sizes by degree

Chapter 4

Fields of definition of 2-group Belyi maps

Recall from Definition 2.1.8, that a field of definition of a Belyi map $\phi\colon X \to \mathbb{P}^1$ is a number field $K \subseteq \mathbb{C}$ such that X and ϕ are defined using algebraic equations having coefficients in K. Recall from Theorem 2.4.5 that the moduli field of a Belyi map has degree bounded by the size of its passport. This is obtained from the action of $\mathrm{Gal}(\mathbb{Q}^{\mathrm{al}} \,|\, \mathbb{Q})$ on the Belyi maps with a given passport. There is an analogous action of $\mathrm{Gal}(\mathbb{Q}^{\mathrm{al}} \,|\, \mathbb{Q}^{\mathrm{ab}})$ on *refined passports* which we define and compute in this chapter.

Section 4.1

Refined passports

Let σ be a 2-group permutation triple. Recall, from Definition 2.2.4, that the passport of σ consists of the data $(g(\sigma), \langle\sigma\rangle, \lambda(\sigma))$ where $g(\sigma)$ is the genus, $\langle\sigma\rangle$ is the monodromy group (2-group in its regular representation) as a subgroup of S_d, and $\lambda(\sigma)$ is the triple of partitions specifying the three ordered S_d conjugacy classes C_0, C_1, C_∞ of $\sigma_0, \sigma_1, \sigma_\infty$ respectively. Let $\mathcal{P}$ be the passport of σ. The size of $\mathcal{P}$ is the cardinality of the set

$$\Sigma_{\mathcal{P}} = \{(\sigma_0, \sigma_1, \sigma_\infty) \in C_0 \times C_1 \times C_\infty : \sigma_\infty\sigma_1\sigma_0 = 1 \text{ and } \langle \sigma_0, \sigma_1, \sigma_\infty \rangle = G\}/\sim \quad (4.1.1)$$

where $(\sigma_0, \sigma_1, \sigma_\infty) \sim (\sigma_0', \sigma_1', \sigma_\infty')$ if the triples are simultaneously conjugate by an element of S_d. By Theorem 2.4.5, the cardinality of $\Sigma_{\mathcal{P}}$ bounds the field of moduli of the Belyi map corresponding to σ.

Let G be a transitive subgroup of S_d and let C be a conjugacy class of S_d. Then C can be partitioned into conjugacy classes of G. To analyze conjugacy in G we make the following definition.

Definition 4.1.2. A refined passport $\mathscr{P}$ consists of the data (g, G, c) where $g \geq 0$ is an integer, $G \leq S_d$ is a transitive subgroup, and $c = (c_0, c_1, c_\infty)$ is a triple of conjugacy classes of G. For a refined passport $\mathscr{P}$ consider the set

$$\Sigma_{\mathscr{P}} = \{(\sigma_0, \sigma_1, \sigma_\infty) \in c_0 \times c_1 \times c_\infty : \sigma_\infty\sigma_1\sigma_0 = 1 \text{ and } \langle \sigma_0, \sigma_1, \sigma_\infty \rangle = G\}/\sim \quad (4.1.3)$$

where $(\sigma_0, \sigma_1, \sigma_\infty) \sim (\sigma_0', \sigma_1', \sigma_\infty')$ if and only if there exists $\alpha \in \mathrm{Aut}(G)$ with $\alpha(\sigma_s) = \sigma_s'$ for $s \in \{0, 1, \infty\}$.

Let σ be a permutation triple and let c_s denote the conjugacy class of $\langle\sigma\rangle$ containing σ_s for $s \in \{0, 1, \infty\}$. We define the refined passport of σ to be

$$\mathscr{P}(\sigma) = (g(\sigma), \langle\sigma\rangle, (c_0, c_1, c_\infty)). \quad (4.1.4)$$

Section 4.2

Computing refined passports

Let σ be a 2-group permutation triple. Let $\mathcal{P}$ and $\mathscr{P}$ denote the passport and refined passport of σ respectively. Let $\Sigma_{\mathcal{P}}$ and $\Sigma_{\mathscr{P}}$ denote the sets in (4.1.1) and (4.1.3) respectively.

Chapter 3 provides us with an explicit list of all 2-group permutation triples (up to simultaneous conjugation in S_d) for fixed degree. Using techniques from Musty, Schiavone, Sijsling, and Voight in [31], we now describe the computation of $\Sigma_{\mathscr{P}(\sigma)}$ for every 2-group permutation triple σ of degree d for $d \leq 256$.

The main tool for efficiently computing refined passports comes from the *passport lemma*, [31, Lemma 2.2.1], which we now state.

Lemma 4.2.1 (Passport lemma)**.** *Let S be a group, let $G \leq S$ be a subgroup, let $N := N_S(G)$ be the normalizer of G in S, and let C_0, C_1 be conjugacy classes in N represented by $\tau_0, \tau_1 \in G$. Let $C_N(g)$ denote the centralizer of g in N. Let*

$$U := \{(\sigma_0, \sigma_1) \in C_0 \times C_1 : \langle \sigma_0, \sigma_1 \rangle \subseteq G\}/\sim \tag{4.2.2}$$

where $\sim$ indicates simultaneous conjugation by elements in S. Then the map

$$\begin{aligned} u\colon C_N(\tau_0)\backslash N/C_N(\tau_1) &\to U \\ C_N(\tau_0)\nu C_N(\tau_1) &\mapsto [(\tau_0, \nu\tau_1\nu^{-1})] \end{aligned} \tag{4.2.3}$$

is surjective, and for all $[(\sigma_0, \sigma_1)] \in U$ such that $\langle \sigma_0, \sigma_1 \rangle = G$, there is a unique preimage under u.

With the passport lemma in hand, we can now describe an efficient algorithm to compute refined passports.

Algorithm 4.2.4.

Input: σ a 2-group permutation triple

Output: Refined passport representatives for $\mathscr{P}(\sigma)$

1. Compute the set U from the passport lemma with $\tau_0 = \sigma_0$, $\tau_1 = \sigma_1$, and $S = G$.

2. Let $U' := \{(g_0, g_1) \in U : \langle g_0, g_1 \rangle = G\}$.

3. Extend all pairs (g_0, g_1) in U' from Step 2 to triples (g_0, g_1, g_∞) satisfying $g_\infty g_1 g_0 = 1$. Let T denote the set of triples obtained in this way from the pairs of U'.

4. Let C_0, C_1, C_∞ denote the conjugacy classes of $\sigma_0, \sigma_1, \sigma_\infty$ respectively and let $T' := \{(g_0, g_1, g_\infty) \in T : g_\infty \in C_\infty\}$.

5. Quotient T' by outer automorphisms of G. That is, for every pair of triples (g_0, g_1, g_∞) and (g_0', g_1', g_∞') in T with $(g_0', g_1', g_\infty') = (\alpha(g_0), \alpha(g_1), \alpha(g_\infty))$ for some $\alpha \in \mathrm{Out}(G)$ only keep one such triple in T. Return this quotient of T' as output.

Proof of correctness. By the passport lemma, we have that T' is equal to the set

$$\{(g_0, g_1, g_\infty) \in C_0 \times C_1 \times C_\infty : g_\infty g_1 g_0 = 1, \text{ and } \langle g_0, g_1, g_\infty \rangle = G\}/\sim \qquad (4.2.5)$$

where $\sim$ denotes simultaneous conjugation in G. The refined passport of σ can now be obtained from T' by eliminating redundant triples that can be identified by an outer automorphism of G. This is done in the last step of the algorithm and completes the proof. $\square$

Applying Algorithm 4.2.4 to the 2-group permutation triples computed from Section 3.3 yields the following result about the sizes of refined passports.

Theorem 4.2.6. *Every* 2*-group permutation triple of degree* d *with* $d \leq 64$ *has refined passport size* 1*. There are* 48 2*-group permutation triples (up to simultaneous conjugation) of degree* 128 *with refined passport size* 2 *and the rest have refined passport size* 1*. There are* 288 2*-group permutation triples (up to simultaneous conjugation) of degree* 256 *with refined passport size* 2 *and the rest have refined passport size* 1*.*

Theorem 4.2.6, together with refined field of definition equal to field of moduli for Galois Belyi maps and a strong version of Beckmann's theorem imply the following.

Theorem 4.2.7. *Every* 2*-group Belyi map of degree* d *with* $d \leq 256$ *is defined over a quadratic extension of an abelian extension of* $\mathbb{Q}$ *ramified only at* 2*.*

For a visual representation of Theorem 4.2.6 see Figure 4.2. The refined passport sizes in Figure 4.2 appear small compared to the passport sizes in Figure 3.4.

