Applications of abelian algebraic structures in quantum computation

by

Kevin C. Zatloukal

B.S., University of Washington (2000)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author .... Signature redacted
Department of Electrical Engineering and Computer Science
May 18, 2016

Certified by .... Signature redacted
Aram W. Harrow
Assistant Professor
Thesis Supervisor

Accepted by .... Signature redacted
Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Applications of abelian algebraic structures in quantum computation

by

Kevin C. Zatloukal

Submitted to the Department of Electrical Engineering and Computer Science
on May 18, 2016, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

Shor's groundbreaking algorithms for integer factoring and discrete logarithm [58], along with their later generalizations [16, 35, 49, 18], demonstrated a unique ability of quantum computers to solve problems defined on abelian groups. In this thesis, we study ways in which that ability can be leveraged in order to solve problems on more complex structures such as non-abelian groups and hypergroups.

This leads to new quantum algorithms for the hidden subgroup problem on nilpotent groups whose order is a product of large primes, the hidden subhypergroup problem on both strongly integral hypergroups and ultragroups, testing equivalence of group extensions, and computing the component parts of the cohomology groups of both group extensions and a generalization of simplicial complexes, amongst other problems. For each of those listed, we also show that no classical algorithm can achieve similar efficiency under standard cryptographic assumptions.

Thesis Supervisor: Aram W. Harrow
Title: Assistant Professor

Acknowledgments

I would first like to thank Aram Harrow for supervising all of the work that went into this thesis. I am indebted to him not only for providing critical insights at many points during this work but also for consistently encouraging me along the way. I would also like to thank Ike Chuang and Scott Aaronson for serving on my both my thesis committee and my qualifying exam committee and for the helpful feedback that they provided. I am also thankful to the Center for Theoretical Physics at MIT for providing a home for me during the time when much of this work was done.

I have been very fortunate to have collaborated with Juan Bermejo Vega. Many of the fruits of that collaboration are described in chapter 5. More importantly, I was lucky to meet such a wonderful person, who is not only a great collaborator but also a great friend. I would like to thank the Max Planck Institute for Quantum Optics as well for hosting me and giving me a chance to experience another culture while I collaborated on this work with Juan.

I was remarkably fortunate to have received so much support from experienced researchers as an undergraduate. I am indebted to Richard Ladner for taking me under his wing when I was an aspiring computer scientist. It is no exaggeration to say that I would have accomplished very little of what I have without having received so much of his time and advice. I would like to thank Martin Tompa as well for encouraging, advising, and supporting me as an undergraduate.

I am deeply grateful to my family for putting up with many absences while I traveled and long hours when I did not, as well as for their encouragement and love.

This work was supported by NSF grant CCF-1111382 and the iQuISE IGERT program at MIT.

This thesis is dedicated to my father, Jim Zatloukal, who lived a life nearly perfect in every respect but its duration.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Motivation Shor's groundbreaking algorithms for integer factoring and discrete logarithm [58] demonstrated the power of quantum computers to solve problems believed to be intractable for classical computers [17]. Since then, researchers have strived to understand the source of this quantum advantage and to find ways to generalize it.

Quickly, the focus shifted from the more narrow number-theoretic problems considered initially by Shor to the Hidden Subgroup Problem (HSP), a more general problem about finite groups [16, 35, 49]. This includes as special cases not only discrete logarithm but also Simon's problem [59], for which an exponential speedup of quantum algorithms over classical ones had also been demonstrated. Brassard and Høyer [16, 35] showed that the techniques developed by Shor can be extended to efficiently solve the HSP for all finite *abelian* groups.

Shortly after, Cheung and Mosca [18] showed, using the solution to the abelian HSP along with Shor's order finding algorithm, that quantum computers can efficiently decompose an abelian group, given as black box [4], into its cyclic factors. This generalizes both the algorithms for factoring and discrete logarithm, in fact, for order finding and a variant of the general abelian HSP. Indeed, the ability to understand abelian groups, by decomposing them into their cyclic factors, makes essentially every natural problem defined on abelian groups efficiently solvable.[1]

[1] See [56] for a list of common problems defined on groups.

A similar story occurred for infinite abelian groups in the form of vector spaces. There, the ability to decompose a subgroup specified as column space of a matrix into its cyclic components (i.e., eigenspaces) leads, after substantial work, to the ability to solve linear systems [33].

In summary, prior work has demonstrated that quantum computers have a unique ability to analyze and understand abelian algebraic structures. It is natural to wonder, then, to what extent this ability extends to more complex structures such as non-abelian groups.

Overview In this thesis, we will examine in detail ways in which we can solve problems on more complex structures by leveraging, in various ways, the ability of quantum computers to understand and solve problems on abelian algebraic structures.

The idea of using simple abelian groups (often vector spaces) in order to understand more complex objects is ubiquitous in mathematics. For example, the approach of approximating a nonlinear function near a point by its tangent plane (a vector space) will be familiar to most. However, similar approaches are also used to analyze differential equations [27], topological spaces [34], differentiable manifolds [14], non-abelian groups [24], amongst others.

Our aim is make this idea computational, that is, to show not only that we can understand more complex structures by relating them to simple abelian ones but also that we can use such relationships to solve problems more efficiently. Furthermore, since quantum computers have a unique ability to solve problems on abelian algebraic structures, this approach is more likely to be successful using quantum rather than classical computation.

We will examine in detail three particular ways of leveraging the ability of quantum computers to understand abelian algebraic structures in order to solve problems on more complex ones:

1. Taking advantage of a well-organized collection of abelian substructures.

2. Translating a non-abelian structure into an abelian one.

3. Examining abelian groups of maps from our structures into an abelian group.

In the coming chapters, we will explain each of these approaches in detail. For each, we will see multiple examples of how we can use that technique to design quantum algorithms that solve problems more efficiently than previously possible. Furthermore, in each case, we will be able to provide strong evidence that no classical algorithm can achieve the same result.

Structure This thesis is organized as follows. We begin, in chapter 2, by reviewing some mathematical background that will be necessary to understand our main results. In chapter 3, we review some background on quantum computing in general as well as existing quantum algorithms that will prove useful to us later on.

We begin the body of the thesis in chapter 4, where we show that the well-organized collection of abelian subgroups in any nilpotent group allow us to solve the HSP more efficiently. Next, in chapter 5, we show that translating a non-abelian group into an abelian hypergroup allows us to efficient solve the HSP for normal subgroups. Then, in chapter 6, we show that examining certain abelian groups of maps into an abelian group allows us to efficiently test whether two group extensions (non-abelian groups) are the same and also to efficiently prove that certain topological spaces are not the same.

Finally, we provide a summary of the results in chapter 7.

Chapter 2

Mathematical Background

In this chapter, we will review some background material on mathematics that will be necessary for understanding the main results.

We will assume that the reader is familiar with standard algebraic objects such as groups, vector spaces, rings, and algebras (rings that are also vector spaces). However, we will review their definitions and those standard results that we will need below, starting with group theory in section 2.1. For proofs of these results, consult any standard algebra text such as [43, 24].

We will assume that the reader is conversant with linear algebra including, for example, direct sums and tensor products of vector spaces as well as invertible and unitary transformations of the same. Definitions of these can also be found in algebra texts like [43, 24] as well as the standard text on quantum computation [51].

We denote the integers and complex numbers by $\mathbb{Z}$ and $\mathbb{C}$, respectively, and the ring of integers modulo a prime p by $\mathbb{Z}_p$. If V is a vector space (e.g., $\mathbb{C}^n$), then we will denote the invertible and unitary transformations of V by $\mathrm{GL}(V)$ and $\mathrm{U}(V)$, respectively. The identity transformation will be denoted by just I, as the space on which it acts will be clear from context.

If a linear transformation is given by a matrix A (i.e., we have chosen a preferred basis), then we denote the (i, j) entry of this matrix by $[A]_{i,j}$. We denote the transpose of A by A^T and the conjugate transpose by $A^\dagger := \overline{A}^T$. Following the usual convention, we denote the (i, j) entry of the identity transformation (in any basis) by $\delta_{i,j}$.

We will work exclusively with the field of complex numbers in this thesis. In particular, the scalars in our vector spaces (and algebras) will always come from $\mathbb{C}$. We will also always assume that our vector spaces possess an inner product, which we denote $\langle v \,|\, w \rangle$ for vectors v and w. In fact, we will most often work not with simple vector spaces but with Hilbert spaces. This requires no special assumptions as all vector spaces will be finite dimensional.

2.1 Groups Theory

2.1.1 Groups

Recall that a **group** G is a set endowed with a special type of binary operation. We will normally refer to this operation, when applied to $x \in G$ and $y \in G$, as the product of x and y and denote the result by the juxtaposition xy. In certain contexts where the group is required to be **abelian** (that is, $xy = yx$ for all $x, y \in G$), we will instead denote the operation by $x + y$ and refer to the result as the sum of x and y.

For G to be a group, the product must be associative, it must have an identity element, which we will normally denote by e, and each $x \in G$ must have an inverse, denoted x^{-1}, which satisfies $xx^{-1} = e = x^{-1}x$.

If K is another group, then a map $\varphi : G \to K$ is a **homomorphism** if, for any $x, y \in G$, we have $\varphi(xy) = \varphi(x)\varphi(y)$. If φ is a bijection (one-to-one and onto), then φ is an **isomorphism**. In the latter case, G and K are isomorphic, denoted $G \cong K$, and are really the same group, just labelled differently.

A subset $H \subset G$ that is still a group (with the same product) is called a **subgroup** and is denoted $H \leq G$. The identity and inverses in the subgroup H are necessarily the same as those of the larger group G.[1]

The set of elements $x \in G$ such that $\varphi(x)$ is the identity is the **kernel** of φ, which we denote $\operatorname{Ker}\varphi$. The kernel and image of a homomorphism φ are always subgroups, that is, $\operatorname{Ker}\varphi \leq G$ and $\operatorname{Im}\varphi \leq K$.

[1] This is because identities and inverses are always unique in any group.

2.1.2 Cosets

The relation $x \sim_H y$, defined to hold when there exists an $h \in H$ such that $xh = y$, is an equivalence relation. The set of elements in the class of x is denoted $xH := \{xh \,|\, h \in H\}$ and referred to as a (left) **coset** of H. If we instead define $x \sim^H y$ to hold when there exists an $h \in H$ such that $hx = y$, then the class containing x is the right coset $Hx := \{hx \,|\, h \in H\}$. The number of such cosets (whether left or right) is denoted $[G : H]$ and equals $|G|/|H|$.

More generally, if $H \leq G$ and $K \leq G$ are two subgroups, then we can also define an equivalence relation with $x \sim_K^H y$ whenever there exists $h \in H$ and $k \in K$ such that $hxk = y$. The class containing x is then the set $HxK := \{hxk \,|\, h \in H,\ k \in K\}$, which is called a **double coset**. (Left and right cosets are the special cases where H or K, respectively, is the trivial subgroup $\{e\}$.)

2.1.3 Conjugation and Normality

For $g, x \in G$, the **congugate** of x (by g) is the element $g^{-1}xg$, which we denote by x^g. Once again, the relation $x \sim_{\text{cong}} y$, defined to hold when there exists a $g \in G$ such that $x^g = y$, is an equivalence relation. The class containing x, called the **conjugacy class** of x, is $C_x := \{x^g \,|\, g \in G\}$.

If $H \leq G$ is a subgroup that contains all the conjugates of its elements, that is, where $h^g \in H$ for every $h \in H$ and $g \in G$, then H is called **normal**. We write $H \lhd G$ to denote that H is a normal subgroup.

If $H \lhd G$, then $xH = Hx$ for any $x \in G$, so we need not distinguish between left and right cosets. This follows from the fact that $xh = xhx^{-1}x = h^x x = h'x$ for some $h' \in H$. This means that we can define multiplication on cosets since $xHyH = xyH$. In this case, the cosets themselves form a group, called the **quotient group** of H, which we denote G/H.[2]

If we have a firm understanding of a normal subgroup $H \lhd G$ and quotient group G/H, then we have made substantial progress toward understanding G. Indeed, one

[2] The identity is eH and the inverse of xH is $x^{-1}H$.

of the basic programs aiming to classify all finite groups proceeds by (1) classifying those groups with no nontrivial normal subgroups and then (2) identifying all ways that groups H and K can be put together into a group G such that $H \lhd G$ and $K \cong G/H$. Step (1) has been accomplished: this is the classification of finite simple groups. Step (2) has been accomplished in the case where H is abelian: this is the theory of group extensions.

If $H \lhd G$ is abelian and $K \cong G/H$, then G is said to be an **extension of K by H**. The theory of group extensions describes all the ways in which this can be done for fixed H and K. The groups that be constructed by a sequence of group extensions starting from an abelian group are precisely the supersolvable groups, which we discuss in section 2.1.4. Thus, the program described above has been essentially carried out for classifying the supersolvable groups.[3]

Normal subgroups arise in many circumstances. One of the most important ways they arise, however, is as kernels of homomorphisms: we noted above that $\operatorname{Ker}\varphi$ is always a subgroup, and it is in fact normal since, if $\varphi(x)$ is the identity, then $\varphi(x^g) = \varphi(g^{-1})\varphi(x)\varphi(g) = \varphi(g^{-1})\varphi(g) = \varphi(g^{-1}g) = \varphi(e)$ is the identity.

In fact, we can see that *every* normal subgroup $H \lhd G$ arises as the kernel of a homomorphism, namely, the map $x \mapsto xH$, which is a homomorphism whose kernel is precisely the subgroup H. This function $G \to G/H$ is called the quotient map.

If $\varphi : G \to K$ is an arbitrary homomorphism, then, since φ is equal on cosets of $\operatorname{Ker}\varphi$, we can define a new homomorphism $\tilde{\varphi} : G/\operatorname{Ker}\varphi \to K$ by $\tilde{\varphi}(x) = \varphi(x\operatorname{Ker}\varphi)$, which is, by definition, an injective map. This means that $\tilde{\varphi}$ is a bijection onto the image $\operatorname{Im}\varphi$, which shows $G/\operatorname{Ker}\varphi \cong \operatorname{Im}\varphi$. This is the First Isomorphism Theorem.

We will also make use, often implicitly, of the fact that, if $H \lhd G$, then the subgroups of G/H are in one-to-one correspondence with the subgroups $K \leq G$ satisfying $H \leq K$. This correspondence simply takes $K \leq G$ to $K/H \leq G/H$. Furthermore, it can be shown that K/H is normal in G/H iff K is normal in G. This correspondence is often called the Fourth (or Lattice) Isomorphism Theorem.[4]

[3] The one difference is that this classifies supersolvable groups not up to isomorphism but rather up to **equivalence**. The latter notion will be defined later on (see chapter 6).

[4] This result tells us, in particular, that, if we know the subgroups of $H \lhd G$ and of G/H, then

2.1.4 Important Types of Groups

Abelian Groups

Abelian groups are the simplest type of groups. Of abelian groups, the simplest kind are the **cyclic** groups. These are simply $\mathbb{Z}_m$, for some $m \in \mathbb{Z}$, where the operation is addition modulo m. The identity is 0 and the inverse of $x \in \mathbb{Z}_m \setminus \{0\}$ is $m - x \in \mathbb{Z}_m$.

If A and B are any two groups (abelian or not), then we can form a new group on the set $A \times B = \{(x, y) \mid a \in A,\ b \in B\}$ by defining the product of (u, x) and (v, y) to be (uv, xy). The identity is the element (e, e) in this set[5], and the inverse of (u, x) is (u^{-1}, x^{-1}). The resulting group, which we also denote by $A \times B$, is called the **direct product** of A and B.

If A and B are themselves abelian, then we can see that $A \times B$ is also an abelian group. Hence, one way to form more complicated abelian groups is to take direct products of the cyclic groups. In fact, this is the only way to form abelian groups.

Theorem 2.1 (Fundamental Theorem of Finite Abelian Groups). *Let G be an abelian group. Then there exist primes $p_1, \dots, p_m \in \mathbb{Z}$ and numbers $t_1, \dots, t_m \in \mathbb{Z}$ such that*

$$G \cong \mathbb{Z}_{p_1^{t_1}} \times \cdots \times \mathbb{Z}_{p_m^{t_m}}.$$

Supersolvable Groups

A group G is called **supersolvable** if there is an increasing sequence of subgroups $\{e\} = N_0 \leq N_1 \leq \cdots \leq N_k = G$, all normal in G, where $[N_{i+1} : N_i]$ is prime $\forall\, i \in \mathbb{Z}_k$.

This, in particular, means that the normal subgroup $N_1 \lhd G$ is isomorphic to $\mathbb{Z}_p$ for some prime p, so it is abelian. Hence, G is put together out of the groups N_1 and G/N_1 with $N_1 \lhd G$ abelian. In the quotient group G/N_1, we also have an increasing sequence of subgroups $\{e\} \cong N_1/N_1 \leq N_2/N_1 \leq \cdots \leq N_k/N_1 = G/N_1$. Hence, G/N_1 is itself supersolvable, and by induction, we can see that G can be built up a sequence of group extensions: we start with the group N_k/N_{k-1}, which is abelian by

we know all the subgroups lying below H and above H in the lattice of subgroups of G.

[5] The first e refers to the identity of A and the second to the identity of B. We will avoid labeling the identities of the various groups since it should be clear from context.

assumption, extend N_{k-1}/N_{k-2} (also abelian) by it, extend N_{k-2}/N_{k-3} by that, and so on until we extend $N_1/N_0 = N_1$ by G/N_1 to get the full group G.

This shows that any supersolvable group can be built from an abelian group by a sequence of extensions. On the other hand, if G is built by starting from a group A_k and then extending by A_{k-1}, then A_{k-2}, and so on down to A_1, with all A_i's abelian, then by the definition of extensions, G has an increasing sequence of normal subgroups $\{e\} \leq N_1 \leq \cdots \leq N_k$, where $N_{i+1}/N_i \cong A_{i+1}$ is abelian for each $i \in \mathbb{Z}_k$. This sequence can be extended to a longer sequence where with factor is cyclic of prime order (instead of just abelian) using the decomposition from Theorem 2.1. This allows us to show the supersolvable groups are precisely those that can be built from an abelian group by a sequence of extensions, as claimed above.

One way to construct a supersolvable group is to form the **semidirect product** of A and B, with A a abelian and B supersolvable. Like the direct product, the set of elements is the cartesian product $A \times B$. However, the product is generally a more complicated operation than in the direct product.

To define a product operation for a semidirect product, we need a homomorphism $\varphi : B \to \operatorname{Aut} A$, where $\operatorname{Aut} A$ denotes the set of automorphisms of A, that is, the set of isomorphisms from A to itself.[6] For any $b \in B$, the image $\varphi(b) \in \operatorname{Aut} A$ is a map $A \to A$, which can be applied to any $a \in A$ to get some $\varphi(b)(a) \in A$. To simplify the notation, we often write the application of φ to b as a subscript, φ_b, which turns the cumbersome notation $\varphi(b)(a)$ into the more sensible $\varphi_b(a)$.

With this homomorphism in hand, we define the product of (u, x) and (v, y) in the semidirect product to be the element $(u\varphi_x(v), xy)$. In that case, (e, e) becomes the identity and $(\varphi_x^{-1}(u^{-1}), x^{-1})$ is the the inverse of (u, x), which tells us that the semidirect product is a group. We denote this group by $A \rtimes B$ or (more explicitly) by $A \rtimes_\varphi B$. Note that the direct product is the special case where φ is the trivial homomorphism, that is, where every φ_x is the identity map.

To see that the semidirect product $A \rtimes B$ is supersolvable, we will show that it

[6] This is, in fact, a group where the product is composition; the identity map is the identity element; and for each automorphism, the inverse mapping (also an automorphism) is its inverse.

is an extension of A. Since B is supersolvable by assumption, B is built up from an abelian group by a sequence of extensions. Thus, forming an extension of A by B gives a supersolvable group in $A \rtimes B$.

To show that $A \rtimes B$ is an extension of A, let us define $A' = \{(a, e) \mid a \in A\}$. Since this is isomorphic to A, it is abelian. Next, we must show that A' is also normal.

To see this, we compute the conjugate $(a, e)^{(u,x)} = (\varphi_x^{-1}(u^{-1}), x^{-1})(a, e)(u, x) = (\varphi_x^{-1}(u^{-1}), x^{-1})(au, x) = (\varphi_x^{-1}(u^{-1})\varphi_x^{-1}(au), x^{-1}x) = (\varphi_x^{-1}(u^{-1}au), e)$. Now, since A is abelian, we have $u^{-1}au = a$, and since $\varphi_x^{-1} \in \operatorname{Aut} A$, we must have $\varphi_x^{-1}(a) \in A$. Thus, we conclude that $(a, e)^{(u,x)} \in A'$, which proves that A' is normal.

Finally, we can see see that the cosets of A' in $A \rtimes B$ are of the form $(A, b) := \{(a, b) \mid a \in A\}$. These are in one-to-one correspondence with elements of B, and once we quotient by A' and ignore the left element in each pair, what remains is isomorphic to the group B. Thus, we have seen that $A \rtimes B$ is an extension of B.

Example 2.2 (Dihedral Group). *The simplest case for a semidirect product is to choose both A and B to be cyclic groups, and the simplest possible choice for the latter is $B = \mathbb{Z}_2$. Now, for any abelian group A, the map $\varphi_{inv} : A \to A$ taking $x \in A$ to $x^{-1} \in A$ is an automorphism, and since $(x^{-1})^{-1} = x$, we can see that φ_{inv}^2 is the identity map. Hence, the map $\varphi : \mathbb{Z}_2 \to \operatorname{Aut} A$ with $\varphi(0) = \mathrm{I}$ and $\varphi(1) = \varphi_{inv}$ is a homomorphism. If we take $A = \mathbb{Z}_n$, then the resulting semidirect product $\mathbb{Z}_n \rtimes_{\varphi_{inv}} \mathbb{Z}_2$ is called a **dihedral group** and is denoted by D_{2n}.*

For the dihedral group, it is normal to write the group operation in both cyclic groups as addition, so the inverse of $x \in \mathbb{Z}_n$ can be written as $-x \pmod n$. Then the group operation has $(u, x)(v, 0) = (u + v, x)$ and $(u, x)(v, 1) = (u - v, x + 1)$.

Nilpotent Groups

The most important class of finite groups for us is the class of nilpotent groups. As we shall see shortly, the nilpotent groups lie between the classes of abelian and supersolvable groups: every nilpotent group is supersolvable but most are not abelian.

To define the nilpotent groups, first, recall that the **center** of G, denoted $Z(G)$, is the set of elements that commute with every other element in the group. In particular, the center is itself an abelian group.

We can define the nilpotent groups recursively as follows: a group G is nilpotent if it is trivial ($\{e\}$) or if $Z(G)$ is nontrivial and $G/Z(G)$ is nilpotent. Hence, the abelian groups are nilpotent since, in that case, $Z(G) = G$ and $G/Z(G) = G/G \cong \{e\}$ is trivial. Continuing one step further, if $G/Z(G)$ is abelian, then G is nilpotent. If G is not abelian but $G/Z(G)$ is abelian, then the group is called **2-nilpotent**. Continuing further, if $(G/Z(G))/Z(G/Z(G))$ is abelian, then G is nilpotent. If $G/Z(G)$ is not abelian but $(G/Z(G))/Z(G/Z(G))$ is abelian, then the group is called **3-nilpotent.**

In general, G is nilpotent if we can eventually reduce it to $\{e\}$ by repeatedly taking the quotient by the center of what remains. If it takes k quotients to reduce G to $\{e\}$, then G is called **k-nilpotent**. If G is not nilpotent, then we will eventually reduce to a group that is not abelian and has a trivial center (so taking the quotient by the center does not reduce the group any further).

This recursive definition immediately implies that a nilpotent group is built up by a sequence of extensions from an abelian group since, at each step, $Z(G)$ is an abelian, normal subgroup.[7] Hence, every nilpotent group is supersolvable. If G is an extension of H by K where $H \lhd G$ is contained in the center of G, then this is referred to as a **central extension**. In other words, the nilpotent groups are the special case of those supersolvable groups that are built up from an abelian group by a sequence of *central* extensions.

We can make some definitions that will simplify our discussions of nilpotent groups. Rather than working with repeated quotients, we can instead work entirely in the group G. Let $Z_1(G) := Z(G)$. By the Fourth Isomorphism Theorem, we know that the subgroups of $G/Z(G)$ are in one-to-one correspondence with subgroups of G containing $Z(G)$. In particular, there is a subgroup of G containing $Z(G)$ that corresponds to $Z(G/Z(G))$. We define this to be $Z_2(G)$. Continuing on, the subgroups of $(G/Z(G))/Z(G/Z(G))$ correspond to those of $G/Z(G)$ containing $Z(G/Z(G))$, which correspond to those of G containing $Z_2(G)$. Thus, there is a subgroup of G containing $Z_2(G)$ that corresponds to the center of $(G/Z(G))Z(G/Z(G))$,

[7] For $z \in Z(G)$ and $g \in G$, we have $z^g = g^{-1}zg = z$ since z commutes with g.

which we define to be $Z_3(G)$. By this process, we get a sequence of normal subgroups $Z_1(G) \leq Z_2(G) \leq Z_3(G) \leq \dots$ in G. If G is nilpotent, then we eventually get $Z_k(G) = G$ for some k and G is k-nilpotent.[8]

The simplest example of a nilpotent group that we are aware of is the following.

Example 2.3. *The (finite)* ***Heisenberg group*** *is the set $\mathbb{Z}_p^3$, for p prime, with a group operation defined by*

$$(x, y, z)(x', y', z') = (x + z', y + y' + xz', z + z').$$

The identity is $(0, 0, 0)$ and the inverse of (x, y, z) is $(-x, -y + xz, -z)$.

This is in fact a special case of the following.

Example 2.4. *A $(k+1)\times(k+1)$ matrix with elements in $\mathbb{Z}_p$ is called* ***unitriangular*** *if it is of the form*

$$\begin{pmatrix} 1 & a_{1,2} & a_{1,3} & \dots & a_{1,k} \\ 0 & 1 & a_{2,3} & \dots & a_{2,k} \\ 0 & 0 & 1 & \dots & a_{k-1,k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix}$$

for some $a_{i,j}$'s in $\mathbb{Z}_p$. We denote by UT_p^k the group of such matrices with product given by matrix multiplication.

A short calculation shows that the set of matrices that are only nonzero in the upper-right corner (at $a_{1,k}$) is the center of the group. More generally, the t-th center, $Z_t(\mathrm{UT}_p^k)$, consists of those matrices that are only nonzero in the t highest superdiagonals, i.e., only at $a_{i,j}$ with $j - i \geq k - t$. Hence, the group UT_p^k is k-nilpotent.

The group UT_p^2 is actually just the Heisenberg group.

One very large source of nilpotent groups are the p-groups. A group G is called a **p-group** if $|G| = p^m$ for some m, where p is prime. It follows from the Class Formula [43] that every p-group has a nontrivial center. Furthermore, since the size of a

[8] The number k is also called the **nilpotency class** of k.

subgroup divides the size of the group, we have $|Z(G)| = p^\ell$ for some $\ell \leq m$, which means that $|G/Z(G)| = p^m/p^\ell = p^{m-\ell}$. Thus, $G/Z(G)$ is also a p-group, so it also has a nontrivial center. Arguing inductively, we see that every p-group is nilpotent.