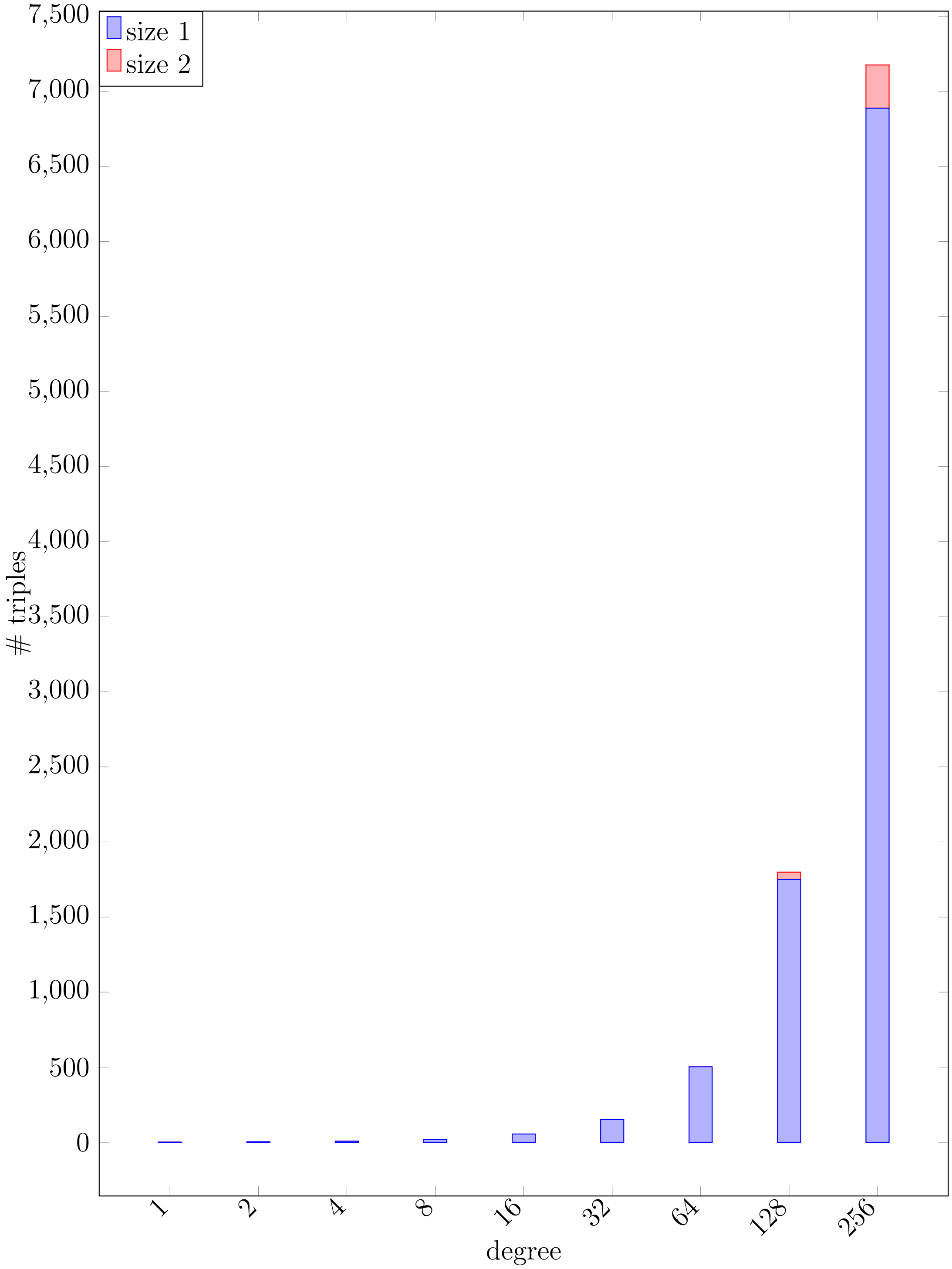

Figure 4.2.7: # permutation triples with refined passports of various sizes by degree

Chapter 5

Computing equations

In this chapter we discuss how to compute equations for 2-group Belyi maps corresponding to the 2-group permutation triples computed in Chapter 3. As was the case for computing the permutation triples, the algorithm to compute equations follows an iterative approach. In this chapter we construct the 2-group Belyi maps as towers of quadratic extensions of function fields. We begin in Section 5.1 by discussing the analogous situation over number fields. In Section 5.2 and Section 5.3 we discuss the relevant background about algebraic function fields. The algorithms to compute equations for 2-group Belyi maps (over $\mathbb{F}_q$) are described in Section 5.4, the implementation in characteristic zero is detailed in Section 5.5, and the results of these computations can be found in Section 5.6.

Section 5.1

Quadratic extensions of number fields

By way of motivation, let F be a number field and let $\mathbb{Z}_F$ denote the ring of integers of F. Kummer theory tells us that quadratic extensions of F are in bijection with nontrivial cosets $dF^{\times 2}$ in the quotient $F^\times/F^{\times 2}$; such a coset defines a quadratic extension $F(\sqrt{d})$. Conversely, let $F(\alpha)$ be a quadratic extension of F. The discriminant of the minimal polynomial of α defines the bijection in the other direction.

Let $\mathrm{Pl}(F)$ denote the set of places of F and let S_∞ denote the archimedean places. For $v \in \mathrm{Pl}(F) \setminus S_\infty$ let $\mathfrak{p}_v$ be the prime ideal of $\mathbb{Z}_F$ corresponding to v. Let $S \subset \mathrm{Pl}(F) \setminus S_\infty$ be a finite set of nonarchimedean places, and further suppose that each place of S has odd order residue field. We aim to answer the following question.

Question 5.1.1. How do we construct a quadratic extension of F ramified at $\mathfrak{p}_v$ for all $v \in S$ and unramified at all nonarchimedean places outside of S? If so, then how *unique* is the construction?

To formulate this question more clearly, let $\mathfrak{a} := \prod_{v \in S} \mathfrak{p}_v$ encode the primes we want to ramify in this quadratic extension. There are three possibilities.

- It is possible that no such extension exists.
- $\mathfrak{a} = (d)$ is principal and the extension $F(\sqrt{d})$ is a quadratic extension ramified exactly at each $\mathfrak{p}_v$, and the generator d is unique up to multiplication by a unit in $\mathbb{Z}_F^{\times 2}$.
- If $\mathfrak{a}$ is not principal, it is possible there exists a fractional ideal $\mathfrak{b}$ such that $\mathfrak{a}\mathfrak{b}^2 = (d)$. In this case, we can again construct the extension $F(\sqrt{d})$ with the prescribed ramification, but d is only unique up to the the ideal $\mathfrak{b}$ used to construct it.

Let us consider the last case more closely. Let Cl_F denote the class group of F and for a fractional ideal $\mathfrak{c}$ let $[\mathfrak{c}]$ denote its ideal class in Cl_F. The equation $\mathfrak{a}\mathfrak{b}^2 = (d)$

means that $[\mathfrak{a}] = [\mathfrak{b}^{-2}]$ so that $[\mathfrak{a}] \in \mathrm{Cl}_F^2$. Moreover, if we take an element $[\mathfrak{c}]$ of order 2 in Cl_F, then

$$[\mathfrak{a}\mathfrak{b}^2] = [\mathfrak{a}\mathfrak{b}^2][\mathfrak{c}^2] = [\mathfrak{a}(\mathfrak{b}\mathfrak{c})^2]. \tag{5.1.2}$$

Thus, in the case where $\mathfrak{a}$ is not principal, but there exists $\mathfrak{b}$ with $\mathfrak{a}\mathfrak{b}^2$ principal, we have $[\mathfrak{a}] \in \mathrm{Cl}_F^2$ and $[\mathfrak{b}]$ is unique up to multiplication by $[\mathfrak{c}] \in \mathrm{Cl}_F[2]$.

We can now formulate our precise goal. Given $\mathfrak{a}$ (encoding ramification data), find $\mathfrak{b}^2$ and d such that $\mathfrak{a}\mathfrak{b}^2 = (d)$. In the following sections we will rephrase this problem in the function field setting.

Section 5.2

Curves and algebraic function fields

In this section we summarize the setting in which the algorithms of this chapter are stated. There are many comprehensive resources on this topic such as [21, I.6], [43], and [34].

First, let K be a perfect field.

Definition 5.2.1. An algebraic function field in one variable over K is a field extension F over K of transcendence degree 1. That is, there exists $x \in F$ such that x is transcendental over K and $[F : K(x)]$ is finite.

Definition 5.2.2. We say the K is the constant field of F. The exact constant field of F is the algebraic closure of K in F.

Remark 5.2.3. In theory we can assume that K is the exact constant field of F, but in practice for K a number field or $\mathbb{F}_q$ we try to work with constant fields that are as simple as possible.

Example 5.2.4. Let X be an irreducible affine plane curve defined by the defining equation $f(x,y) = 0$ with $f \in K[x,y]$. Then the function field of X, denoted by $K(X)$, is defined to be the field of fractions of the coordinate ring $\frac{K[x,y]}{(f(x,y))}$ of X.

Definition 5.2.5. A place P of F is the maximal ideal of some discrete valuation ring $\mathcal{O}_P$ of F. We denote the valuation on F corresponding to P by ord_P. The set of places of F is denoted $\mathrm{Pl}(F)$. The degree of a place P, denoted $\deg(P)$, is the index $[\mathcal{O}_P/P : K]$ of the residue class field as an extension of K.

Definition 5.2.6. The divisor class group of F, denoted $\mathrm{Div}(F)$, is the free abelian group generated by the places of F. A divisor $D \in \mathrm{Div}(F)$ is represented by a formal sum of places $D = \sum_{P \in \mathrm{Pl}(F)} a_P P$ with $a_P \in \mathbb{Z}$ for all P and $a_P = 0$ for all but finitely many P. We define $\mathrm{ord}_P(D)$ to be the coefficient a_P in the representation of D.

Definition 5.2.7. The support of a divisor $D = \sum_{P \in \mathrm{Pl}(F)} a_P P$, denoted $\mathrm{supp}(D)$, is $\{P \in \mathrm{Pl}(F) : a_P \neq 0\}$. The degree of D is defined to be $\deg(D) := \sum_{P \in \mathrm{Pl}(F)} a_P \deg(P)$. The subgroup of $\mathrm{Div}(F)$ consisting of the set of degree zero divisors of F is denoted by $\mathrm{Div}^0(F)$.

Definition 5.2.8. The image of the map $\mathrm{div}\colon F^\times \to \mathrm{Div}(F)$ defined by

$$\mathrm{div}(f) = \sum_{P \in \mathrm{Pl}(F)} \mathrm{ord}_P(f) P \tag{5.2.9}$$

is the subgroup of principal divisors of F and denoted $\mathrm{Princ}(F)$. Two divisors $D_1, D_2 \in \mathrm{Div}(F)$ are linearly equivalent if $D_1 - D_2 \in \mathrm{Princ}(F)$.

Definition 5.2.10. The Picard group of F is defined by $\mathrm{Pic}(F) := \mathrm{Div}(F)/\mathrm{Princ}(F)$. The Jacobian of F is defined by $\mathrm{Pic}^0(F) := \mathrm{Div}^0(F)/\mathrm{Princ}(F)$.

Definition 5.2.11. There is a partial order on $\mathrm{Div}(F)$ defined by $D_1 \geq D_2$ if and only if $\mathrm{ord}_P(D_1) \geq \mathrm{ord}_P(D_2)$ for all $P \in \mathrm{Pl}(F)$. We say that $D \in \mathrm{Div}(F)$ is effective if $D \geq 0$.

Definition 5.2.12. The Riemann-Roch space of a divisor $D \in \mathrm{Div}(F)$ is defined by $\mathscr{L}(D) := \{f \in F : \mathrm{div}(f) + D \geq 0\} \cup \{0\}$.