This gives us a proof that the unitriangular matrix groups UT_p^k are nilpotent since each is manifestly a p-group. As another special case, we can see that, if $n = 2^m$, then D_{2n} is a 2-group. Hence, while dihedral groups are usually not nilpotent, they are in the special case where n is a power of 2.

Another way to construct nilpotent groups is to take direct products of groups that we know are nilpotent. This follows since $Z(A \times B) = Z(A) \times Z(B)$. Hence, if A is k-nilpotent and B is ℓ-nilpotent, then we certainly have $Z_{\max\{k,\ell\}}(A \times B) = Z_{\max\{k,\ell\}}(A) \times Z_{\max\{k,\ell\}}(B) = A \times B$, meaning that $A \times B$ is $\max\{k, \ell\}$-nilpotent.

In fact, the following standard result [43] shows, amongst other things, that taking direct products of p-groups is actually the *only* way to make nilpotent groups.

Theorem 2.5. *The following are equivalent:*

1. *G is a nilpotent group.*

2. *G is a direct product of its maximal p-subgroups*[9] *(for different primes p).*

3. *Every maximal subgroup of G is normal.*

4. *Every proper subgroup $H \leq G$ is a proper subgroup of its normalizer.*

The normalizer of $H \leq G$, denoted $N_G(H)$, is the largest subgroup of G in which H is normal. Since H is a subgroup, we certainly have $H^h = H$ for every $h \in H$, so we know that $H \leq N_G(H)$. In a nilpotent group, however, the theorem tells us that the normalizer is always strictly larger, so there is some $g \in G \setminus H$ for which $H^g = H$.

Lastly, we will show that nilpotent groups can also be defined in a different manner, not with centers but rather with so-called derived subgroups.

To explain this concept, we first need to define **commutators**. If $x, y \in G$, then we can see that $xy = yxx^{-1}y^{-1}xy$. If we define the commutator $[x, y] := x^{-1}y^{-1}xy$,

[9] A p-subgroup of G is a subgroup that is itself a p-group.

then we have $xy = yx[x, y]$. The commutator $[x, y]$ measures the extent to which x and y commute (in particular, it is e if they do commute). They are often useful in calculations because they allow us to swap the order of multiplication of x and y. This has the cost of introducing a factor of $[x, y]$, but if we know something about commutators in that group, then it is often easy to deal with the extra factor. If our group is nilpotent, then we will have quite a bit of information about commutators.

The **derived subgroup** of G, denoted $[G, G]$, is the subgroup generated by the commutators. (Unfortunately, the product of two commutators is not always a commutator, so the set of commutators does not always form a subgroup. Instead, we must take the smallest subgroup that contains all the commutators.) Let us define $G^1 := [G, G]$ and then, inductively, $G^i := [G^{i-1}, G]$. It is can be shown that there is a k such that $G^k = \{e\}$ iff G is nilpotent [24], and in fact, this is the same k such that $Z_k(G) = G$, i.e., the nilpotency class of G.

In a k-nilpotent group, every commutator $[x, y]$ lies in the subgroup Z_{k-1}. More generally, the elements of G^i, which are those that can be written as $[[y_i, y_{i-1}], \dots], y_1]$ for some $y_1, \dots, y_i \in G$, lie in an even smaller subgroup [24]:

Theorem 2.6. *Let G be k-nilpotent. Then, for $0 \leq i \leq k$, we have $G^i \leq Z_{k-i}(G)$.*

We mention one final fact about p-groups that will be useful to us in developing quantum algorithms. Since $Z(G)$ is abelian, we know by Theorem 2.1 that it is of the form $\mathbb{Z}_{p_1^{t_1}} \times \cdots \times \mathbb{Z}_{p_m^{t_m}}$ for some p_i's and t_i's. This means that we can choose $g_{1,1}, \dots, g_{1,m} \in Z(G)$ such that $g_{1,i}$ generates the part isomorphic to $\mathbb{Z}_{p_i^{t_i}}$. More generally, since $Z_{i+1}(G)/Z_i(G)$ is abelian, it is isomorphic to a direct product of cyclic groups, and we can choose $g_{i,1}, \dots, g_{i,m_i}$ such that $g_{i,j}Z_i(G)$ generates the j-th cyclic group in this direct product decomposition. Repeating this for $i = 1, \dots, k$, where G is k-nilpotent gives a set of $g_{i,j}$'s that generators G and have the property that each element of G can be written *uniquely* as $\prod_{i,j} g_{i,j}^{e_{i,j}}$ for some $e_{i,j}$'s in $\mathbb{Z}$.

Unique representation is a useful feature that is typically assumed by quantum algorithms for group problems, while it is typically not assumed by classical ones. For our case, however, the discussion above shows that we can assume unique representation without loss of generality.

Above, we defined a supersolvable group to be one that can be constructed by a sequence of extensions. We have now seen two examples how to construct an group extension: (1) a semidirect product as in a dihedral group and (2) a central extension as in the Heisenberg group. Appendix B describes some of the deeper theory about group extensions. In particular, we will see in the appendix that (1) and (2) are essentially the only two ways to construct extensions: every group extension can be thought of as just a combination of (1) and (2). This and other important facts about extensions are proven by studying the properties of the second cohomology group of the extension, something that will play a large role in chapter 6.

2.2 Representation Theory

2.2.1 Basic Definitions

A (complex linear) **representation** of a group G is a homomorphism $G \to \mathrm{GL}(V)$ into the space of invertible transformations of some vector space V. If ρ is a representation, we will denote the vector space V that appears in the codomain by V_ρ and the dimension of that vector space by $d_\rho := \dim V_\rho$.

Suppose that ρ is a representation, with $V_\rho = \mathbb{C}^n$, and we choose another basis for $\mathbb{C}^n$. Let $T \in \mathrm{GL}(\mathbb{C}^n)$ be the change to this new basis. Then we can define a new representation $\sigma : G \to \mathrm{GL}(\mathbb{C}^n)$ just by changing to this basis: if we define $\sigma(g) = T\rho(g)T^{-1}$, then σ is also a representation.

While ρ and σ are technically distinct representations, they do not differ in any important sense. We will say that they are isomorphic as representations, meaning that, while they are not identical, they are really "the same" representation.

More generally, suppose that $\rho : G \to \mathrm{GL}(V)$ and $\sigma : G \to \mathrm{GL}(W)$ are two representations of G. A linear transformation $T : V \to W$ satisfying $\sigma(g) = T\rho(g)T^{-1}$ (or equivalently, $\sigma(g)T = T\rho(g)$) for all $g \in G$ — i.e., a map transforming ρ into σ — is called an **intertwining map**. If T is invertible, then T^{-1} also transforms σ into ρ

and we say that ρ and σ are **isomorphic**.

To remove this redundancy of isomorphic representations, we can instead consider the **character** of ρ defined by $\chi_\rho(g) := \operatorname{tr} \rho(g)$. By the cyclic property of the trace, we have $\chi_\sigma \equiv \chi_\rho$. Hence, the character functions are equal for all representations that are identical after a change of basis. In fact, the opposite is also true: representations with equal characters only differ by a change of basis. (See appendix A for details.) Hence, the character function characterizes a representation up to the choice of basis.

2.2.2 Examples of Representations

Example 2.7. *For any group G, we can define a representation* triv $: G \to U(\mathbb{C})$ *by* $\text{triv}(g) = 1$ *for all $g \in G$. This is called the* ***trivial representation****. We have* $\chi_{\text{triv}} \equiv 1$.

Example 2.8. *For any group G, let V_G be the vector space with a basis vector $|g\rangle$ for each $g \in G$. We can define a representation* reg $: G \to \mathrm{U}(V_G)$ *by making* $\text{reg}(g)$ *act on $|h\rangle$ by* $\text{reg}(g)|h\rangle = |gh\rangle$*. This is called the (left)* ***regular representation****. For this representation, we have $\chi_{\text{reg}}(g) = 0$ for $g \neq e$ and $\chi_{\text{reg}}(e) = |G|$.*

We will sometimes use a subscript (e.g., reg_G) to clarify the group in question.

Example 2.9. *Given two representations ρ and σ, we can define a new representation* $\rho \oplus \sigma : G \to \mathrm{GL}(V_\rho \oplus V_\sigma)$ *by* $(\rho \oplus \sigma)(g) = \rho(g) \oplus \sigma(g)$*. This is called the* ***direct sum*** *of the representations ρ and σ. It has the property that $\chi_{\rho\oplus\sigma} = \chi_\rho + \chi_\sigma$.*

Example 2.10. *Replacing $\oplus$ with $\otimes$ in the previous example gives the* ***tensor product*** *of the representations ρ and σ. We can check that $\chi_{\rho\otimes\sigma}(g) = \chi_\rho(g)\chi_\sigma(g)\ \forall g \in G$ by direct calculation.*

If ρ is a representation such that $\rho(g)$ is a unitary transformation for every $g \in G$, then ρ is said to be a **unitary** representation. As indicated above, both triv and reg are unitary representations. Furthermore, the direct sum and tensor product of unitary representations are themselves unitary.

Appendix A describes a method of building representations of a group G from those of a normal subgroup $N \lhd G$. Such representations of G are said to be **induced** from those of N. In particular, the appendix describes how the key representations of any supersolvable group can constructed, starting from the trivial representation, by a process of induction and forming tensor products.

2.2.3 Basic Properties

Since any representation ρ is, by definition, a homomorphism, we have $\rho(e) = \mathrm{I}$. This means that $\chi_\rho(e) = \operatorname{tr} \mathrm{I} = n = d_\rho$, the dimension of ρ. Futhermore, for any $g \in G$, we have $\mathrm{I} = \rho(e) = \rho(gg^{-1}) = \rho(g)\rho(g^{-1})$, which shows that $\rho(g^{-1}) = \rho(g)^{-1}$. Since we are working over $\mathbb{C}$, it is always possible to find a change of basis such that $\rho(g)$ is a unitary matrix for every $g \in G$ (see appendix A for details), and, in that basis, we have $\rho(g^{-1}) = \rho(g)^\dagger$. Since characters are unaffected by a change of basis, this means that $\chi_\rho(g^{-1}) = \operatorname{tr}(\rho(g)^\dagger) = \overline{\chi_\rho(g)}$ for any representation ρ.

It also follows immediately that characters are **class functions**, that is, equal on all elements of the same conjugacy class: if $g, x \in G$, then we can see that $\chi_\rho(g^x) = \operatorname{tr}(\rho(g^x)) = \operatorname{tr}(\rho(x^{-1}gx)) = \operatorname{tr}(\rho(x^{-1})\rho(g)\rho(x)) = \operatorname{tr}(\rho(x)^{-1}\rho(g)\rho(x)) = \operatorname{tr}\rho(g) = \chi_\rho(g)$, where we have used the cyclic property of the trace once again. The fact that characters are class functions means that the vector space of characters has dimension no larger than the number of conjugacy classes.

For any group, the most important characters to understand are the *irreducible* ones (see appendix A for a formal definition) as any character can be written as a sum of irreducible characters. Hence, understanding the irreducible characters is sufficient to understand the whole space of characters. In particular, we can show (see appendix A) that the irreducible characters themselves span the space of all class functions and, hence, that the number of distinct irreducible characters is equal to the number of conjugacy classes of the group. In particular, there are only finitely many irreducible characters of any finite group.

The key feature that makes characters easy to work with is that they form an inner product space. In particular, the sesquilinear[10] form

$$\langle \chi_\rho \,|\, \chi_\sigma \rangle := |G|^{-1} \sum_{g \in G} \chi_\rho(g^{-1}) \chi_\sigma(g) = |G|^{-1} \sum_{g \in G} \overline{\chi_\rho(g)} \chi_\sigma(g)$$

defines an inner product on characters. It can be shown (see appendix A) that $\langle \chi_\rho \,|\, \chi_\rho \rangle = 1$ iff ρ is irreducible. Furthermore, if ρ and σ are both irreducible, then $\langle \chi_\rho \,|\, \chi_\sigma \rangle = 1$ if $\chi_\rho \equiv \chi_\sigma$ and 0 otherwise. In other words, the inner product is 1 if ρ and σ are the same representation and 0 otherwise.

Appendix A contains a more thorough review of representation theory, including all of the results that we will need below along with sketches of their proofs.

2.2.4 Examples of Representations of Important Groups

Abelian Groups

We start with the simplest type of abelian groups.

Example 2.11. *Let p be prime and ω_p the principle p-th root of unity. Then, for each $a \in \mathbb{Z}_p$, the map χ_a given by $\chi_a(x) := \omega_p^{ax}$ is an irreducible representation of $\mathbb{Z}_p$.*

We can see that

$$\begin{aligned} \langle \chi_a \,|\, \chi_a \rangle &= |G|^{-1} \sum_{x \in G} \chi_a(x) \chi_a(-x) \\ &= |G|^{-1} \sum_{x \in G} \omega_p^{ax-ax} \\ &= |G|^{-1} \sum_{x \in G} 1 = |G|^{-1}|G| = 1, \end{aligned}$$

which proves that each χ_a is an irreducible representation.

Since $\chi_a(1) \neq \chi_b(1)$ for $a \neq b$, we can see that all p such representations are distinct. Now, we noted above that the number of irreducible representations is equal to the number of conjugacy classes of G. In an abelian group, every element is in its

[10] We use physics conventions: a sesquilinear form $\langle x \,|\, y \rangle$ is linear in y and conjugate linear in x.

own conjugacy class (since $x^y = y^{-1}xy = x$ when the group is abelian). This tells us that $\mathbb{Z}_p$ has p distinct irreducible representations, so the χ_a's are all of them.