Now that we have some of the basic definitions of algebraic function fields, we also need to introduce some terminology concerning extensions of algebraic function fields.

Definition 5.2.13. Let F, F' be algebraic function fields over constant fields K, K' respectively and suppose that $F \subseteq F'$ and $K \subseteq K'$. When these conditions are satisfied we say that F' is an algebraic function field extension of F.

Every place P' of F' lies above a unique place $P = F \cap P'$ of F. Every place P of F lies below finitely many places P' of F'. We denote a place P' above P by $P'|P$. When $P'|P$ we can view $\mathcal{O}_{P'}$ as a free $\mathcal{O}_P$-module of rank $[F' : F]$ and $\mathcal{O}_P = \mathcal{O}_{P'} \cap F$.

We now summarize the fundamental identity from algebraic number theory in the function field setting. Let F' over K' be an extension of F over K and let P' be a place of F' above $P \in \mathrm{Pl}(F)$. There is a unique positive integer denoted $e(P'|P)$ such that $\mathrm{ord}_{P'}(f) = e(P'|P)\,\mathrm{ord}_P(f)$ for all $f \in F$. The positive integer $e(P'|P)$ is called the ramification index of $P'|P$. The residue degree, denoted $f(P'|P)$ is defined to be the index $[\mathcal{O}_{P'}/P' : \mathcal{O}_P/P]$ which makes sense after embedding $\mathcal{O}_P/P$ into $\mathcal{O}_{P'}/P'$. The fundamental identity is then given by the equation

$$[F' : F] = \sum_{P'|P} e(P'|P) f(P'|P). \tag{5.2.14}$$

We now summarize some necessary facts about extending the field of constants of and algebraic function field.

Definition 5.2.15. An extension F' (with constants K') of F (with constants K) is a constant field extension if $F' = FK'$.

Constant field extensions are one way in which the function field setting differs from the number field setting. When $F' = FK'$ is a constant field extension of F over K, there are several observations to make. First, the relative degree over the rational function field does not change, that is, $[F : K(x)] = [F' : K'(x)]$ for all $x \in F \setminus K$. Second, no places of F ramify in F'. Lastly, define the conorm map by

$$\mathrm{con}_{F'|F}(P) := \sum_{P'|P} e(P'|P)P' \in \mathrm{Div}(F'). \tag{5.2.16}$$

The conorm map extends to a homomorphism on divisor, principal divisors, and hence on divisor classes. Since constant field extensions are unramified, the conorm map induces an injection $\mathrm{Pic}(F) \hookrightarrow \mathrm{Pic}(F')$.

We conclude this section by proving a lemma we will need later in this chapter.

Lemma 5.2.17. *Let $aF^{\times 2}$ be a nontrivial coset of $F^\times/F^{\times 2}$ and consider the extension $K := F(\sqrt{a})$. Then a prime P of F is ramified in K if and only if $\mathrm{ord}_P(a)$ is odd.*

Proof. Since a is not a square in F, the extension is quadratic. Suppose $\mathrm{ord}_P(a)$ is odd and let $\mathfrak{p}$ be a place above P in K. Then we have

$$2\,\mathrm{ord}_{\mathfrak{p}}(\sqrt{a}) = \mathrm{ord}_{\mathfrak{p}}(a) = e(\mathfrak{p}/P)\,\mathrm{ord}_P(a). \tag{5.2.18}$$

Since $\mathrm{ord}_P(a)$ is odd, (5.2.18) implies that 2 divides $e(\mathfrak{p}/P)$ so that P is ramified in

K. Moreover, this says that $e(\mathfrak{p}/P) = 2$.

For the converse we check ramification locally at the place P. Suppose $e := \mathrm{ord}_P(a)$ is even. Choose a uniformizer t at P and let $b = a/t^{(e/2)}$. Then $F(\sqrt{a}) = F(\sqrt{b})$, and $\mathrm{ord}_P(b) = 0$ implies that $F(\sqrt{b}) = F(\sqrt{a})$ is unramified at P.

For a more general proof of this in arbitrary Kummer extensions see [34, Proposition 10.3]. □

Section 5.3

Quadratic extensions of function fields

We now address two tasks concerning quadratic extensions of function fields that we need for the algorithms in Section 5.4.

The first task is the problem (analogous to the problem in Section 5.1) of finding a quadratic extension $F(\sqrt{f})/F$ with ramification (in the relative extension) prescribed by $R \in \mathrm{Div}(F)$. By Proposition 5.2.17, we can take all nonzero coefficients of R to have absolute value 1. As was the case for number fields, there are three possibilities.

First, it could be the case that no such extension exists in which case there is nothing to do. The other easy case occurs when R is a principal divisor so that $R = \mathrm{div}(f)$ for some $f \in F^\times$. In this case, the extension $F(\sqrt{f})$ has the desired ramification determined by R.

The last case occurs when R is not principal, but there exists $D \in \mathrm{Div}(F)$ with $R - 2D = \mathrm{div}(f)$ for some $f \in F$. By Proposition 5.2.17, the extension $F(\sqrt{f})/F$ will be ramified precisely at the places in the support of R. For $D \in \mathrm{Div}(F)$, let $[D]$ denote the class of D in $\mathrm{Pic}(F)$. Since $[R - 2D] = 0 \in \mathrm{Pic}^0(F)$, we have that

$R \in 2\operatorname{Pic}(F)$. Moreover, if we let $[T] \in \operatorname{Pic}^0(F)[2]$, then

$$[R - 2D] = [R - 2D] - [2T] = [R - 2(D + T)]. \tag{5.3.1}$$

Thus, in the case where $R - 2D$ is principal, we have $R \in 2\operatorname{Div}(F)$ and D is unique up to an order 2 element of $\operatorname{Pic}(F)$. The fact that we cannot determine D exactly requires us to compute $\operatorname{Pic}(F)$ to carry out the desired computations. This forces us to work over $\mathbb{F}_q$ where Picard group computations are implemented.

The other task is to determine when the quadratic extension $F(\sqrt{f})$ over F is Galois (as an absolute extension of $\mathbb{F}_q(x)$) given that F is Galois over $\mathbb{F}_q(x)$. Kummer theory tells us precisely when such an extension is Galois in the following Lemma.

Lemma 5.3.2. *Let F be a Galois extension of $\mathbb{F}_q(x)$ with Galois group G and let $f \in F^\times/F^{\times 2}$. Then the quadratic extension $F(\sqrt{f})$ is Galois as an absolute extension of $\mathbb{F}_q(x)$ if and only if $\sigma(f)/f$ is a square in F for every $\sigma \in G$.*

We can now formulate these concepts into an algorithm over $\mathbb{F}_q$.

Section 5.4

An algorithm over $\mathbb{F}_q$

Let F be function field with field of constants $\mathbb{F}_q$ with $q = p^r$ and $p \neq 2$. Let $\mathbb{F}_q(x)$ denote the rational function field in the variable x.

Definition 5.4.1. A tame Belyi map over $\mathbb{F}_q$ is a tame extension of function fields $\mathbb{F}_q(x) \hookrightarrow F$ with $[F : \mathbb{F}_q(x)]$ coprime to p unramified outside of all places above $\{0, 1, \infty\}$.

Definition 5.4.2. A 2-group Belyi map modulo q is a tame Belyi map over $\mathbb{F}_q$ with $[F : \mathbb{F}_q(x)]$ a power of 2.

Remark 5.4.3. The theory of tame Belyi maps is similar to the theory in characteristic zero.

We now describe the algorithms to iteratively compute 2-group Belyi maps modulo q. The basic idea is to compute a tower of quadratic extensions by extracting square roots to work our way up the tower. Since we are concerned with Galois Belyi maps, we want to make sure that each intermediate field is Galois as an absolute extension of $\mathbb{F}_q(x)$. To start, we describe the degree 2 Belyi maps.

Lemma 5.4.4. *The three degree* 2 *Belyi maps modulo q up to isomorphism are*

$$F_{(1,2,2)} = \frac{\mathbb{F}_q(x)[y]}{(y^2 + x - 1)}, \; F_{(2,1,2)} = \frac{\mathbb{F}_q(x)[y]}{(y^2 - x)}, \; \textit{and } F_{(2,2,1)} = \frac{\mathbb{F}_q(x)[y]}{(y^2 - x^2 + x)}. \tag{5.4.5}$$

Proof. All three degree 2 passports $(0, \mathbb{Z}/2\mathbb{Z}, (1, 2, 2))$, $(0, \mathbb{Z}/2\mathbb{Z}, (2, 1, 2))$, and $(0, \mathbb{Z}/2\mathbb{Z}, (2, 2, 1))$ have size 1 by Theorem 3.4.1 and Theorem 3.4.3. Since each field is a 2-group Belyi map with passport specified by its subscript with no two isomorphic, this is an exhaustive list. $\square$

Remark 5.4.6. All three 2-group Belyi maps in Lemma 5.4.4 are lax isomorphic.

Next, we discuss the algorithms to test when a quadratic extension is Galois (over the rational function field).

Algorithm 5.4.7 (IsGalois)**.**

Input:

- F a Galois extension of $\mathbb{F}_q(x)$

- $\mathrm{Gal}(F \mid \mathbb{F}_q(x))$ explicitly given as automorphisms of F
- $f \in F$

Output: True if the quadratic extension $F(\sqrt{f})$ of F is a Galois extension over $\mathbb{F}_q(x)$ and False otherwise

1. For each generator σ of $\mathrm{Gal}(F \mid \mathbb{F}_q(x))$ test if $\sigma(f)/f$ is a square in $F^\times$.
2. If $\sigma(f)/f \in F^{\times 2}$ for all generators σ, then return True otherwise return False.

Proof of correctness. The correctness of this algorithm follows from Kummer theory as discussed in Lemma 5.3.2. It suffices to test on generators since the property of being a square is multiplicative. □

Algorithm 5.4.8 (IsGaloisOverExtension)**.**

Input:

- F a Galois extension of $\mathbb{F}_q(x)$
- $\mathrm{Gal}(F \mid \mathbb{F}_q(x))$ explicitly given as automorphisms of F
- $f \in F$

Let F' be the function field F with the constant field extended from $\mathbb{F}_q$ to $\mathbb{F}_{q^2}$.
Output: True if the quadratic extension $F(\sqrt{f})$ of F is a Galois extension over $\mathbb{F}_{q^m}(x)$ after extending the field of constants from $\mathbb{F}_q$ to $\mathbb{F}_{q^m}$ (for some positive integer m) and False otherwise

1. For each generator σ of $\mathrm{Gal}(F' \mid \mathbb{F}_{q^2}(x))$ test if $\sigma(f)/f$ is a square in F'.
2. If $\sigma(f)/f$ is a square in F' for all generators σ, then return True otherwise return False.

Proof of correctness. The proof is similar to the previous algorithm. The proof that it is sufficient to check if elements are square over $\mathbb{F}_{q^2}$ can be found in [43, Corollary 3.7.4]. $\square$

The next algorithm details the process of finding the appropriate candidate function to obtain a quadratic extension by extracting a square root.

Algorithm 5.4.9 (GetCandidateFunctions)**.**

Input:

- F a 2-group Belyi map modulo q of degree $d = 2^m$ corresponding to a 2-group permutation triple σ
- A passport $\mathcal{P} = (\widetilde{G}, (a, b, c))$ with $\widetilde{G}$ a 2-group of order $2d$ such that there exists a 2-group permutation triple $\widetilde{\sigma}$ with passport $\mathcal{P}$ that is a lift of σ
- $\operatorname{Gal}(F \,|\, \mathbb{F}_q(x)) \cong \langle \sigma \rangle$ explicitly given as automorphisms of F

Output: A list of candidate functions $\{f_i\}$ with each $f_i \in F$ such that $F(\sqrt{f_i})$ is a 2-group Belyi map modulo q with passport $\mathcal{P}$.

1. For $s \in \{0, 1, \infty\}$ compute

$$r_s := \begin{cases} 0 & \text{if } \operatorname{order}(\sigma_s) = \operatorname{order}(\widetilde{\sigma}_s) \\ 1 & \text{if } \operatorname{order}(\sigma_s) < \operatorname{order}(\widetilde{\sigma}_s) \end{cases} \tag{5.4.10}$$

2. Compute

$$R := \sum_{s \in \{0,1,\infty\}} r_s R_s \in \operatorname{Div}(F) \tag{5.4.11}$$

where R_0, R_1, R_∞ are defined to be the supports of $\mathrm{div}(x)$, $\mathrm{div}(x-1)$, and $\mathrm{div}(1/x)$ respectively.

3. Compute the abelian group $\mathrm{Pic}(F) = T \oplus \mathbb{Z}$ (with T a finite abelian group) along with a map $\psi\colon \mathrm{Div}(F) \to \mathrm{Pic}(F)$.

4. Compute $[R] := \psi(R)$.

5. Check that $[R] \in 2\,\mathrm{Pic}(F)$. If not, then return the empty set, otherwise continue.

6. For each $a \in \mathrm{Pic}(F)[2]$ compute the following:

   (a) Let $D_a := \psi^{-1}(a + [R]/2) \in \mathrm{Div}(F)$.

   (b) Compute $\mathscr{L}(R - 2D_a)$.

   (c) If $\mathscr{L}(R-2D_a)$ has dimension 1, then compute $f_a \in F$ with $\mathrm{div}(f_a)$ generating $\mathscr{L}(R-2D_a)$ and go to Step 6d. Otherwise go to the next $a \in \mathrm{Pic}(F)[2]$.

   (d) Apply Algorithm 5.4.7 to F, $\mathrm{Gal}(F \,|\, \mathbb{F}_q(x))$, and f_a from Step 6c to see if $F(\sqrt{f_a})$ generates a Galois extension. If $F(\sqrt{f_a})$ is Galois over $\mathbb{F}_q(x)$ then save f_a and go to the next $a \in \mathrm{Pic}(F)[2]$. If $F(\sqrt{f_a})$ is not Galois over $\mathbb{F}_q(x)$, then go to Step 6e.

   (e) Let F' be the function field F after extending the field of constants $\mathbb{F}_q$ to $\mathbb{F}_{q^2}$. Apply Algorithm 5.4.8 to F', $\mathrm{Gal}(F' \,|\, \mathbb{F}_{q^2}(x))$, and f_a (viewed as an element of F') from Step 6c to see if $F'(\sqrt{f_a})$ generates a Galois extension. If $F(\sqrt{f_a})$ is Galois over $\mathbb{F}_{q^2}(x)$ then save f_a. Go to the next $a \in \mathrm{Pic}(F)[2]$.

7. Let S be the set of f_a saved in Step 6d. Let S' be the set of f_a saved in Step 6e.

8.
   - If S is nonempty, then for each $f_a \in S$ compute $F(\sqrt{f_a})$,

     $$G_a \cong \mathrm{Gal}(F(\sqrt{f_a}) \,|\, \mathbb{F}_q(x)),$$

     and let $S'' = \{f_a \in S : G_a \cong \widetilde{G}\}$.

   - If S is empty, then for each $f_a \in S'$ compute $F'(\sqrt{f_a})$,

     $$G_a \cong \mathrm{Gal}(F'(\sqrt{f_a}) \,|\, \mathbb{F}_{q^2}(x)),$$

     and let $S'' = \{f_a \in S' : G_a \cong \widetilde{G}\}$.

9. Return the list S'' from Step 8.

Proof of correctness. First, note that since we enumerated the isomorphism classes of 2-group Belyi maps in Chapter 3, we know the size of each passport $\mathcal{P}$ as input to this algorithm. The divisor R computed in Step 2 encodes the ramification required to obtain a 2-group Belyi map with ramification matching the passport $\mathcal{P}$. From the discussion in Section 5.3, $R \in 2\,\mathrm{Div}(F)$, and we can find all solutions to the equation

$$[R - 2D] = [0] \tag{5.4.12}$$

in $\mathrm{Pic}(F)$. For every element $a \in \mathrm{Pic}(F)[2]$ we get a solution to (5.4.12). More precisely, the divisor D_a computed in Step 6a satisfies $[R-2D_a] = [0]$, and all solutions to (5.4.12) are of the form D_a for some $a \in \mathrm{Pic}(F)[2]$. Now, since $R - 2D_a$ is principal for each a, we can find a candidate function $f_a \in F$ with $\mathrm{div}(f_a) = R - 2D_a$. After collecting the candidate functions f_a, we first use Algorithm 5.4.7 and Algorithm 5.4.8 to eliminate f_a that do not generate Galois extensions. Lastly, in Step 8, we only keep candidate functions f_a that generate extensions with Galois group isomorphic to the group $\widetilde{G}$ specified by the passport $\mathcal{P}$. Algorithm 5.4.7 and Algorithm 5.4.8 guarantee that that no further constant field extension is required. $\square$

The next algorithm details the process of extracting a square root of a candidate function (obtained from the output of Algorithm 5.4.9) and lifting automorphisms.

Algorithm 5.4.13 (LiftBelyiMap)**.**

Input:

- The same input as in Algorithm 5.4.9
- Additionally, a specific f_a from the output of Algorithm 5.4.9

Output: A 2-group Belyi map modulo q with passport $\mathcal{P}$ and explicit automorphisms identified with its Galois group $\widetilde{G}$

1. Compute $m_{f_a,\mathbb{F}_q(x)} \in \mathbb{F}_q(x)[y]$ the minimal polynomial of f_a over $\mathbb{F}_q(x)$ and let α be a root of $m_{f_a,\mathbb{F}_q(x)}(y^2)$. Let $\widetilde{F}$ denote the extension $\mathbb{F}_q(x)(\alpha)$.
2. Let $m_{\alpha,\mathbb{F}_q(x)}$ be the minimal polynomial of α over $\mathbb{F}_q(x)$ and compute the set
$$R := \{r : r \text{ is a root of } m_{\alpha,\mathbb{F}_q(x)} \text{ in } \widetilde{F}\}. \tag{5.4.14}$$
3. Return the following:
   - The absolute extension $\widetilde{F}$ of $\mathbb{F}_q(x)$
   - The set of field automorphisms $\{\tau_r : r \in R\}$ where $\tau_r \colon \widetilde{F} \to \widetilde{F}$ is defined by $\alpha \mapsto r$.

Proof of correctness. Since f_a is obtained from the output of Algorithm 5.4.9, the extension $\widetilde{F}$ is Galois so that $m_{\alpha,\mathbb{F}_q(x)}$ has exactly $\deg(\widetilde{F})$ roots in $\widetilde{F}$. Again by Algorithm 5.4.9, the extension $\widetilde{F}$ defines a 2-group Belyi map modulo q with passport $\mathcal{P}$. The maps $\tau_r\colon \alpha \mapsto r$ define $\deg(\widetilde{F})$ automorphisms of $\widetilde{F}$ over $\mathbb{F}_q(x)$. □

With Algorithm 5.4.9 and Algorithm 5.4.13 at our disposal, we can now explain how to compute *all* 2-group Belyi maps with a given passport.

Algorithm 5.4.15 (ComputePassport)**.**

Input:

- A passport $\mathcal{P}_{\text{above}} = (\widetilde{G}, (a, b, c))$
- A list of passports $\mathcal{P}_1, \dots, \mathcal{P}_k$
- For each $\mathcal{P}_i$ a list of triples of data $(\sigma_i^1, F_i^1, G_i^1), \dots (\sigma_i^{\#\mathcal{P}_i}, F_i^{\#\mathcal{P}_i}, G_i^{\#\mathcal{P}_i})$ with the F_i^j pairwise non-isomorphic and each $(\sigma_i^j, F_i^j, G_i^j)$ satisfying the following:
  - σ_i^j is a 2-group permutation triple with passport $\mathcal{P}_i$
  - There exists a 2-group permutation triple $\widetilde{\sigma}_i^j$ with passport $\mathcal{P}_{\text{above}}$ that is a lift of σ_i^j
  - F_i^j is a 2-group Belyi map modulo q
  - G_i^j is the Galois group of F_i^j over $\mathbb{F}_q(x)$ explicitly given as automorphisms of F_i^j

Output: A list of triples of data $(\widetilde{F}^1, \widetilde{G}^1), \dots, (\widetilde{F}^{\#\mathcal{P}_{\text{above}}}, \widetilde{G}^{\#\mathcal{P}_{\text{above}}})$ with $\widetilde{F}^j$ a 2-group Belyi map modulo q' (with q' a power of q), $\widetilde{G}^j$ the Galois group of F_i^j explicitly given as automorphisms of $\widetilde{F}^j$, and the $\widetilde{F}^j$ pairwise non-isomorphic.