To extend to all abelian groups, we will use the following fact.

Theorem 2.12. *If ρ and σ are irreducible representations of A and B, respectively, then the map $(a, b) \mapsto \rho(a) \otimes \sigma(b)$ is an irreducible representation of $A \times B$. Furthermore, every irreducible representation of $A \times B$ is isomorphic to one of this form.*

Proof. The character of this representation is the map $\chi_{\rho,\sigma}$ defined by $\chi_{\rho,\sigma}(a, b) := \chi_\rho(a)\chi_\sigma(b)$. If ρ' and σ' are another pair of representations of A and B, respectively, then a short calculation shows that $\langle \chi_{\rho,\sigma} \,|\, \chi_{\rho',\sigma'} \rangle = \langle \chi_\rho \,|\, \chi_{\rho'} \rangle \langle \chi_\sigma \,|\, \chi_{\sigma'} \rangle$. From this, it follows that $\langle \chi_{\rho,\sigma} \,|\, \chi_{\rho,\sigma} \rangle = 1$, which shows that these representations are irreducible, and that $\langle \chi_{\rho,\sigma} \,|\, \chi_{\rho',\sigma'} \rangle = 0$ if $\rho \ncong \rho'$ or $\sigma \ncong \sigma'$, which shows that these are all distinct. Finally, since the number of conjugacy classes in $A \times B$ is equal to the number of conjugacy classes in A times the number of conjugacy classes in B, it follows that these are all the irreducible representations of $A \times B$. □

By the Fundamental Theorem 2.1, every finite abelian group is isomorphic to $\mathbb{Z}_{p_1^{t_1}} \times \cdots \times \mathbb{Z}_{p_m^{t_m}}$ for some primes $p_1, \ldots, p_m$ and numbers $t_1, \ldots, t_m$. Thus, it follows from the above facts that every irreducible representation is of the form

$$\chi_{a_1,\ldots,a_m}(x_1, \ldots, x_m) := \prod_{i=1}^{m} \omega_{p_i^{t_i}}^{a_i x_i},$$

for some $a_1, \ldots, a_m$ with $a_i \in \mathbb{Z}_{p_i^{t_i}}$ and, furthermore, that every irreducible representation is of this form. (Here, since these representations are 1-dimensional, we do not even need to worry about non-isomorphic irreducible representations. Every irreducible representation must be one of these exactly.)

Note that the elements $a_1, \ldots, a_m$ labelling the representation are themselves an element of the group $\mathbb{Z}_{p_1^{t_1}} \times \cdots \times \mathbb{Z}_{p_m^{t_m}}$. Hence, this also shows that the irreducible representations of every abelian group are in simple 1-to-1 correspondence with the elements of that group.

Supersolvable Groups

We return to our above example of supersolvable groups: the dihedral groups.

Recall that the dihedral group of the form $\mathbb{Z}_p \rtimes \mathbb{Z}_2$ has a normal abelian subgroup $H \lhd G$ isomorphic to $\mathbb{Z}_p$. H consists of the elements of the form $(x, 0)$ for any $x \in \mathbb{Z}_p$. If we take the quotient by this group, then we are left with a group isomorphic to $\mathbb{Z}_2$: its two elements are the cosets $(\mathbb{Z}_p, 0)$ and $(\mathbb{Z}_p, 1)$.

Example 2.13. *Let $\pi : D_{2p} \to \mathbb{Z}_2$ be the projection into the second component of $D_{2p} = \mathbb{Z}_p \rtimes \mathbb{Z}_2$. If χ_b, for $b \in \mathbb{Z}_2$, is an irreducible representation of $\mathbb{Z}_2$, then $\chi_b \circ \pi$ is an irreducible representation of D_{2p}.*

It is a general fact that, if $\varphi : G \to K$ is a homomorphism and ρ is a representation of K, then $\rho \circ \varphi$ is a representation of G simply because the composition of two homomorphisms is a homomorphism. Another general fact is that any 1-dimensional representation, like this, is irreducible since it cannot be decomposed into representations of any smaller dimension. Hence, these are indeed irreducible representations.

Example 2.14. *For each $a \in \mathbb{Z}_p$ with $0 < a \leq \frac{1}{2}(p-1)$, the map*

$$\rho_a((x,y)) = \begin{pmatrix} \omega_p^{ax} & 0 \\ 0 & \omega_p^{-ax} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^y$$

is an irreducible representation of D_{2p}. More explicitly, we have

$$\rho_a((x,0)) = \begin{pmatrix} \omega_p^{ax} & 0 \\ 0 & \omega_p^{-ax} \end{pmatrix} \qquad \text{and} \qquad \rho_a((x,1)) = \begin{pmatrix} 0 & \omega_p^{ax} \\ \omega_p^{-ax} & 0 \end{pmatrix}.$$

Note that replacing a by $-a$ gives a representation with an identical character (since $\chi_{\rho_a}((x,0)) = \omega_p^{ax} + \omega_p^{-ax}$ and $\chi_{\rho_a}((x,1)) = 0$), so we get all the distinct irreducible representations just by taking $a \leq \frac{1}{2}(p-1)$.

We can check most easily that this is irreducible by looking at the character χ_{ρ_a}.

In paricular, we can calculate

$$\begin{aligned}\langle \chi_{\rho_a} \,|\, \chi_{\rho_b} \rangle &= \tfrac{1}{2p} \sum_{x \in Z_p} (\omega_p^{ax} + \omega_p^{-ax})(\omega_p^{-bx} + \omega_p^{bx}) \\ &= \tfrac{1}{2p}(p\delta_{a,b} + p\delta_{a,-b} + p\delta_{-a,b} + p\delta_{-a,-b}) = \delta_{a,b},\end{aligned}$$

verifying that the representations are distinct and irreducible.[11]

To check that these are all the irreducible representations, it is easiest to use the fact, from Proposition A.12, that the sum of the squared degrees of the irreducible representations is equal to $|G|$. In this case, this sum is $2 \cdot 1^2 + \frac{1}{2}(p-1) \cdot 2^2 = 2p$, so these are all the irreducible representations.

These irreducible representations of dihedral groups and the nilpotent groups we consider in the next section can be derived using Wigner and Mackey's "method of little groups" [57], which is a special case of the method described in section A.5.[12] However, as we just saw, if we are handed the irreducible representations, it is much simpler to verify that they are representations by checking that they are homomorphisms; verifying that they are irreducible, by checking the inner product of their characters with themselves; and verifying that these are all the irreducible representations, by comparing the sum of the squared degrees to the size of the group.

Nilpotent Groups

Let us start with our simplest example of a nilpotent group.

Example 2.15. *The Heisenberg group (over $\mathbb{Z}_p$) has two classes of irreducible repre-*

[11] Although we will not need it here, it is not hard to check, using the machinery discussed in Appendix A, that this representation is $\mathrm{Ind}_H^G \chi_a$. Indeed, just the fact that χ_{ρ_a} is zero outside of the normal subgroup H strongly suggests that ρ_a is induced.

[12] The method of little groups applies to semidirect products. It is not hard to check that the group of $k \times k$ unitriangular matrices is the semidirect product of the normal, abelian subgroup of matrices nonzero entries only in the right-most column and the quotient by this subgroup. Furthermore, the latter quotient is isomorphic to the group of $(k-1) \times (k-1)$ unitriangular matrices. Thus, it is possible, at least in principle, to inductively construct the representations of $k \times k$ unitriangular matrices, for all k, using just the method of little groups.

sentations. First, it has 1-dimensional representations of the form

$$\chi_{a,b}((x,y,z)) := \omega_p^{ax+bz}$$

for any $a, b \in \mathbb{Z}_p$. As with the dihedral group, we can recognize this as the composition of the homomorphism that quotients out the center with a character of the abelian group $\mathbb{Z}_p^2$ that remains. (Recall that, since the Heisenberg group is 2-nilpotent, taking the quotient with the center must leave an abelian group.) As we noted above, this must be an irreducible representation since it is a homomorphism and 1-dimensional.

The Heisenberg group also has p-dimensional representations. For each $\kappa \in \mathbb{Z}_p$ with $\kappa \neq 0$, there is one of the form[13]

$$\sigma_\kappa((x,y,z)) := \omega^{\kappa y} \sum_{r \in \mathbb{Z}_p} \omega^{-\kappa z(r+x)} |r + x\rangle\langle r|.$$

As usual, we can check that these are irreducible by verifying that $\langle \chi_{\sigma_\kappa} \mid \chi_{\sigma_\kappa} \rangle = 1$.

We can check that these are all the representations, as we did above, by summing the squared degrees of the representations. In this case, that gives us $p^2 \cdot 1 + (p-1) \cdot p^2 = p^3 = |\mathbb{Z}_p^2 \rtimes \mathbb{Z}_p|$, which confirms that we have found all the irreducible representations.

As we discussed above, the Heisenberg group can be generalized into k-nilpotent groups for $k > 2$. Here, we look at the representations of the 3-nilpotent version.

Example 2.16. *The group of 4×4 unitriangular matrices over $\mathbb{Z}_p$ has four classes of irreducible representations. To define these, let us label the entries of a matrix as*

$$g := \begin{pmatrix} 1 & u & v & x \\ 0 & 1 & w & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

[13] Note that, since $r, x \in \mathbb{Z}_p$, the expression "$r + x$" means addition modulo p. This applies to the next example as well.

Then, for $\alpha, \gamma, \zeta \in \mathbb{Z}_p$, there is a 1-dimensional representation

$$\chi_{\alpha,\gamma,\zeta}(g) := \omega_p^{\alpha u+\gamma w+\zeta z}.$$

There are two classes of p-dimensional irreducible representations. First, for $\beta, \zeta \in \mathbb{Z}_p$ with $\beta \neq 0$, there is a representation

$$\sigma_{\beta,\zeta}(g) := \omega_p^{\beta v+\zeta z} \sum_{r\in\mathbb{Z}_p} \omega_p^{-\beta w(r+x)} |r+u\rangle\langle u|.$$

Second, for $\alpha, \beta, \epsilon \in \mathbb{Z}_p$ with $\epsilon \neq 0$, there is another representation

$$\xi_{\alpha,\beta,\epsilon}(g) := \omega_p^{\alpha u+\beta v+\epsilon z} \sum_{r\in\mathbb{Z}_p} \omega_p^{\beta ur-\epsilon z(r+w)} |r+w\rangle\langle u|.$$

Lastly, for $\gamma, \delta \in \mathbb{Z}_p$ with $\delta \neq 0$, there is a p^2-dimensional representation

$$\rho_{\gamma,\delta}(g) := \omega_p^{\gamma w+\delta x} \sum_{r,s\in\mathbb{Z}_p} \omega_p^{r\delta y+s\delta z} |r,s\rangle\langle r+u, s+v+rw|.$$

As above, we can check that each of these is a representation by verifying that it is a homomorphism, that it is irreducible by verifying that the inner product of character with itself is 1, and that these are all the irreducible representations, by verifying that sum of the squared degrees is equal to the size of the group.

In this last example, the representations are becoming increasingly complicated. However, as shown in Theorem A.24, every irreducible representation of a nilpotent group can actually be written in a very simple form by using induced representations.

Induced representations are defined formally in section A.3. Informally, the idea is to take a representation ρ of a subgroup $H \leq G$ and extend it to all of G by making it act like the regular representation of G/H on cosets of H while still acting like ρ on H itself. The resulting representation is denoted $\operatorname{Ind}_H^G \rho$.

Theorem A.24 says that every irreducible representation of a nilpotent group G is of the form $\operatorname{Ind}_H^G \varphi$, where $H \leq G$ is a subgroup of G containing $Z(G)$ and φ is a

1-dimensional representation of H. Let us now describe, in detail, the representation $\mathrm{Ind}_K^G \varphi$ when φ is 1-dimensional and $K \lhd G$ is *normal.*[14]

The underlying vector space of $\mathrm{Ind}_K^G \varphi$ has one copy of V_φ for each coset of K. Since φ is 1-dimensional, each V_φ is just a 1-dimensional space, so we can take $|xK\rangle$, for $xK \in G/K$ as a basis for $V_{\mathrm{Ind}_K^G \varphi}$.

For elements $k \in K$, action of $\mathrm{Ind}_K^G \varphi$ on the subpace spanned by $|eK\rangle$ will be just $\varphi(k)$, so in this subspace, it is exactly the representation φ. More generally, on the subspaced spanned by $|xK\rangle$, the action of $k \in K$ will be $\varphi(k^x)$.

The overall action of $\mathrm{Ind}_K^G \varphi$ combines permutation on different cosets of K with the action just described for elements of K. In detail, if we let $x_1K, \ldots, x_mK$ be the cosets of K, then every $g \in G$ is of the form x_ik for some $i \in \{1, \ldots, m\}$ and $k \in K$. In that case, the action of g can be written as

$$\begin{aligned}(\mathrm{Ind}_K^G \varphi)(x_ik) &:= \sum_{j=1}^{m} \varphi(x_{t(i,j)}^{-1}(x_ik)x_j)\ |x_ix_jK\rangle\langle x_jK| \\ &= \sum_{j=1}^{m} \varphi(x_{t(i,j)}^{-1}x_ix_j)\varphi(k^{x_j})\ |x_ix_jK\rangle\langle x_jK|,\end{aligned}$$

where $t(i,j)$ is the index of the representative of the coset x_ix_jK.

Ignoring the scale factors, we can see that $(\mathrm{Ind}_K^G \varphi)(g)$ simply permutes the cosets as in the regular representation of G/K. In addition, however, there are two scalar factors introduced. The first, $\varphi(k^{x_j})$, matches our above description of how $\mathrm{Ind}_K^G \varphi$ acts on $k \in K$. This part is independent of how we choose the representatives $\{x_iK\}$ of the cosets of K. The second factor, $\varphi(x_{t(i,j)}^{-1}x_ix_j)$, is seemingly a pure function of how we chose representatives. We will see later on, when we study group cohomology, that these factors are also independent of the representatives chosen. In fact, the set of numbers of the form $x_{t(i,j)}^{-1}x_ix_j$ is an important quantity characterizing the group.[15]

Let us now see how each of the representations of nilpotent groups we saw above

[14] We will not always have H normal in G even if G is nilpotent, as we will see in a moment. However, this assumption simplifies the description significantly.

[15] In particular, as we will see later on, we always have $x_{t(i,j)}^{-1}x_ix_j = e$ in a semidirect product, which makes their induced representations especially easy to identify (and to construct using the method of little groups).

can be described in these terms.

Example 2.17. *Of the representations of the Heisenberg group, those of the form $\chi_{a,b}$ are already of the form described in the theorem: we can write these as $\operatorname{Ind}_G^G \chi_{a,b}$ since $\chi_{a,b}$ is already 1-dimensional and G/G is the trivial group, so inducing from G to G does not change the representation.*

Let $A := \{(0, y, z) \mid y, z \in \mathbb{Z}_p\}$. Then A is an abelian, normal subgroup. We can write the other representations as $\sigma_\kappa \cong \operatorname{Ind}_A^G \chi_\kappa$, where χ_κ is the 1-dimensional representation of A defined by $\chi_\kappa(0, y, z) := \omega_p^{\kappa y}$.

Cosets of A are labeled simply by the x coordinate of (x, y, z). We can see that σ_κ permutes these cosets as we would expect in an induced representation. Furthermore, we can see that $(xx', 0, 0)^{-1}(x, 0, 0)(x', 0, 0) = (0, 0, 0)$ for any $x, x' \in \mathbb{Z}_p$, so the induced representation only has only scalar factors of the form $\chi_\kappa((0, y, z)^{(r,0,0)}) = \chi_\kappa((0, y - rz, z)) = \omega_p^{\kappa y - \kappa r z}$. Thus, $\operatorname{Ind}_A^G \chi_\kappa$ matches the definition of σ_κ above.

From the theorem, we know that this must also be true for the higher nilpotent generalizations of the Heisenberg group.

Example 2.18. *The irreducible representations of the 4×4 unitriangular matrices can be written as follows:*

$$\begin{aligned}
\chi_{\alpha,\gamma,\zeta} &= \operatorname{Ind}_G^G \varphi_{\alpha,\gamma,\zeta} & \textit{with} \quad \varphi_{\alpha,\gamma,\zeta}(g) &:= \omega_p^{\alpha u + \gamma w + \zeta z} \\
\sigma_{\beta,\zeta} &= \operatorname{Ind}_{G_{u=0}}^G \varphi_{\beta,\zeta} & \textit{with} \quad \varphi_{\beta,\zeta}(g) &:= \omega_p^{\beta v + \zeta z} \\
\xi_{\alpha,\beta,\epsilon} &= \operatorname{Ind}_{G_{w=0}}^G \varphi_{\alpha,\beta,\epsilon} & \textit{with} \quad \varphi_{\alpha,\beta,\epsilon}(g) &:= \omega_p^{\alpha u + \beta v + \epsilon y} \\
\rho_{\gamma,\delta} &= \operatorname{Ind}_{G_{u=v=0}}^G \varphi_{\gamma,\delta} & \textit{with} \quad \varphi_{\gamma,\delta}(g) &:= \omega_p^{\gamma w + \delta x},
\end{aligned}$$

where $G_{u=0}$ is the set of matrices with element u being zero and $G_{w=0}$ and $G_{u=v=0}$ defined analogously.[16] *In this case, the representations described this way are not all identical to those given above. However, they have identical characters, which tells us that they are isomorphic.*

[16] Note that this last subgroup, $G_{u=v=0}$, is not normal in G.

Chapter 3

Quantum Computation

In this chapter, we will review the basics of quantum computation. We will also discuss some prior work that relates to problems discussed later on. Finally, we will make an effort to develop tools that allow us to work, wherever possible, at a higher level than vector spaces and matrices in subsequent chapters.

Unless otherwise stated, all of the foundational material discussed here can be found in [51].

3.1 Quantum Mechanics

For us, a finite-dimensional quantum mechanical system will be a finite-dimensional (complex) vector space V with a distinguished orthonormal basis, which makes it into a Hilbert space using the standard sesquilinear inner product. We denote the elements of this basis by $|0\rangle, \ldots, |N-1\rangle$, where N is the dimension of the space.

A **state** of the system is simply a normalized vector $|\psi\rangle \in V$. (That is, a vector satisfying $\langle \psi \mid \psi \rangle = 1$.) Those states in the distinguished basis are **classical**, as they normally correspond to states of some ordinary (non-quantum) system.

A quantum mechanical system operates under the following rules[1]:

Preparation The system can be started in any classical state.

[1] Here, we focus only on discrete-time quantum mechanical systems. Continuous-time systems replace the evolution rule below with the Schrödinger equation.

Evolution At each step, the state can be changed from $|\psi\rangle$ to $U|\psi\rangle$, where $U \in \mathrm{U}(V)$ is any unitary transformation.

Measurement Alternatively, the state can be **measured**, which takes state $|\psi\rangle$ to the classical state $|i\rangle$ with probability $|\langle i \,|\, \psi\rangle|^2$. (Normalization ensures that these probabilities sum to unity.) The **outcome** of the measurement is the label i of the classical state. Unlike the intermediate states of the system, measurement outcomes are visible to outside observers.

Operation according to the above rules is termed **physical**. In particular, a non-unitary transformation A is sometimes called **non-physical** since evolution according to this transformation would not be physical.

It is worth noting that operation staying entirely in the classical states is always physical. In particular, transformations that simply permute the classical states are always physical since permutations are unitary transformations.

Later on, we will also consider quantum systems that operate on vector spaces of the form $V \otimes W$ and perform measurements on only part of such states. By the principle of deferred measurement [51], operation using intermediate measurements is always equivalent to some operation that only performs measurement at the end, where the entire state is measured. Hence, we can allow both partial measurements and intermediate measurements without loss of generality.

If we have a state $|\psi\rangle \in V \otimes W$, the probability of measuring label i on the first part is given by $|(\langle i| \otimes \mathrm{I})|\psi\rangle|^2$. We can see that this generalizes the case above. The state remaining after measuring label i is $(\langle i| \otimes \mathrm{I})|\psi\rangle$ scaled to have squared norm 1.

3.1.1 Efficient Operation

Physical operation describes that which is physically possible, in principle. However, we are normally interested in operation that is efficient, meaning (loosely) that it could be carried out with reasonable time and resources.

The building blocks of efficient operation are constant-sized quantum systems. A quantum system with basis labeled by elements of $\mathbb{Z}_2$ is referred to as a **qubit**, and more generally, a system with basis labeled by elements of $\mathbb{Z}_d$ is a **qudit**. It is fair to assume that preparation, evolution, and measurement of qudits with $d = O(1)$ can be performed in constant time. (See [51] for a discussion of actual physical systems, such as ion traps, that behave like qubits.)

By forming the tensor product of n qubits, we can construct a 2^n-dimensional quantum mechanical system $V \cong \mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$. We will think of each qubit as one unit of space. Hence, the joint system represents n units of space, even though the dimension of the underlying system is exponentially larger.

As noted above, we can assume that a unitary applied to only a single qubit can be performed in constant time. Hence, transformations of the form $\cdots \otimes \mathrm{I} \otimes U \otimes \mathrm{I} \otimes \cdots$, which act nontrivially on only a single qubit, take constant time on the joint system. Moreover, since the tensor product of $C = O(1)$ qubits is isomorphic to a qudit with $d = 2^C = O(1)$, we may likewise apply transformations affecting only a constant number of qubits in constant time.

In summary, we will think of our quantum mechanical systems as formed from the tensor product of m qubits (or qudits with $d = O(1)$), which represents m units of space. We will assume the ability to perform arbitrary unitary transformations on $O(1)$ qubits at a time, with each taking constant time. All other transformations must consist of a sequence of such transformations, which requires time proportional to the length of the sequence.

Following the usual computer science conventions, we will say that the operation of a quantum mechanical system is **efficient** if it operates using $\mathrm{poly}(n)$ space and its transformations require $\mathrm{poly}(n)$ time, where n is a parameter describing the size of the problem in question, which will be problem-specific.

3.1.2 Example Transformations

Let us start with unitary transformations of individual qubits, operating on the vector space $\mathbb{C}^2$. Some of the most important examples are

$$\mathrm{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \ D_\phi = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{pmatrix}, \text{ and } H = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

The first, NOT, is a classical transformation as it permutes the classical states. The second, a diagonal matrix with phase ϕ, is purely quantum in the sense that it has no measurable affects on classical states.[2] The third, called a Hadamard transformation, takes the classical states into uniform superpositions. This is a critical component in many quantum algorithms including, e.g., the solution to Simon's problem [59].

Transformations of two qubits operate on the space $\mathbb{C}^2 \otimes \mathbb{C}^2 \cong \mathbb{C}^4$. The most important such transformation is

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix},$$

called a controlled NOT, which applies a NOT transformation to the second qubit if the first qubit is in state $|1\rangle$ and leaves it unchanged if the first qubit is in state $|0\rangle$.

Surprisingly, this small set of examples already contains the full power of quantum computation. In fact, just the set of three transformations, H, $D_{\pi/4}$, and CNOT, is already **universal**, meaning that an arbitrary unitary transformation can be approximated to any accuracy by a sequence of transformations from this set [51]. Replacing $D_{\pi/4}$ by the even simpler $D_{\pi/2}$ transformation still gives a universal set [15]. However, replacing $D_{\pi/4}$ by D_π gives a set of transformations that can be efficiently simulated by a classical computer [30].

Above, we defined a qudit as a vector space with basis labeled by elements of the

[2] Note that replacing $|\psi\rangle$ by $e^{i\phi}|\psi\rangle$ has no effect on the probability of measuring state $|i\rangle$ since $|\langle i|e^{i\phi}|\psi\rangle|^2 = |e^{i\phi}|^2|\langle i \mid \psi\rangle|^2 = |\langle i \mid \psi\rangle|^2$.

group $\mathbb{Z}_d$. More generally, for any group G, we can define a vector space with basis labeled by elements of the group, $V_G := \text{span}\{|g\rangle \mid g \in G\}$. For each $g \in G$, we can define a transformation

$$X_G(g) := \sum_{h \in G} |gh\rangle\langle h|.$$

This is simply the regular representation of G. Since it permutes the basis states, this is a classical transformation.

Next, if ρ is a representation of G, then its character χ_ρ gives rise to a transformation of V_G defined by

$$Z_G(\chi_\rho) := \sum_{h \in H} \chi_\rho(h)|h\rangle\langle h|.$$

This transformation is only physical if $|\chi_\rho(g)| = 1$ for all $g \in G$. The latter holds for all representations of abelian groups. More generally, it holds whenever ρ is a 1-dimensional representation of G.

Two of the transformations we met above are special cases of these constructions. Specifically, the NOT transformation is the special case $\text{NOT} = X_{\mathbb{Z}_2}(1)$ and the diagonal π-phase transformation is $D_\pi = Z_{\mathbb{Z}_2}(\chi_{\text{sign}})$, where $\text{sign} : \mathbb{Z}_2 \to \mathbb{C}$ is the sign representation defined by $\text{sign}(a) := (-1)^a$.

If $\alpha : G \to G$ is an automorphism of the group G, then we can define

$$U_\alpha := \sum_{h \in H} |\alpha(h)\rangle\langle h|.$$

Like $X_G(g)$, this is a classical transformation that simply permutes the classical states.

We have also seen a special case of this construction above, not for the group $\mathbb{Z}_2$ but rather the group $\mathbb{Z}_2 \times \mathbb{Z}_2 = \mathbb{Z}_2^2$. The vector space for the latter group is isomorphic to $\mathbb{C}^2 \otimes \mathbb{C}^2$. If we define the map $\alpha : \mathbb{Z}_2^2 \to \mathbb{Z}_2^2$ defined by $\alpha(a, b) = (a, a + b)$, which is an automorphism, then the transformation U_α is the CNOT operation we saw above.

Finally, the most important transformation we will define for a general group is the **Quantum Fourier Transform** (QFT). If G is abelian, then this is defined by

$$\mathcal{F}_G := \frac{1}{\sqrt{|G|}} \sum_{g \in G} \sum_{\chi \in \widehat{G}} \chi(g) |\rho\rangle\langle g|,$$

where $\widehat{G}$ denotes the set of irreducible representations of G. The orthonormality of characters under their inner product implies that this is a unitary operation.

We have also met a special case of this operation above. For $\mathbb{Z}_2$, the QFT is just the Hadamard transformation, $\mathcal{F}_{\mathbb{Z}_2} \cong H$, if we choose our basis of representations such that $|\text{triv}\rangle = |0\rangle$ and $|\text{sign}\rangle = |1\rangle$.

It is worth noting that the QFT lies at the heart of all of the algorithms we will see in this thesis. It the key element in the algorithms for solving all of the problems on abelian groups that we will discuss below. Furthermore, when we study abelian objects beyond groups, in a later chapter, we will again see that a QFT for those objects remains the essential element of our algorithm. Hence, all of the quantum speedups demonstrated in this work derived from use of a QFT, either directly or via our reductions to problems on abelian groups.

3.1.3 Approximate Transformations

One natural worry with the above formalism is that actual physical systems intended to behave like qubits will not, even in a carefully controlled environment, carry out the unitary transformations exactly (to infinite precision). The best we can hope for is that they carry out the desired transformations *approximately.*

Indeed, most physical implementations of qubits come with only a small set of supported transformations, and other transformations must be approximated by a sequence of those. While such approximations may be sufficient for experiments with individual qubits or pairs of qubits, it is natural to worry that the errors could blow up as we work with the large systems needed to demonstrate a substantial advantage of quantum systems over classical ones.