1. Apply Algorithm 5.4.9 to every triple of data $(\sigma_i^j, F_i^j, G_i^j)$ downstairs (along with the passport $\mathcal{P}_{\text{above}}$) to obtain a list of candidate functions $\mathrm{CFS}_q := \{f_i^{j,k}\}$ with $f_i^{j,k} \in F_i^j$ for each k.

2. For each $f_i^{j,k} \in \mathrm{CFS}_q$, apply Algorithm 5.4.13 with input $(\sigma_i^j, F_i^j, G_i^j)$ and $f_i^{j,k}$ to obtain $\widetilde{F_i^{j,k}}$ a 2-group Belyi map modulo q with passport $\mathcal{P}$ and Galois group $\widetilde{G_i^{j,k}}$. Let BELYI_q denote the list of all pairs $(\widetilde{F_i^{j,k}}, \widetilde{G_i^{j,k}})$ obtained in this step.

3. Test isomorphism of fields $\widetilde{F_i^{j,k}}$ and $\widetilde{F_i^{j,k'}}$ over $\mathbb{F}_q(x)$ for each pair of fields in BELYI_q. Keep exactly one representative of each isomorphism class from BELYI_q and store this data (including the Galois group) in $\mathrm{BELYIISO}_q$.

4. If $\#\mathrm{BELYIISO}_q = \#\mathcal{P}_{\text{above}}$ then return $\mathrm{BELYIISO}_q$. Otherwise, extend the constant field from $\mathbb{F}_q$ to $\mathbb{F}_{q^2}$ and repeat Steps 1, 2, and 3 to obtain lists CFS_{q^2}, BELYI_{q^2}, and $\mathrm{BELYIISO}_{q^2}$. Continue this process for $q, q^2, q^3, \ldots$ and return $\mathrm{BELYIISO}_{q^m}$ for the first $m \in \{1, 2, \ldots\}$ with the same cardinality as $\mathcal{P}_{\text{above}}$.

Proof of correctness. This algorithm is largely bookkeeping and applying Algorithm 5.4.9 and Algorithm 5.4.13. The triples of data downstairs enumerate the isomorphism classes of 2-group Belyi maps modulo q in all passports that have a representative with a lift that has passport $\mathcal{P}_{\text{above}}$.

It is important to note at this point that for every downstairs passport $\mathcal{P}_i$, we need *all* $\#\mathcal{P}_i$ triples of data $(\sigma_i^j, F_i^j, G_i^j)$. This is because Algorithm 5.4.9 only identifies candidate functions that produce 2-group Belyi maps with the correct passport. It does not provide a way to identify the precise isomorphism class. That is why, in this algorithm, we must be content with a list of pairwise non-isomorphic Belyi maps. By testing isomorphisms we can ensure that we have a representative from every isomorphism class, but we cannot identify the permutation triple corresponding to a Belyi map. In this algorithm the permutation triples (obtained from the algorithms in Section 3.3) are simply a bookkeeping tool.

The one subtle point is explaining how to obtain the q'. After each round of computing the lists CFS_{q^i}, BELYI_{q^i}, and $\mathrm{BELYIISO}_{q^i}$ it is possible that we failed to find all candidate functions over the constant field $\mathbb{F}_{q^i}$. The enumeration of isomorphism classes in Section 3.3 ensures that this process of extending the constant field will terminate, but does not provide an a priori bound on the q' required. $\square$

Applying Algorithm 5.4.15 to every degree d passport allows us to enumerate all 2-group Belyi maps modulo q *one degree at a time*. Section 5.6 details how far we were able to push these computations in practice using [30].

Remark 5.4.16. The algorithms in this section rely on the Magma implementations to compute class groups $\mathrm{Pic}(F)$ for global function fields, and the implementations to compute Riemann-Roch spaces $\mathscr{L}(D)$.

We conclude this section with an example of how these algorithms are applied in a specific example.

Example 5.4.17. In this example we use the algorithms in this section to compute all three 2-group Belyi maps modulo 3 with passport $(3, G, (4, 4, 4))$ where $G = (\mathbb{Z}/4\mathbb{Z}) : (\mathbb{Z}/4\mathbb{Z})$ is the Galois group described at the following link.

http://www.lmfdb.org/GaloisGroup/16T8

The Magma script can be run using the following command from the repository [30].

Shell

```
magma thesis_examples/compute_passport_16T8_444_g3.m
```

The source code in this file is as follows.

Magma

```
load "config.m";
SetVerbose("TwoDBPassport", 3);
SetVerbose("TwoDB", 1);
objs := GetPassportObjects(16);
s := objs[#objs-2];
ComputeBelyiMaps(s : optimized := false);
```

This example has 7 size 1 passports $\mathcal{P}_i$ as in Algorithm 5.4.15. Each isomorphism class downstairs yields a candidate Belyi map upstairs, and the isomorphism checking in Algorithm 5.4.15 Step 3 correctly identifies the 3 distinct isomorphism classes out of 7.

Section 5.5

An implementation over $\mathbb{Q}^{\text{al}}$

We now discuss the situation in characteristic zero. The procedure has the same broad strokes as that in characteristic $p \neq 2$, but there is a key difference in the technique to get candidate functions to extract a square root of in Algorithm 5.4.9.

In characteristic zero there is no implementation to compute $\operatorname{Pic}(F)$ (for general F). To show how we can get around this (in some cases), we now rewrite Algorithm 5.4.9 in the characteristic zero setting.

Algorithm 5.5.1.

Input:

- $K(x) \hookrightarrow F := K(X)$ a 2-group Belyi map of degree $d = 2^m$ corresponding to a

2-group permutation triple σ

- A passport $\mathcal{P} = (\widetilde{G}, (a,b,c))$ with $\widetilde{G}$ a 2-group of order $2d$ such that there exists a 2-group permutation triple $\widetilde{\sigma}$ with passport $\mathcal{P}$ that is a lift of σ

- $\mathrm{Gal}(F \,|\, K(x)) \cong \langle \sigma \rangle$ explicitly given as automorphisms of F

Output: A list of candidate functions $\{f_i\}$ with each $f_i \in F$ such that $K(x) \hookrightarrow F(\sqrt{f_i})$ is a 2-group Belyi map with passport $\mathcal{P}$.

1. For $s \in \{0, 1, \infty\}$ compute

$$r_s := \begin{cases} 0 & \text{if } \operatorname{order}(\sigma_s) = \operatorname{order}(\widetilde{\sigma}_s) \\ 1 & \text{if } \operatorname{order}(\sigma_s) < \operatorname{order}(\widetilde{\sigma}_s) \end{cases} \tag{5.5.2}$$

2. Compute

$$R := \sum_{s \in \{0,1,\infty\}} r_s R_s \in \operatorname{Div}(F) \tag{5.5.3}$$

where R_0, R_1, R_∞ are defined to be the supports of $\operatorname{div}(x)$, $\operatorname{div}(x-1)$, and $\operatorname{div}(1/x)$ respectively.

3. Let M denote the set $(R + 2\mathbb{Z}R) \cap \operatorname{Div}^0(F)$ and for $B \in \mathbb{Z}_{\geq 1}$ let

$$M_B = \Big\{ R + 2nR : n \in \{-B, -B+1, \ldots, B-1, B\} \Big\} \cap \operatorname{Div}^0(F).$$

4. For each $D \in M$ compute the following:

   (a) Compute $\mathscr{L}(D)$.

(b) If $\mathscr{L}(D)$ has dimension 1, then compute $f_D \in F$ with $\mathrm{div}(f_D)$ generating $\mathscr{L}(D)$ and go to the next step. Otherwise, go to the next $D \in M$.

(c) Check to see if $F(\sqrt{f_D})$ is Galois. If $F(\sqrt{f_D})$ is Galois, then save f_D. and go to the next $D \in M$. If $F(\sqrt{f_D})$ is not Galois, then go to the next Step.

(d) Let F' be the function field F after extending the field of constants to the compositum of the residue fields of all places in the support of D. Check to see if $F'(\sqrt{f_D})$ is Galois. If $F'(\sqrt{f_D})$ is Galois, then save f_D. Go to the next $D \in M$.

5. Let S be the set of f_D saved in Step 4c. Let S' be the set of f_a saved in Step 4d.

6.
   - If S is nonempty, then for each $f_D \in S$ compute $F(\sqrt{f_D})$,

$$G_D \cong \mathrm{Gal}(F(\sqrt{f_a}) \,|\, K(x)),$$

     and let $S'' = \{f_D \in S : G_D \cong \widetilde{G}\}$.

   - If S is empty, then for each $f_D \in S'$ compute $F'(\sqrt{f_D})$,

$$G_D \cong \mathrm{Gal}(F'(\sqrt{f_D}) \,|\, K(x)),$$

     and let $S'' = \{f_D \in S' : G_D \cong \widetilde{G}\}$.

7. Return the list S'' from Step 6.

Although Algorithm 5.5.1 in characteristic zero resembles Algorithm 5.4.9 over $\mathbb{F}_q$, the characteristic zero algorithm is unfortunately not guaranteed to find any candidate functions! This is due to the fact that we are not computing representatives of $\mathrm{Pic}^0(F)[2]$.

Without enumerating representatives of $\mathrm{Pic}^0(F)[2]$, the approach in characteristic zero will always be ad hoc in the sense that in Step 3 we are blindly looking at all combinations of points that yield the desired ramification. This process is guaranteed to succeed when F has class number 1, but this condition is not often satisfied (in fact there are only 8 such non-rational function fields over $\mathbb{F}_q$).

This ad hoc approach does, however, allow us to compute some 2-group Belyi maps in characteristic zero. We conclude this section by describing the results of these computations along with those in positive characteristic.

Section 5.6

Results of computations

In this section we summarize the computations carried out in [30] based on the algorithms discussed in Section 5.4 and Section 5.5.