Fortunately, this fear is unnecessary, as we can show (following [51]) that errors do not accumulate beyond control. To state this precisely, let us first define a measure of how well one unitary transformation $U \in \mathrm{U}(V)$ approximates another transformation $W \in \mathrm{U}(V)$. We define the error of approximation by

$$E(U, W) := \max \|(U - W)|\psi\rangle\| \text{ over } |\psi\rangle \in V \text{ with } \langle\psi \mid \psi\rangle = 1.$$

We will prove that $E(U_n \ldots U_1, W_n \ldots W_1) \leq \sum_{i=1}^{m} E(U_i, W_i)$ by induction on n. There is nothing to prove for $n = 1$. For $n > 1$, let $\tilde{U} = U_n \ldots U_2$ and $\tilde{W} = W_n \ldots W_2$. Then, we can see that

$$\begin{aligned}
\|(\tilde{U}U_1 - \tilde{W}W_1)|\psi\rangle\| &= \|(\tilde{U}U_1 + (-\tilde{W}U_1 + \tilde{W}U_1) + \tilde{W}W_1)|\psi\rangle\| \\
&= \|((\tilde{U} - \tilde{W})U_1 + \tilde{W}(U_1 - W_1))|\psi\rangle\| \\
&\leq \|(\tilde{U} - \tilde{W})U_1|\psi\rangle\| + \|\tilde{W}(U_1 - W_1)|\psi\rangle\| \\
&\leq E(\tilde{U}, \tilde{W}) + E(U_1, W_1) \\
&\leq \textstyle\sum_{i=1}^{n} E(U_i, W_i)
\end{aligned}$$

by the induction hypothesis. This bound holds for all $|\psi\rangle$; hence, taking the maximum over all choices of $|\psi\rangle$ gives the desired result.

This "union bound" for quantum systems shows that we can limit the error to ϵ for a sequence of n transformations provided that we limit the error to ϵ/n on each individual transformation in the sequence. The latter requirement is rarely onerous.

In the particular case of error resulting from approximating arbitrary unitary transformations by those from universal set, it is possible to show that this can be accomplished with only a polynomial increase in the time required [51]. In fact, the Solovay-Kitaev theorem [51] shows that we can achieve n/ϵ error using only $\operatorname{poly}\log(n/\epsilon)$ transformations from our universal set, thus, increasing the total running by at most a $\operatorname{poly}\log(n/\epsilon)$ factor.

In other words, if the operation of our quantum mechanical system is efficient in the original model above, then it is also efficient in a more realistic model that accounts for the limited accuracy and limited set of natively supported transformations of actual physical implementations.[3]

3.2 Group Problems in Quantum Computation

3.2.1 The Hidden Subgroup Problem

Two of the most important early successes of quantum algorithms were the results of Simon [59] and Shor [58]. Simon proved a separation between what can be computed efficiently on quantum and classical computers in the presence of a certain oracle by demonstrating a problem (now called "Simon's problem") that can be solved efficiently on a quantum computer but not a classical one. Shor proved that quantum computers can efficiently factor integers and compute discrete logarithms, both of which are widely believed to be intractable on classical computers. (In particular, the security of notable cryptographic protocols such as RSA and Diffie-Hellman are based on these assumptions [17].) These groundbreaking results caused great excitement and led to the rapid development of the field of quantum computation.

Much of the early work sought to understand generalize these results. Brassard and Høyer [16, 35] showed that Simon's problem and the discrete logarithm problem are both special cases of a more general problem about finite groups, later termed the **Hidden Subgroup Problem** [49]. (See also [31, 13, 20] for contemporaneous efforts.) This problem will be important to much of what we discuss in this thesis, so let us now define it formally.

In the Hidden Subgroup Problem (HSP) over a group G, we are given (as an oracle) a function $f : G \to \{0,1\}^m$ that is promised to be constant on cosets of some subgroup $H \leq G$. That is, we have $f(g) = f(g')$ iff $g' = gh$ for some $h \in H$. The function f is called the **hiding function** as it hides the subgroup H. The goal in this problem is to identify the H and output a generating set for it.

We will say that an algorithm solves the hidden subgroup problem efficiently for

[3] We will ignore entirely the important question of how to compute effectively when uncontrolled errors can affect the quantum system. See [51] for a discussion of techniques in quantum error correction designed to address this problem.

an (infinite) class of groups $\mathcal{G}$ if it runs in time polynomial in $\log |G|$ for each $G \in \mathcal{G}$. Brassard and Høyer [16, 35] showed that the techniques developed by Shor can be extended to efficiently solve the HSP for the class of finite abelian groups. We will see a generalization of their result below.

First, we consider a variant of the HSP, termed the Hidden Kernel Problem (HKP) [10], that will also be important for us later on. In this variant, the hiding function f is a homomorphism $G \to K$ for some group K. Here, rather than being an opaque label in the set $\{0,1\}^m$, as in the HSP, the value $f(g)$ lies in a given group K. The requirement that f is a homomorphisms means that we cannot, when constructing f, permute the label space $\{0,1\}^m$ arbitrarily so that non-identical labels provide no information. In particular, Simon's problem is not an instance of the HKP.

Nonetheless, the HKP is still apparently hard for a classical computer. For example, the discrete logarithm problem uses a hiding function that is a homomorphism of the form $f : \mathbb{Z}_{p-1}^2 \to \mathbb{Z}_p^\times$. Hence, the HKP is still hard under standard cryptographic assumptions, despite requiring the hiding function to be a homomorphism.

Below, we will see how to efficiently solve the abelian HSP (and the HSK) on a quantum computer. But first, we look at some other important group problems that can be solved efficiently on a quantum computer.

3.2.2 Order Finding

Above, we mentioned Shor's algorithm for computing discrete logarithms but not his celebrated algorithm for integer factoring. That algorithm reduces integer factoring to the problem of computing the order of an element in a finite group. That is, given an element $x \in G$, we wish to find the smallest $n \in \mathbb{Z}$ such that $x^n = e$.

Order finding is actually a special case of the HSP (or HKP) considered above. However, it fits somewhat awkwardly into this framework.

To see why, let us consider how we would construct a hiding function that reveals the order of $x \in G$. The most natural approach would be to define $f : \mathbb{Z}_{|G|} \to G$ by $f(t) = x^t$. If our algorithm is going to invoke this f, then we should assume that we know its domain $\mathbb{Z}_{|G|}$, which means that we should know $|G|$. However, for integer factoring, where $G = \mathbb{Z}_N^\times$, simply knowing $|G|$ is enough to classically factor $N = pq$ into the primes p and q; hence, there is no way to perform this reduction efficiently.

One way to get around the problem is to instead define the hiding function as $f : \mathbb{Z} \to G$, working with the infinite abelian group $\mathbb{Z}$. This method works. In fact, it is possible to view Shor's algorithm as a clever discrete approximation to the standard abelian HSP algorithms (discussed below) applied to $\mathbb{Z}$ [11].

The above method works because it allows us to avoid revealing $\mathbb{Z}_{|G|}$. However, it also has the cost of forcing us to deal with infinite groups. Fortunately, there is an alternative way of accomplishing the former without accepting the latter. Namely, we can specify G to the algorithm as a black-box group [4].

As the name suggests, a black-box group does not reveal the group itself. Elements of the group are represented by opaque labels, and the algorithm is given only (1) the label of the identity element, (2) labels of a generating set for the group, (3) an oracle that takes the labels of two elements and returns the label of their product, and (4) an oracle that takes the label of an element and returns the label of its inverse. (Note that we are requiring that elements of the group have *unique* labels.[4])

This is sufficient for integer factoring since we can efficiently multiply elements from $\mathbb{Z}_N^\times$ and compute their inverses on a classical computer without ever having to factor N. Thus, we will define the Order Finding Problem to take as input a black-box group G and the label of an element $x \in G$ and to return as output the smallest $n \in \mathbb{Z}$ such that $x^n = e$.[5]

It follows from the work of Kitaev [40] that the Order Finding Problem can be solved efficiently on a quantum computer for an arbitrary abelian group. The techniques involved are analogous to some of those we will discuss below, but working with the infinite group $\mathbb{Z}$ (which should probably not surprise us per the discussion above). As we are interested only in finite groups for the main results, we will not have a need to look at these in any detail.

[4] This is not assumed by some classical algorithms, but it is usually required in the quantum setting since we want to use these labels as the basis for our Hilbert space.

[5] It is worth noting that we can still efficiently compute (the label of) x^n in the black-box model: we can use the method of repeated squaring to compute x^n with $O(\log n)$ oracle calls instead of $n - 1$ calls of a naive evaluation.

It is worth noting that these method work even for *non-abelian* groups. This is because the algorithm only works with the subgroup $\langle x \rangle$, which is abelian even when G is not. Hence, Kitaev's work demonstrates that we can compute the order of elements in *arbitrary* black-box groups.

In fact, order finding is only one of many problems about black-box groups that can be solved efficiently on quantum computers. Babai and Beals [8] showed that many problems about black-box groups could be solved classically assuming access to oracles for factoring, discrete logarithm, and the constructive membership problem on elementary abelian groups[6]. Since quantum computers can solve these problems efficiently, all of these problems are solvable quantum computers.

Theorem 3.1 (adapted from [36]). *Let G be a finite* solvable *black-box group.*[7] *Then there are efficient quantum algorithms to compute each of the following:*

1. *Constructive membership testing for (subgroups of) G.*

2. *Computing the order of G.*

3. *Find generators for the center of G.*

4. *Construct a composition series for G with nice representations of the factors.*

5. *Find generators for the Sylow subgroups of G*

As noted above, we will assume that we have a unique label for each element of G. This poses a problem for us only when we wish to work with the group G/N, for a normal subgroup $N \lhd G$, as it is not obvious, in that case, how to choose unique labels for cosets $xN \in G/N$ since any $x' = xn$, for $n \in N$, gives rise to the same coset. However, the classical techniques of [8] combined with the quantum techniques of [62] allow this case to be handled as well.

[6] This means finding whether a given element is in the (sub)group and, if so, showing how it is expressed in terms of the generators of that group.

[7] All of this generalizes to non-solvable groups but with running time that depends in a more complicated fashion on the structure of the non-abelian factors of the group, so we will focus, here, on the simpler case of solvable groups.

Corollary 3.2 (from [36]). *Let G be a finite solvable black-box group and $N \lhd G$ a normal subgroup. Then each of the above problems can also be solved for G/N.*

Using the ability to compute the center recursively gives us the following corollary:

Corollary 3.3. *Let G be a finite* nilpotent *black-box group. Then there is an efficient quantum algorithm finding a generating set $\{g_{i,j}\}_{i=1}^{k}$ such that the $g_{i,*}$'s generate $Z_i(G)/Z_{i-1}(G)$ for each $i \in \{1, \dots, k\}$.*

3.2.3 Abelian Group Decomposition

Suppose that G is an *abelian* group given to us as a black-box. By Theorem 2.1, we know that $G \cong \mathbb{Z}_{p_1^{t_1}} \times \cdots \times \mathbb{Z}_{p_m^{t_m}}$ for some primes $p_1, \dots, p_m$ and $t_1, \dots, t_m \in \mathbb{Z}$. The goal of **abelian group decomposition** is to determine how to describe G in this manner.

Specifically, in addition to the p_i's and t_i's, we want to find a set of generators $h_1, \dots, h_m$ such that h_i generates the i-th part in this decomposition, i.e., $\langle h_i \rangle \cong \mathbb{Z}_{p_i^{t_i}}$. Furthermore, we want to be able to easily switch between these new generators and the original generators, $g_1, \dots, g_m$, that came with description of the black-box group G. Specifically, for each $1 \leq i \leq m$, we want numbers $a_{i,j}$ and $b_{i,j}$, for $1 \leq j \leq m$, such that $h_i = g_1^{a_{i,1}} \cdots g_m^{a_{i,m}}$ and $g_i = h_1^{b_{i_1}} \cdots h_m^{b_{i,m}}$. With these, we can efficiently translate (by substitution) any expression involving the h_i's into one involving the g_i's and vice versa.

Cheung and Mosca [18], followed by Bermejo-Vega, Lin, and Van den Nest [10], showed that this problem can solved efficiently on a quantum computer. Their method works by reducing abelian group decomposition to the two problems we discussed above, the HKP and order finding.

We start by computing the order, d_i, of each generator g_i. Then we consider the homomorphism $f : Z_{d_1} \times \cdots \times Z_{d_m} \to G$ defined by $f(x_1, \dots, x_m) = g_1^{x_1} \cdots g_m^{x_m}$. This function can be computed efficiently using efficient multiplication in the group G. We can then solve the HKP to find the subgroup $K = \operatorname{Ker} f$.

At this point, we know that $G \cong Z_{d_1} \times \cdots \times Z_{d_m}/K$ since f is surjective.[8] Cheung and Mosca showed that we can then find the desired generating set $h_1, \ldots, h_m$, with $\langle h_i \rangle \cong Z_{p_i^{t_i}}$, and the $a_{i,j}$'s that relate the h_i's to the original generators using just *classical* post processing. Bermejo-Vega, Lin, and Van den Nest later showed that additional classical post processing can identify the $b_{i,j}$'s that relate the g_i's to the h_i's. All of this classical post processing runs in time polynomial in $\log |G|$.