In characteristic 3 we were able to compute all 2-group Belyi maps modulo 3 up to degree 32. The results of these computations can be accessed in Magma (with working directory the repository [30]) using the following code.

Magma

```
load "config.m";
d := 16;
objs := [ReadTwoDBPassport(f) : f in PassportFilenames(d)];
```

The information for a given passport can be accessed using the following code.

Magma

```
s := Random(objs);
FunctionFields(s);
BelyiMaps(s);
FunctionFieldAutomorphisms(s);
```

In addition to the systematic computation of 2-group Belyi maps modulo 3, we were also able to apply the implementation in Section 5.5 to compute hundreds of 2-group Belyi maps in characteristic zero with degrees up to 256.

We conclude this chapter with interesting examples encountered during these computations. First, in Section 5.7 we set up some notation to help with the description of these examples.

Section 5.7

Naming conventions for database examples

In this section we explain the conventions used for filenames in [30]. This will be useful for the rest of this chapter when referring to specific examples.

The first point to explain is how we deal with isomorphism classes of 2-group Belyi maps as opposed to passports. As discussed in Chapter 3, isomorphism classes of 2-group Belyi maps correspond to 2-group permutation triples each with a unique filename of the form

$$\texttt{DNG-a,b,c-gE-H} \tag{5.7.1}$$

where

$$\begin{aligned}
&\texttt{D}: \text{ degree in } \{2,4,8,16,32,64,128,256\}\\
&\texttt{N}: \text{ either } \texttt{T} \text{ or } \texttt{S} \text{ identifying group database}\\
&\texttt{G}: \text{ a positive integer identifying the group}\\
&\texttt{a}: \text{ ramification index of } 0 \text{ in } \{2,4,8,16,32,64,128,256\}\\
&\texttt{b}: \text{ ramification index of } 1 \text{ in } \{2,4,8,16,32,64,128,256\}\\
&\texttt{c}: \text{ ramification index of } \infty \text{ in } \{2,4,8,16,32,64,128,256\}\\
&\texttt{g}: \text{ just the letter } \texttt{g}\\
&\texttt{E}: \text{ the genus in } \mathbb{Z}_{\geq 0}\\
&\texttt{H}: \text{ the hash of the 2-group permutation triple a positive integer}
\end{aligned} \tag{5.7.2}$$

For example, the filename `16T12-4,8,2-g2-1396531181` corresponds to a degree 64 Belyi map with monodromy group identified in the transitive group database (`T` for transitive)

http://magma.maths.usyd.edu.au/magma/handbook/text/753

by `16T12`. The `4,8,2` encodes the ramification above $0, 1, \infty$ respectively, the `g2` indicates that this Belyi map has genus 2, and the `1396531181` is the hash of the permutation triple corresponding to this Belyi map.

Another example, `64S7-8,8,4-g17-1653847134` corresponds to a degree 16 Belyi map with monodromy group identified in the small group database (`S` for small)

http://magma.maths.usyd.edu.au/magma/handbook/text/748

by `64S7`. The `8,8,4` encodes the ramification above $0, 1, \infty$ respectively, the `g17` indicates that this Belyi map has genus 17, and the `1653847134` is the hash of the permutation triple corresponding to this Belyi map. In this example, the hash is necessary to distinguish this example from

`64S7-8,8,4-g17-2483683244`,

`64S7-8,8,4-g17-623082418`,

and `64S7-8,8,4-g17-964508325`.

A passport is given a similar filename. The only difference for a passport is that the hash is no longer required. For example, the Belyi maps with filenames in the previous paragraph all have the same passport `64S7-8,8,4-g17`. The hash is to distinguish between isomorphism classes within a passport.

The reason for this distinction is that the algorithms for computing equations in this chapter are designed for passports. Now that we have a concise way of talking about example, we use this in the rest of the chapter to discuss examples.

Section 5.8 Degree 2

In degree 2, there are 3 isomorphism classes of 2-group Belyi maps. In characteristic zero they are represented by

$$\frac{\mathbb{Q}(x)[y]}{(y^2+x-1)},\ \frac{\mathbb{Q}(x)[y]}{(y^2-x)},\ \text{and}\ \frac{\mathbb{Q}(x)[y]}{(y^2-x^2+x)} \tag{5.8.1}$$

and each is the unique Belyi map with passport `2T1-1,2,2-g0`, `2T1-2,1,2-g0`, and `2T1-2,2,1-g0`, respectively.

Section 5.9

Degree 4

In degree 4 there are 7 isomorphism classes of 2-group Belyi maps and each has passport size 1.

All function fields are of the form $\mathbb{Q}(x)[y]/(f(x,y))$ with f one of the polynomials in (5.9.1). The subscripts indicate the passport.

$$
\begin{aligned}
f_{\texttt{4T1-1,4,4-g0}} &= y^4 + x - 1 \\
f_{\texttt{4T1-2,4,4-g1}} &= y^4 + x^3 - x^2 \\
f_{\texttt{4T1-4,1,4-g0}} &= y^4 - x \\
f_{\texttt{4T1-4,2,4-g1}} &= y^4 - x^3 + 2x^2 - x \\
f_{\texttt{4T1-4,4,1-g0}} &= y^4 - x^4 + x^3 \\
f_{\texttt{4T1-4,4,2-g1}} &= y^4 - x^2 + x \\
f_{\texttt{4T2-2,2,2-g0}} &= y^4 + (4x-2)y^2 + 1
\end{aligned}
\tag{5.9.1}
$$

Although every function field has constant field $\mathbb{Q}$, lifting the automorphisms requires extending the constant field to $\mathbb{Q}(\zeta_4)$ in all examples except for the Belyi map defined by $f_{\texttt{4T2-2,2,2-g0}}$.

Use the following code from [30] to obtain a list of these function fields in Magma.

Magma

```
load "config.m";
objs := [ReadTwoDBPassportChar0(f) : f in PassportFilenames(4)];
fields := [FunctionFields(s)[1] : s in objs];
```

Section 5.10

Degree 8

In degree 8 there are 13 isomorphism classes of 2-group Belyi maps that have size 1 passports and there are 3 size 2 passports.

All function fields are of the form $\mathbb{Q}(x)[y]/(f(x,y))$ with f one of the polynomials in (5.10.1), (5.10.2), (5.10.5), (5.10.3), (5.10.4), for size 1 passports, or (5.10.6), for size 2 passports. The subscripts indicate the passport.

$$\begin{aligned} f_{\texttt{8T1-1,8,8-g0}} &= y^8 + x - 1 \\ f_{\texttt{8T1-8,1,8-g0}} &= y^8 - x \\ f_{\texttt{8T1-8,8,1-g0}} &= y^8 - x^8 + x^7 \end{aligned} \tag{5.10.1}$$

$$\begin{aligned} f_{\texttt{8T1-2,8,8-g2}} &= y^8 + x^5 - x^4 \\ f_{\texttt{8T1-8,2,8-g2}} &= y^8 - x^5 + 4x^4 - 6x^3 + 4x^2 - x \\ f_{\texttt{8T1-8,8,2-g2}} &= y^8 - x^4 + 3x^3 - 3x^2 + x \end{aligned} \tag{5.10.2}$$

$$\begin{aligned} f_{\texttt{8T4-2,2,4-g0}} &= y^8 + (4x-2)y^4 + 1 \\ f_{\texttt{8T4-2,4,2-g0}} &= y^8 + (8x^4 - 16x^3 + 16x - 8)y^4 + 16x^8 - 128x^7 + 448x^6 \\ &\quad - 896x^5 + 1120x^4 - 896x^3 + 448x^2 - 128x + 16 \\ f_{\texttt{8T4-4,2,2-g0}} &= y^8 + (-8x^4 + 16x^3)y^4 + 16x^8 \end{aligned} \tag{5.10.3}$$

$$f_{\texttt{8T5-4,4,4-g2}} = y^8 + (1/2x^3 - 3/2x^2 + x)y^4 + 1/16x^6 - 1/8x^5 + 1/16x^4 \tag{5.10.4}$$

$$\begin{aligned}
f_{\texttt{8T2-2,4,4-g1}} &= y^8 + (8x-4)y^6 + (4x-4)y^5 + (16x^2 - 31/2x + 11/2)y^4 \\
&\quad + (-8x+8)y^3 + (2x^2+x+1)y^2 + (-15x^2+18x-3)y \\
&\quad + 4x^3 - 127/16x^2 + 35/8x + 9/16 \\
f_{\texttt{8T2-4,2,4-g1}} &= y^8 + (1/16x^3 - 1/16x^2 + 1/128x)y^4 + 1/65536x^2 \\
f_{\texttt{8T2-4,4,2-g1}} &= y^8 + (8x-4)y^6 + (-1/2x^4 - 23/2x^3 + 28x^2 - 16x + 6)y^4 \\
&\quad + (6x^5 - 25x^4 + 27x^3 - 8x^2 + 8x - 4)y^2 + 1/16x^8 - 9/8x^7 \\
&\quad + 97/16x^6 - 9x^5 + 7/2x^4 + 9/2x^3 - 4x^2 + 1
\end{aligned} \tag{5.10.5}$$

$$\begin{aligned}
f_{\texttt{8T1-4,8,8-g3}} &= y^8 + x^5 - 3x^4 + 3x^3 - x^2 \\
f_{\texttt{8T1-8,4,8-g3}} &= y^8 - x^5 + 2x^4 - x^3 \\
f_{\texttt{8T1-8,8,4-g3}} &= y^8 - x^2 + x
\end{aligned} \tag{5.10.6}$$

Although every function field has constant field $\mathbb{Q}$, lifting the automorphisms requires extending the constant field to $\mathbb{Q}(\zeta_8)$ in all examples except for the those that are covers of the Belyi map defined by $f_{\texttt{4T2-2,2,2-g0}}$ in Section 5.9.

Use the following code from [30] to obtain a list of these function fields in Magma.

Magma

```
load "config.m";
objs := [ReadTwoDBPassportChar0(f) : f in PassportFilenames(8)];
fields := [FunctionFields(s)[1] : s in objs];
```

Perhaps the most interesting note to make in degree 8 is that it is the lowest degree where the characteristic zero approach appears to fail. Indeed, the genus 3 passports represented in (5.10.6) have size 2, but only a single representative Belyi map for each. In these examples, all candidate functions obtained from the adhoc approach in Algorithm 5.5.1 yield isomorphic function fields over $\mathbb{Q}(x)$, so we know we are missing a Belyi map in each one of these passports.

The techniques in Section 5.4, however, do manage to succeed for these size 2 passports. Working over $\mathbb{F}_3$, using Algorithm 5.4.15, we obtain the function fields

$$\frac{\mathbb{F}_3(x)[y]}{(y^8+2x^6+2x^5+2x^4+x^3+x^2+x)} \quad \text{and} \quad \frac{\mathbb{F}_3(x)[y]}{(y^8+x^6+2x^3)} \tag{5.10.7}$$

for the passport `8T1-8,8,4-g3`, and one can check that the fields in (5.10.7) are not isomorphic.

To obtain a list of degree 8 function fields corresponding to 2-group Belyi maps in characteristic 3, use the following code from [30].

Magma

```
load "config.m";
objs := [ReadTwoDBPassport(f) : f in PassportFilenames(8)];
fields := [* *];
for s in objs do
  fields cat:= FunctionFields(s);
end for;
```

Section 5.11

Degree 16

In degree 16 there are 55 isomorphism classes of 2-group Belyi maps with 41 distinct passports. All passports are of size 1 except for Passport `16T8-4,4,4-g3` which has size 3, Passports `16T1-4,16,16-g6`, `16T1-16,4,16-g6`, `16T1-16,16,4-g6` which have size 2, and Passports `16T1-8,16,16-g7`, `16T1-16,8,16-g7`, `16T1-16,16,8-g7` which have size 4.

Algorithm 5.4.15 succeeds in characteristic 3, but the equations are too large to write here. For an illustrative example of Algorithm 5.4.15 in action see Example 5.4.17. To obtain a list of degree 16 function fields corresponding to 2-group Belyi maps in characteristic 3, use the following code from [30].

Magma

```
load "config.m";
objs := [ReadTwoDBPassport(f) : f in PassportFilenames(16)];
fields := [* *];
for s in objs do
  fields cat:= FunctionFields(s);
end for;
```

Below are some of the characteristic zero function fields corresponding to 2-group Belyi maps computed using Algorithm 5.5.1. At this point the equations tend to be too big to fit on a page.

$$f_{\texttt{16T1-1,16,16-g0}} = y^{16} + x - 1 \tag{5.11.1}$$

$$f_{\texttt{16T13-2,2,8-g0}} = y^{16} + (4x - 2)y^8 + 1 \tag{5.11.2}$$

$$f_{\texttt{16T1-8,16,16-g7}} = y^{16} + x^9 - 7x^8 + 21x^7 - 35x^6 + 35x^5 - 21x^4 + 7x^3 - x^2 \tag{5.11.3}$$

$$\begin{aligned} f_{\texttt{16T12-4,2,8-g2}} = {} & y^{16} + (-1/8x^4 + 1/8x^3)y^{12} + (1/1024x^9 - 1/2048x^8 \\ & - 1/256x^7 + 1/256x^6)y^8 + (1/32768x^{12} - 1/32768x^{11})y^4 \\ & + 1/16777216x^{16} \end{aligned} \tag{5.11.4}$$

$$\begin{aligned} & f_{\texttt{16T10-4,2,4-g1}} = \\ & y^{16} + (-1/4x^4 + 1/2x^3 + 6x^2 + 8x - 4)y^{12} + (-5x^4 - 15x^3 + 20x^2)y^{10} \\ & + (3/128x^8 - 1/16x^7 + 19/16x^6 + 15x^5 - 83/4x^4 \\ & + 15/2x^3 + 34x^2 - 16x + 6)y^8 \\ & + (1/8x^8 - 47/8x^7 + 31/4x^6 + 6x^5 - 46x^4 + 46x^3 - 8x^2)y^6 \\ & + (-1/1024x^{12} + 1/512x^{11} - 11/128x^{10} + 17/16x^9 - 63/64x^8 \\ & - 29/8x^7 + 113/8x^6 - 13x^5 - 63/4x^4 + 79/2x^3 - 22x^2 + 8x - 4)y^4 \\ & + (3/256x^{12} - 23/256x^{11} + 1/64x^{10} + 9/16x^9 - 11/8x^8 \\ & + 9/8x^7 + 7/4x^6 - 6x^5 - x^4 + 17x^3 - 12x^2)y^2 \\ & + 1/65536x^{16} - 1/2048x^{14} + 1/512x^{13} + 3/1024x^{12} - 15/512x^{11} \\ & + 9/128x^{10} - 3/32x^9 + 35/128x^8 - 9/16x^7 + 3/16x^6 + 3/4x^4 \\ & + 1/2x^3 - 2x^2 + 1 \end{aligned} \tag{5.11.5}$$

$$
\begin{aligned}
&f_{\texttt{16T8-4,4,4-g3}} = \\
&y^{16} + (-1/4x^4 + 3/4x^3 - 3/4x^2 + 33/4x - 4)y^{12} \\
&+ (-20x^3 + 40x^2 - 20x)y^{10} \\
&+ (3/128x^8 - 9/64x^7 + 45/128x^6 + 481/32x^5 - 6899/128x^4 \\
&+ 4455/64x^3 - 2909/128x^2 - 33/4x + 6)y^8 \\
&+ (-11/2x^7 + 55/2x^6 - 55x^5 + 39x^4 + 25/2x^3 - 53/2x^2 + 8x)y^6 \\
&+ (-1/1024x^{12} + 9/1024x^{11} - 9/256x^{10} + 269/256x^9 - 3287/512x^8 \\
&+ 8991/512x^7 - 5305/256x^6 + 65/256x^5 + 15527/1024x^4 \\
&+ 3201/1024x^3 - 1135/64x^2 + 63/4x - 4)y^4 \\
&+ (-5/64x^{11} + 5/8x^{10} - 35/16x^9 + 27/8x^8 + 1/32x^7 - 65/8x^6 \\
&+ 205/16x^5 - 75/8x^4 + 987/64x^3 - 49/2x^2 + 12x)y^2 \\
&+ 1/65536x^{16} - 3/16384x^{15} + 33/32768x^{14} - 23/16384x^{13} \\
&- 721/65536x^{12} + 549/8192x^{11} - 2009/16384x^{10} - 995/8192x^9 \\
&+ 63855/65536x^8 - 33495/16384x^7 + 78241/32768x^6 - 20131/16384x^5 \\
&- 63679/65536x^4 + 2097/1024x^3 - 157/128x^2 + 1/4x + 1
\end{aligned}
\tag{5.11.6}
$$

Section 5.12

Degree 32

In degree 32 there are 151 isomorphism classes of 2-group Belyi maps with 96 distinct passports. All passports are of size 1 except for the following. All size 8 passports are listed in (5.12.1), the size 6 passport is listed in (5.12.2), all size 4 passports are listed in (5.12.3), the size 3 passport is listed in (5.12.4), and all size 2 passports are listed in (5.12.5).

$$\texttt{32S1-16,32,32-g15}, \quad \texttt{32S1-32,16,32-g15}, \quad \texttt{32S1-32,32,16-g15} \tag{5.12.1}$$

$$\texttt{32S15-8,8,8-g11} \tag{5.12.2}$$

$$\texttt{32S1-8,32,32-g14}, \quad \texttt{32S1-32,8,32-g14}, \quad \texttt{32S1-32,32,8-g14} \tag{5.12.3}$$

$$\texttt{32S6-4,4,4-g5} \tag{5.12.4}$$

$$\begin{array}{l}
\texttt{32S1-4,32,32-g12}, \quad \texttt{32S1-32,4,32-g12}, \quad \texttt{32S1-32,32,4-g12}, \\
\texttt{32S10-4,4,8-g7}, \quad \texttt{32S10-4,8,4-g7}, \quad \texttt{32S10-8,4,4-g7}, \\
\texttt{32S11-4,4,8-g7}, \quad \texttt{32S11-4,8,4-g7}, \quad \texttt{32S11-8,4,4-g7}, \\
\texttt{32S12-4,8,8-g9}, \quad \texttt{32S12-8,4,8-g9}, \quad \texttt{32S12-8,8,4-g9} \\
\texttt{32S16-8,16,16-g13}, \quad \texttt{32S16-16,8,16-g13}, \quad \texttt{32S16-16,16,8-g13} \\
\texttt{32S17-8,16,16-g13}, \quad \texttt{32S17-16,8,16-g13}, \quad \texttt{32S17-16,16,8-g13}
\end{array} \tag{5.12.5}$$

To obtain a list of degree 16 function fields corresponding to 2-group Belyi maps in characteristic 3, use the following code from [30].

5.12 Degree 32

Magma

```
load "config.m";

objs := [ReadTwoDBPassport(f) : f in PassportFilenames(32)];

fields := [* *];

for s in objs do

  fields cat:= FunctionFields(s);

end for;
```

Chapter 6

Future work

As is typically the case with any work of mathematics, there is more to investigate about 2-group Belyi maps.

Section 6.1

Implementation

One task is to compute more examples and optimize the implementations to aid in computing permutation triples and equations. In particular, there are two possible improvements to the implementations used here that could aid in this process.

1. Implement a way to lift Belyi maps over $\mathbb{F}_q$ to characteristic zero.

2. Take advantage of the iterative structure of these Belyi maps in the computation of Picard groups.

Another task is to extend these techniques to deal with p-group Belyi maps for $p \geq 3$ and non-Galois Belyi maps.

Section 6.2

Applications

Besides pushing the computations and generalizing the implementations, there is the more interesting question of how to apply these computations. Chapter 4, for example, provides evidence that 2-group Belyi maps appear to have small refined passports. This might suggest that the moduli fields of 2-group Belyi maps are *close to being abelian*. With Gross's conjecture in mind, we pose the following question.

Question 6.2.1. Are the moduli fields of 2-group Belyi maps always solvable?