As one simple application, we can use abelian group decomposition to compute the size of the group: the size of the group is simply the product of the orders of the h_i's above, which can also be computed efficiently on a quantum computer, as we saw earlier. A careful analysis these algorithms [48, 66] gives the following result:

Theorem 3.4. *Let G be a black-box abelian group. If G is specified by a list of L generators and every element in G has order at most M, then there is a quantum algorithm that computes $|G|$ in $O(L^3 \log^3 M)$ time.*

Abelian group decomposition is a very powerful tool in quantum computation. Indeed, Bermejo-Vega, Lin, and Van den Nest showed that it is *complete* for the class of problems that can be solved efficiently on quantum computers that use only quantum Fourier transforms over abelian groups along with automorphism gates and a broad class of diagonal unitaries [9]. Specifically, they showed that a classical computer augmented with just the ability to perform abelian group decomposition would be as powerful as all such quantum algorithms.

While we saw that abelian group decomposition can be solved by reducing to order finding and the HKP, it is not hard to see that both order finding the HKP over abelian groups can be reduced to abelian group decomposition.[9] In particular, this implies that abelian group decomposition can be used both to solve integer factoring and to compute discrete logarithms.

[8] The function f is surjective iff the g_i's generate G, and the latter was assumed.

[9] The order of $x \in G$ is the size of the cyclic group $\langle x \rangle$. The latter abelian group decomposes into just $\mathbb{Z}_N$, where N is the order of x. To find the kernel of $f : G \to K$, we first decompose G and K into products of cyclic groups. We can then write f as a matrix in terms of these new generators. Using techniques described in [9, Theorem 1], we can classically find generators of the kernel of this map, which are then translated into generators of the hidden subgroup via the abelian decomposition of G.

This last fact demonstrates how the power of quantum computers to understand abelian groups (in particular, to decompose them into their cyclic components) can be applied to solve important problems that are believed to be intractable on classical computers. In later chapters, we will see how this ability can be further applied to solve even more sophisticated problems. But first, we return to the HSP to see how this important problem, which is also used as a subroutine in abelian group decomposition, can be solved efficiently on a quantum computer.

3.3 Solving the Hidden Subgroup Problem

3.3.1 Abelian Groups

We now describe the standard method [46] for solving the HSP for abelian groups or the special case of the HKP.

Recall that we are given a hiding function of the form $f : G \to K$. In the HSP, we always have $K = \{0,1\}^m$, but we will stick to the more general description.

Our algorithm will work in the Hilbert space $V_G \otimes V_K$. The oracle for invoking f operates by taking a state $|g\rangle \otimes |k\rangle \in V_G \otimes V_K$ to the state $|g\rangle \otimes |f(g)k\rangle$.

Recall also that the quantum Fourier transform, $\mathcal{F}_G$, allows us to switch between the basis labeled by elements of G and the basis labeled by representations of G. (More on this below.) The algorithm starts using the basis of representations and performs the following steps:

1. Initialize the state to $|\text{triv}\rangle \otimes |e\rangle$.

2. Apply the inverse quantum Fourier transform to the first vector space to prepare

a uniform superposition over the elements of G.

$$\begin{aligned}
(\mathcal{F}_G^\dagger \otimes \mathrm{I})|\mathrm{triv}\rangle \otimes |e\rangle &= \left(\frac{1}{\sqrt{|G|}} \sum_{g\in G} \sum_{\rho \in \widehat{G}} \overline{\rho(g)} |g\rangle\langle\rho||\mathrm{triv}\rangle\right) \otimes |e\rangle \\
&= \left(\frac{1}{\sqrt{|G|}} \sum_{g\in G} \overline{\mathrm{triv}(g)} |g\rangle\right) \otimes |e\rangle \\
&= \frac{1}{\sqrt{|G|}} \sum_{g\in G} |g\rangle \otimes |e\rangle,
\end{aligned}$$

where we have used the fact that $\mathrm{triv}(g) = 1$ for all $g \in G$.

3. Apply the oracle for f, giving us the state

$$\frac{1}{\sqrt{|G|}} \sum_{g\in G} |g\rangle \otimes |f(g)\rangle.$$

4. Measure the V_K portion of the state.[10]

The probability of measuring $k \in K$ is $|\{g \in G \,|\, f(g) = k\}|/|G|$. If $k = f(g)$ for some $g \in G$, then, by assumption, the set of elements mapping to k under f is precisely gH. Hence, after the measurement, we have a **coset state**

$$|gH\rangle := \frac{1}{\sqrt{|H|}} \sum_{h\in H} |gh\rangle$$

for some $g \in G$. All such states have the same norm, so we can see that the result is a uniform distribution over coset states.

[10] By the principle of deferred measurement [51], this measurement can be done at the end, but it will simplify our discussion to use an intermediate measurement like this.

5. Apply the (regular) quantum Fourier transform to the remaining state.

$$\begin{aligned}
\mathcal{F}_G\,|gH\rangle &= \left(\frac{1}{\sqrt{|G|}}\sum_{g'\in G}\sum_{\rho\in\widehat{G}}\rho(g')|\rho\rangle\langle g'|\right)\frac{1}{\sqrt{|H|}}\sum_{h\in H}|gh\rangle \\
&= \frac{1}{\sqrt{|G||H|}}\sum_{\rho\in\widehat{G}}\sum_{h\in H}\rho(gh)|\rho\rangle \\
&= \frac{1}{\sqrt{|G||H|}}\sum_{\rho\in\widehat{G}}\left(\sum_{h\in H}\rho(h)\right)\rho(g)|\rho\rangle
\end{aligned}$$

   where we have used the fact that representations are homomorphisms, so we have $\rho(gh) = \rho(g)\rho(g)$.

6. Measure the remaining state.

   We can see that the probability of measuring ρ is equal to $|\sum_{h\in H}\rho(h)|^2/|G||H|$. Note that the factor of $\rho(g)$ disappears when we take $|\cdot|^2$, so the final measurement probabilities are unchanged by which coset we measured in step 4.

It remains to determine $\sum_{h\in H}\rho(h)$ for each choice of $\rho\in\widehat{G}$. To analyze this, note that we are only examining the restriction of ρ to the subgroup H, which we denote $\mathrm{Res}^G_H\,\rho$. In fact, this sum is simply $|H|\langle\mathrm{triv}\,|\,\mathrm{Res}^G_H\,\rho\rangle$, the inner product of the trivial representation with the restricted representation (up to a missing factor of $|H|^{-1}$). Hence, we can conclude that ρ is measured with probability $|H|/|G|$ if $\mathrm{Res}^G_H\,\rho\equiv\mathrm{triv}_H$ and 0 otherwise. Thus, the quantum procedure above generates a uniformly random element $\rho\in\widehat{G}$ with the property that its restriction to H is trivial.

To solve the problem, we need to be able to compute H from poly log $|G|$ random samples of such representations. This reduces to analyzing a system of modular equations, and the computation required to determine H can be performed efficiently on a classical computer. See [46] for details.

A couple of issues remain in showing that the quantum procedure is also efficient.

First, we must be able to efficiently implement the quantum Fourier transform, $\mathcal{F}_G$. Kitaev [40] showed that this can be done for any abelian group G. (Indeed, any abelian group is a product of cyclic groups, and it turns out that the QFT for the whole group is simply the product of QFTs for the cyclic parts, so all that is needed is to show that we can implement the QFT for any $\mathbb{Z}_N$ efficiently.)

Second, our quantum procedure started and ended in a basis labelled by the irreducible representations of G, so we need to demonstrate that we can efficiently prepare and measure states in this basis. Fortunately, this is easy to do. As we saw in section 2.2.4, each irreducible representation of an abelian group G is uniquely identified by an element $g \in G$. Hence, we can simply use the normal basis of V_G as a basis of irreducible representations. This is precisely what is done in the standard implementations of QFTs for such groups.

3.3.2 Non-abelian Groups

The quantum procedure defined above can be generalized to work on non-abelian groups. In order to describe this, we first need to generalize the quantum Fourier transform to non-abelian groups. We define the QFT for an arbitrary group G to be

$$\mathcal{F}_G := \frac{1}{|\sqrt{G}|} \sum_{g \in G} \sum_{\rho \in \widehat{G}} \sum_{i,j=1}^{d_\rho} \sqrt{d_\rho} [\rho(g)]_{i,j} |\rho, i, j\rangle \langle g|.$$

This is equivalent to the earlier definition (up to a simple isomorphism) if all of the irreducible representations are 1-dimensional. In this more general case, for each irreducible representation ρ, we have replaced the label of ρ by the label of a particular element in the matrix form ρ. By doing so, this transformation incorporates the full information about the irreducible representations for each $g \in G$.

This transformation is unitary provided that each matrix $\rho(g)$ is unitary. As we saw earlier, we can assume the latter without loss of generality.

Following the same procedure as above, in step 5, we produce the state

$$\begin{aligned}\mathcal{F}_G\,|gH\rangle &= \left(\frac{1}{\sqrt{|G|}}\sum_{g'\in G}\sum_{\rho\in\widehat{G}}\sum_{i,j=1}^{d_\rho}\sqrt{d_\rho}[\rho(g')]_{i,j}|\rho,i,j\rangle\langle g'|\right)\frac{1}{\sqrt{|H|}}\sum_{h\in H}|gh\rangle \\ &= \frac{1}{\sqrt{|G||H|}}\sum_{\rho\in\widehat{G}}\sum_{i,j=1}^{d_\rho}\sqrt{d_\rho}\left(\sum_{h\in H}[\rho(gh)]_{i,j}\right)|\rho,i,j\rangle.\end{aligned}$$

In step 6, as above, we measure the irreducible representation label ρ. In this case, that means we are leaving unmeasured the matrix indices $|i,j\rangle$. The probability of measuring ρ is, hence, equal to the norm of the state

$$\frac{1}{\sqrt{|G||H|}}\sum_{i,j=1}^{d_\rho}\sqrt{d_\rho}\left(\sum_{h\in H}[\rho(gh)]_{i,j}\right)|i,j\rangle,$$

which is equal to

$$\frac{d_\rho}{|G||H|}\sum_{i,j=1}^{d_\rho}\left|\sum_{h\in H}[\rho(gh)]_{i,j}\right|^2 = \frac{d_\rho}{|G||H|}\left\|\sum_{h\in H}\rho(gh)\right\|^2.$$

We can see that $\|\sum_{h\in H}\rho(gh)\|^2 = \|\rho(g)\sum_{h\in H}\rho(h)\|^2 = \|\sum_{h\in H}\rho(g)\|^2$ because multiplying by a unitary (in this case, $\rho(g)$) does not change the $\|\cdot\|$ norm.[11] Hence, we can see that, as in the abelian case, the measurement probabilities for the irreducible representation labels are affected by which coset gH appears in step 4. (Also note that the same would be true if we multiplied on the right by $\rho(g)$ instead of on the left. Hence, the results would be unchanged if we were given a hiding function that was constant on right cosets instead of on left cosets.)

Likewise, as in the abelian case, only the restriction of ρ to H appears in the formula for the probability of measuring ρ. In particular, suppose that we have $\mathrm{Res}_H^G\,\rho \cong \oplus_{i=1}^k\sigma_i$, then, since the $\|\cdot\|^2$ norm is unchanged by multiplying by a unitary

[11] To see this, note that $\|A\|^2 = \mathrm{tr}(A^\dagger A)$, so we have $\|UA\|^2 = \mathrm{tr}[(UA)^\dagger(UA)] = \mathrm{tr}(A^\dagger U^\dagger UA) = \mathrm{tr}(A^\dagger A) = \|A\|^2$.

matrix (a fact we used above), we can rewrite this expression as

$$\left\|\sum_{h\in H}\rho(h)\right\|^2 = \left\|\sum_{h\in H}(\mathrm{Res}_H^G\,\rho)(h)\right\|^2 = \left\|\sum_{h\in H}\bigoplus_{i=1}^{k}\sigma_i(h)\right\|^2 = \sum_{i=1}^{k}\left\|\sum_{h\in H}\sigma_i(h)\right\|^2.$$

The symmetry of $\sum_{h\in H}\sigma_i(h)$ suggests that we can simplify further. In fact, this is an intertwining map. Using the terminology of Corollary A.4, it is precisely T_{triv}, which tells us that $T_{\text{triv}} = (|H|/d_{\sigma_i})\langle \text{triv} \,|\, \chi_{\sigma_i}\rangle\, \mathrm{I}$. Next, we can see that $\|T_{\text{triv}}\|^2 = |H|^2|\langle \text{triv} \,|\, \chi_{\sigma_i}\rangle|^2 = |H|^2\langle \text{triv} \,|\, \chi_{\sigma_i}\rangle$ since $\langle \text{triv} \,|\, \chi_{\sigma_i}\rangle$ is either 0 or 1. Putting this all together, we have

$$\left\|\sum_{h\in H}\rho(h)\right\|^2 = \sum_{i=1}^{k}|H|^2\langle \text{triv} \,|\, \chi_{\sigma_i}\rangle = |H|^2\langle \text{triv} \,|\, \sum_{i=1}^{k}\chi_{\sigma_i}\rangle = |H|^2\langle \text{triv} \,|\, \chi_{\mathrm{Res}_H^G\,\rho}\rangle,$$

which means that the probability of measuring ρ is $d_\rho(|H|/|G|)\langle \text{triv} \,|\, \chi_{\mathrm{Res}_H^G\,\rho}\rangle$.

The above analysis is from [32]. Hallgren et al. go on to show that, if $H \lhd G$ is normal, then the only irreducible representations ρ for which $\langle \text{triv} \,|\, \chi_{\mathrm{Res}_H^G\,\rho}\rangle$ is nonzero are those that are uniformly trivial on H. This means that we measure ρ with probability $(|H|/|G|)d_\rho^2$ if $H \leq \operatorname{Ker}\rho$ and 0 otherwise.