More work is needed to conjecture on this question, but either answer is intersting. Since the moduli field of a 2-group Belyi map is ramified only at 2, a nonsolvable moduli field would be an example of a nonsolvable number field ramified only at 2. To illustrate the type of computation that would be helpful in answering Question 6.2.1, consider the permutation triple in (6.2.2) provided by David P. Roberts.

$$\begin{aligned}
\sigma_0^a &= (1, 20, 16, 28, 7, 21, 11, 32, 4, 17, 14, 26, 5, 24, 9, 29)\\
\sigma_0^b &= (2, 19, 15, 27, 8, 22, 12, 31, 3, 18, 13, 25, 6, 23, 10, 30)\\
\sigma_1^a &= (1, 11, 3, 10)(2, 12, 4, 9)(5, 14, 8, 15, 6, 13, 7, 16)\\
\sigma_1^b &= (17, 27, 23, 29, 19, 26, 21, 31, 18, 28, 24, 30, 20, 25, 22, 32)\\
\sigma_0 &= \sigma_0^a\sigma_0^b\\
\sigma_1 &= \sigma_1^a\sigma_1^b\\
\sigma_\infty &= (\sigma_1\sigma_0)^{-1}
\end{aligned}\tag{6.2.2}$$

The triple defined in (6.2.2) is a non-Galois permutation triple corresponding to a

2-group Belyi map of degree 2^{20} with refined passport size 16. Modifying our implementations to compute equations for these non-Galois triples could potentially shed light on an answer to Question 6.2.1.

Another interesting application of these computations would be to use the iterative structure of 2-group Belyi maps to explain the splitting of passports.

Lastly, even if all moduli fields of 2-group Belyi maps end up being solvable, there is still the possibility of finding a nonsolvable field by computing 2-torsion fields as described in Section 1.1. Torsion fields could be computed using techniques in [14, 28]. Torsion fields could also be computed using Christian Neurohr's implementation of Riemann surfaces in [10] which will be available in a future release.

Bibliography

[1] Sybilla Beckmann, *Ramified primes in the field of moduli of branched coverings of curves*, J. Algebra **125** (1989), no. 1, 236–255. MR 1012673

[2] G. V. Belyi, *On extensions of the maximal cyclotomic field having a given classical Galois group*, J. Reine Angew. Math. **341** (1983), 147–156. MR 697314

[3] Yakov Berkovich, *Groups of prime power order. Vol. 1*, De Gruyter Expositions in Mathematics, vol. 46, Walter de Gruyter GmbH & Co. KG, Berlin, 2008, With a foreword by Zvonimir Janko. MR 2464640

[4] Yakov Berkovich and Zvonimir Janko, *Groups of prime power order. Vol. 2*, De Gruyter Expositions in Mathematics, vol. 47, Walter de Gruyter GmbH & Co. KG, Berlin, 2008. MR 2464641

[5] ———, *Groups of prime power order. Volume 3*, De Gruyter Expositions in Mathematics, vol. 56, Walter de Gruyter GmbH & Co. KG, Berlin, 2011. MR 2814214

[6] ———, *Groups of prime power order. Vol. 4*, De Gruyter Expositions in Mathematics, vol. 61, De Gruyter, Berlin, 2016. MR 3445161

[7] Yakov G. Berkovich and Zvonimir Janko, *Groups of prime power order. Vol. 5*, De Gruyter Expositions in Mathematics, vol. 62, De Gruyter, Berlin, 2016. MR 3445342

[8] ———, *Groups of prime power order. Vol. 6*, De Gruyter Expositions in Mathematics, vol. 65, De Gruyter, Berlin, 2018. MR 3793194

[9] Wieb Bosma and John Cannon (eds.), *Discovering mathematics with Magma*, Algorithms and Computation in Mathematics, vol. 19, Springer-Verlag, Berlin, 2006, Reducing the abstract to the concrete. MR 2265375

[10] Wieb Bosma, John Cannon, and Catherine Playoust, *The Magma algebra system. I. The user language*, J. Symbolic Comput. **24** (1997), no. 3-4, 235–265, Computational algebra and number theory (London, 1993). MR MR1484478

[11] Kenneth S. Brown, *Cohomology of groups*, Graduate Texts in Mathematics, vol. 87, Springer-Verlag, New York, 1994, Corrected reprint of the 1982 original. MR 1324339

[12] Pete L. Clark and John Voight, *Algebraic curves uniformized by congruence subgroups of triangle groups*, Trans. Amer. Math. Soc. **371** (2019), no. 1, 33–82. MR 3885137

[13] Kevin Coombes and David Harbater, *Hurwitz families and arithmetic Galois groups*, Duke Math. J. **52** (1985), no. 4, 821–839. MR 816387

[14] Edgar Costa, Nicolas Mascot, Jeroen Sijsling, and John Voight, *Rigorous computation of the endomorphism ring of a Jacobian*, Math. Comp. **88** (2019), no. 317, 1303–1339. MR 3904148

[15] Lassina Dembélé, *A non-solvable Galois extension of $\mathbb{Q}$ ramified at 2 only*, C. R. Math. Acad. Sci. Paris **347** (2009), no. 3-4, 111–116. MR 2538094

[16] Lassina Dembélé, Matthew Greenberg, and John Voight, *Nonsolvable number fields ramified only at 3 and 5*, Compos. Math. **147** (2011), no. 3, 716–734. MR 2801398

[17] Luis V. Dieulefait, *A non-solvable extension of $\mathbb{Q}$ unramified outside 7*, Compos. Math. **148** (2012), no. 3, 669–674. MR 2925394

[18] David S. Dummit and Richard M. Foote, *Abstract algebra*, third ed., John Wiley & Sons, Inc., Hoboken, NJ, 2004. MR 2286236

[19] H. M. Farkas and I. Kra, *Riemann surfaces*, second ed., Graduate Texts in Mathematics, vol. 71, Springer-Verlag, New York, 1992. MR 1139765

[20] Alexandre Grothendieck, *Esquisse d'un programme*, Geometric Galois actions, 1, London Math. Soc. Lecture Note Ser., vol. 242, Cambridge Univ. Press, Cambridge, 1997, With an English translation on pp. 243–283, pp. 5–48. MR 1483107

[21] Robin Hartshorne, *Algebraic geometry*, Springer-Verlag, New York-Heidelberg, 1977, Graduate Texts in Mathematics, No. 52. MR 0463157

[22] D. F. Holt, *A computer program for the calculation of a covering group of a finite group*, J. Pure Appl. Algebra **35** (1985), no. 3, 287–295. MR 777260

[23] ______, *The mechanical computation of first and second cohomology groups*, J. Symbolic Comput. **1** (1985), no. 4, 351–361. MR 849042

[24] Gareth Jones and Manfred Streit, *Galois groups, monodromy groups and cartographic groups*, London Mathematical Society Lecture Note Series **243** (1997), 25–66.

[25] Michael Klug, Michael Musty, Sam Schiavone, and John Voight, *Numerical calculation of three-point branched covers of the projective line*, LMS J. Comput. Math. **17** (2014), no. 1, 379–430. MR 3356040

[26] Bernhard Köck, *Belyi's theorem revisited*, Beiträge Algebra Geom. **45** (2004), no. 1, 253–265. MR 2070647

[27] Gunter Malle and B. Heinrich Matzat, *Inverse Galois theory*, Springer Monographs in Mathematics, Springer, Berlin, 2018, Second edition [MR1711577]. MR 3822366

[28] Nicolas Mascot, *Computing modular Galois representations*, Rend. Circ. Mat. Palermo (2) **62** (2013), no. 3, 451–476. MR 3118315

[29] Rick Miranda, *Algebraic curves and Riemann surfaces*, Graduate Studies in Mathematics, vol. 5, American Mathematical Society, Providence, RI, 1995. MR 1326604

[30] Michael Musty, *2-group dessins*, `https://github.com/michaelmusty/2GroupDessins`, 2019.

[31] Michael Musty, Sam Schiavone, Jeroen Sijsling, and John Voight, *A database of Belyi maps*, Proceedings of the Thirteenth Algorithmic Number Theory Symposium, Open Book Ser., vol. 2, Math. Sci. Publ., Berkeley, CA, 2019, pp. 375–392. MR 3952023

[32] David P. Roberts, *Fractalized cyclotomic polynomials*, Proc. Amer. Math. Soc. **135** (2007), no. 7, 1959–1967. MR 2299467

[33] ______, *Nonsolvable polynomials with field discriminant* 5^A, Int. J. Number Theory **7** (2011), no. 2, 289–322. MR 2782660

[34] Michael Rosen, *Number theory in function fields*, Graduate Texts in Mathematics, vol. 210, Springer-Verlag, New York, 2002. MR 1876657

[35] Jean-Pierre Serre, *Abelian l-adic representations and elliptic curves*, McGill University lecture notes written with the collaboration of Willem Kuyk and John Labute, W. A. Benjamin, Inc., New York-Amsterdam, 1968. MR 0263823

[36] ______, *Congruences et formes modulaires [d'après H. P. F. Swinnerton-Dyer]*, (1973), 319–338. Lecture Notes in Math., Vol. 317. MR 0466020

[37] ______, *Topics in Galois theory*, second ed., Research Notes in Mathematics, vol. 1, A K Peters, Ltd., Wellesley, MA, 2008, With notes by Henri Darmon. MR 2363329

[38] Tanush Shaska, *Determining the automorphism group of a hyperelliptic curve*, Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation, ACM, New York, 2003, pp. 248–254. MR 2035219

[39] J. Sijsling and J. Voight, *On computing Belyi maps*, Numéro consacré au trimestre "Méthodes arithmétiques et applications", automne 2013, Publ. Math. Besançon Algèbre Théorie Nr., vol. 2014/1, Presses Univ. Franche-Comté, Besançon, 2014, pp. 73–131. MR 3362631

[40] Joseph H. Silverman, *Advanced topics in the arithmetic of elliptic curves*, Graduate Texts in Mathematics, vol. 151, Springer-Verlag, New York, 1994. MR 1312368

[41] ______, *The arithmetic of elliptic curves*, second ed., Graduate Texts in Mathematics, vol. 106, Springer, Dordrecht, 2009. MR 2514094

[42] David Singerman, *Finitely maximal Fuchsian groups*, J. London Math. Soc. (2) **6** (1972), 29–38. MR 0322165

[43] Henning Stichtenoth, *Algebraic function fields and codes*, second ed., Graduate Texts in Mathematics, vol. 254, Springer-Verlag, Berlin, 2009. MR 2464941