It follows that $H \lhd G$ is contained in the intersection of the kernels of all the measured irreducible representations. Hallgren et al. show that, after $O(\log|G|)$ samples, the probability that the intersection is not precisely H is exponentially small. Hence, we get an algorithm for solving the HSP in any group where all subgroups are normal—the so-called Hamiltonian groups—assuming that we can efficiently compute the QFT for such groups and compute the intersections of kernels of irreducible representations.[12] Hallgren et al. show that these things can be done, giving an efficient quantum algorithm for an interesting class of non-abelian groups.

Even though the QFT for a non-abelian group produces information along with the matrix indexes, this part of the state is never examined by the above algorithm, which measures only the irreducible representation label. This approach is called

[12] We also need to show that we can prepare and measure in the basis of irreducible representations, but that usually follows, as it does in this case, when we have an efficient QFT.

weak Fourier sampling, in contrast to **strong Fourier sampling**, where the matrix index registers are also used.

3.3.3 Generalized Hiding Functions

Another important non-abelian result was that of Ivanyos, Sanselme, and Santha [37], which gave an efficient algorithm for the HSP on groups of nilpotency class 2. The latter class also includes the Hamiltonian groups as well as the Heisenberg group and many others [6, 5]. As far as we are aware, their algorithm covers the largest general class of groups for which the HSP can be solved efficiently.

We will have more to say on their result in the next chapter. However, we mention one of their important observations here: since the standard method outlined above for solving the HSP (on abelian or non-abelian groups) does not use the measured value of the hiding function, the algorithm would operate identically if we multiplied that part of the state by an arbitrary unitary.

More generally, we can replace the hiding function by any other quantum process that produces states that are equal to those of a hiding function after applying some unitary. The hiding function produces states $|f(g)\rangle$ with the property that $\langle f(g') \mid f(g)\rangle$ is 1 if $g' = gh$ for some $h \in H$ and 0 otherwise. Hence, we can see that it is sufficient to have a quantum process that produces states with these orthogonality properties in some basis, not necessarily the standard basis of our vector space. That is, we can use any quantum process taking, for $g \in G$, the state $|g\rangle \otimes |0\rangle$ to $|g\rangle \otimes |u_g\rangle$, where the $|u_g\rangle$'s have the property that $\langle u_{g'} \mid u_g\rangle = 1$ if $g' = gh$ for some $h \in H$ and 0 otherwise. We will call this a (quantum) **hiding process** for H.

Ivanyos et al. take advantage of this generality by constructing a more general quantum process that hides a subgroup that they wish to find. We will use the same approach in the next chapter.

Finally, the above discussion shows that having a process taking $|g\rangle \otimes |0\rangle$ to $|g\rangle \otimes |u_g\rangle$ is sufficient to produce uniformly random coset states of the hidden subgroup $H \leq G$ via Fourier sampling. However, it is interesting to note that these two capabilities are essentially equivalent.

If we have a quantum process taking $|0\rangle \mapsto |g_* H\rangle$ for a uniformly random $g_* \in G$, then we can compose this with the process performing right multiplication—i.e., taking $|g\rangle \otimes |g_* H\rangle$ to $|g\rangle \otimes \text{reg}_R(r)|g_* H\rangle = |g\rangle \otimes |g_* Hg\rangle$—to get a process suitable for use in Fourier sampling. This holds since $g_* hg = g_* h'g'$ iff $hg = h'g'$ iff $g' = h''g$, which shows that the resulting states are disjoint unless $g' = h''g$ for some $h'' \in H$ (in which case, they are equal). The only minor difference is that measuring this will give us a right coset, rather than a left coset, but, as we saw above, this gives identical measurement outcomes in Fourier sampling.

Chapter 4

An Application of Abelian Substructure

In this chapter, we turn to the main focus of this thesis, which is to examine ways in which the ability of quantum computers to solve problems on abelian algebraic structures can be leveraged to solve problems on non-abelian ones. We begin, here, with the simplest such case, which occurs when a non-abelian algebraic structure contains within it a well-organized set of substructures that are abelian.

The neatest example of this type is the class of nilpotent groups. As we saw earlier, each nilpotent group has a nontrivial center, which is an abelian subgroup. More importantly, if we quotient out by the center, we are left with a smaller nilpotent group, which has the same properties including its own nontrivial center. As we will see in this chapter, this tower of abelian subgroups provides enough structure to allow us to solve interesting problems on non-abelian nilpotent groups by reducing them to problems on the abelian subgroups.

The chapter is organized as follows. We start in section 4.1 by showing that the nilpotent groups have special status with regard to the HSP: they are the only groups for which weak Fourier sampling algorithm is always successful (Corollary 4.5). Then, we turn to constructing more efficient algorithms. In section 4.2, we consider the problem of constructing a hiding process for the normal closure of the hidden subgroup, and in section 4.3, we discuss the average-case subset sum problem. In section 4.4, we use those two pieces to formulate useful results on solving the HSP on nilpotent groups: we give an efficient quantum algorithm for HSP on k-nilpotent groups with $k = O(1)$ assuming an oracle for average-case subset sum (Theorem 4.10), we demonstrate a polynomial quantum speedup for general nilpotent groups (Corollary 4.11), and we give a subexponential time algorithm for nilpotent groups whose order is divisible by only large primes (Theorem 4.12). Finally, we conclude in section 4.5.

4.1 Weak Fourier Sampling of Nilpotent Groups

In this section, we describe the close connection between successfully finding hidden subgroups via weak Fourier sampling (WFS) and nilpotency. Specifically, we show that the groups on which the HSP can always be solved by *recursive* application of WFS are precisely the nilpotent groups.

An immediate consequence of the formula given in section 3.3.2, which describes the probability distribution of the irrep label ρ produced by WFS, is that WFS cannot distinguish H from any conjugate H^g since every character χ_ρ is a class function: χ_ρ is identical on H and H^g, so the probability distribution gives us no information about whether H or H^g is hidden. This shows that WFS cannot solve the HSP for every $H \leq G$ whenever G is non-abelian. Hallgren et al. do show, however, that for normal subgroups, which have no conjugates, WFS is always successful.

That is not the end of the story, however, for non-normal subgroups. While it is true that the distribution is identical for H and H^g, this does not rule out the possibility that we could learn something collectively about H and its conjugates that would allow us to make progress on finding H. We might hope, for example, to find the normal closure of H — the smallest normal subgroup containing H and every H^g for $g \in G$. If the normal closure of H is not the whole group (which it will never be in a nilpotent group, for example), then this would identify a smaller subgroup $K \leq G$ containing H and we could continue searching for H in K.

We refer to the approach just outlined as **recursive weak Fourier sampling** (RecWFS). More formally, we apply WFS to the group G and use the distribution on irrep labels produced to determine (information theoretically) a smaller subgroup $K \leq G$ such that $H^g \leq K$ for all $g \in G$. If $H = K$, then the method succeeds.[1] If $K = G$, on the other hand, then the method fails. Otherwise, we have $H \lneq K \lneq G$, and we apply the same approach *recursively* to the smaller group K.[2]

The next result shows that RecWFS cannot be successful for non-nilpotent groups.

Theorem 4.1. *If G is not nilpotent, then* RecWFS *fails for some $H \leq G$.*

Proof. If G is not nilpotent, then it contains a proper self-normalizing[3] subgroup H by Theorem 2.5. This means, in particular, that each of the conjugate subgroups H^g, for $g \in G/H$, is distinct, and this remains true if we restrict to any subgroup $K \leq G$ strictly larger than H.

By the formula from section 3.3.2, WFS will fail to distinguish H from its conjugates. And since H is self-normalizing, it will have $[G : H] \neq 1$ conjugates, all of which must be included in K. When we restrict to K, it remains true that we have $[K : H] \neq 1$ conjugates, which must be included in the next subgroup. While the group may shrink on each recursive application, it can never shrink to $K = H$ since H always has conjugate subgroups in K. Since G is finite, we cannot continue to move to smaller subgroups forever, so we eventually reach a K such that $\langle H^g \mid g \in G \rangle = K$, at which point the method fails. □

The above theorem is not all there is to say for non-nilpotent groups, however, because, even though these conjugate subgroups cannot be distinguished by WFS, they may not be difficult to distinguish classically. As a trivial example, the above theorem says that WFS fails for a family of exponentially large abelian groups if we simply adjoin a constant size non-nilpotent factor to each. However, we could find a hidden subgroup in this non-nilpotent part of the group efficiently by brute force.

The following result fixes most of this problem.

[1] As we will see below, we can identify whether we have reached this case without knowing H.

[2] This approach can be generalized by also identifying a $J \lhd G$ such that $J \leq H^g$ for all $g \in G$ and then recursing on K/J rather than K.

[3] H is self-normalizing if $H^g = H$ iff $g \in H$.

Corollary 4.2. *If G contains a self-normalizing subgroup H with $[G : H]$ exponentially large and $|H|$ prime, WFS and classical computation cannot find H efficiently.*

Proof. As described in the previous theorem, there are $[G : H]$ conjugates of H, which is exponentially many, and these cannot be distinguished by WFS applied to any group that contains H and all of its conjugates. With polynomially many queries, we also cannot check more than a vanishingly small fraction of them classically, so any algorithm of this sort will have a vanishingly small probability of succeeding.[4] □

Typically, we focus on the problem of finding hidden subgroups that are as small as possible, either trivial or cyclic of prime order, since that is the case that is, intuitively, the hardest classically. Hence, this corollary tells us that WFS cannot solve the typical hidden subgroup problems for non-nilpotent groups.

Our positive results will rely on the following basic fact.

Proposition 4.3. *The probability of measuring irrep ρ by WFS is $(|H|/|G|)d_\rho^2$ if $H \subset \operatorname{Ker}\rho$ and strictly smaller if $H \not\subset \operatorname{Ker}\rho$.*

Proof. For any unitary $U \in \mathbb{C}^{n \times n}$, we note that $|\operatorname{tr} U| \leq n$, as the left hand side is the sum of the eigenvalues of U each of which lies on the unit circle. By the same reasoning, we have equality iff $U = I$. For any representation ρ, this means that $|\chi_\rho(g)| \leq d_\rho$ for any $g \in G$, with equality iff $g \in \operatorname{Ker}\rho$. With this, the proposition follows from our formula from section 3.3.2. □

With this in hand, we can now prove the following theorem, which is the converse to the negative result above.

Theorem 4.4. *If G is nilpotent, then* RecWFS *succeeds (given sufficient time) for any hidden subgroup $H \leq G$.*

Proof. Proposition 4.3 tells us that WFS distinguishes those irreps whose kernels include H. The intersection of those kernels is a normal subgroup $K \leq G$ containing

[4] Here, we are using the fact that conjugates H^g and H^h have trivial intersection, so queries give us no information about H unless we guess it correctly.

H. (In fact, it is the normal closure of H.[5]) K must be a proper subgroup of G since H is contained in some maximal proper subgroup $M \lneq G$ and, in a nilpotent group, maximal proper subgroups are normal.[6] In more detail, we have $H^g \leq M^g = M$, and since K is the smallest normal subgroup containing all H^g, K is contained in M as well. Finally, since M is a proper subgroup, so is $K \leq M$.

The above argument shows that we can always find, by WFS, a strictly smaller subgroup $K \lneq G$ that contains H. Since subgroups of nilpotent groups are nilpotent, when we use WFS in K, we either find H or get an even smaller subgroup $K^{(2)} \lhd K$. Since G is finite and we shrink at each step by at least a factor of 2 in size, this process must eventually reach $K^{(n)} = H$ for some n. □

All together, we have proven the following:

Corollary 4.5. RecWFS *succeeds for every $H \leq G$ iff G is nilpotent.*

Corollary 4.5 shows that recursive weak Fourier sampling is successful, at least information theoretically, for precisely the nilpotent groups. Hence, the nilpotent groups are a natural class in which to look for *efficient* algorithms. Unfortunately, WFS can take exponential time to identify a normal subgroup $K \lneq G$ containing H for some nilpotent groups.[7] Hence, we will need further tools in order to get potentially efficient algorithms. We discuss the first of these in the next section.

4.2 A Hiding Process for the Normal Closure

In this section, we consider the problem of finding a process for hiding the normal closure of the hidden subgroup H. That is, we want a quantum process that prepares, for each $g \in G$, a state $|u_g\rangle$ such that $\langle u_{g'} \,|\, u_g\rangle = 1$ if $g' = gh$ for some $h \in K$ and 0

[5] This follows from the results of Hallgren, Russell, and Ta-Shma [32]. In particular, their results imply that any normal subgroup is the intersection if the kernels of the irreps that contain it. Hence, the intersection of the kernels of the irreps that contain H is the smallest normal subgroup containing H, i.e., the normal closure of H.

[6] If a maximal subgroup is not normal, then it is self-normalizing, so this follows from the fact that nilpotent groups do not contain self-normalizing subgroups 2.5.

[7] The 3-nilpotent unitriangular matrices are an example.

otherwise, where K is the normal closure of H. We will show that this can be done provided that we can solve the Average-Case Subset Sum Problem (AvgSSP).

The value of having such a hiding process is that the normal closure of H is, by definition, a normal subgroup. As we saw earlier, we should be able to efficiently produce a generating for set this (or any normal) subgroup by weak Fourier sampling. In fact, in this case, this will leave us with an *abelian* HSP, which we know can be solved efficiently.

We will describe a quantum process for hiding the subgroup $HZ_{k-1}(G)$, where G is a k-nilpotent group. For any $g \in G$, since $h^g = h[h, g]$ and $Z_{k-1}(G)$ contains all commutators, we can see that the normal closure of H is contained in $HZ_{k-1}(G)$. In principle, the normal closure of H could be a proper subgroup of this; however, there is no harm in working with $HZ_{k-1}(G)$ instead since it is also normal[8], and, as we shall see, finding this subgroup will also give us the information we need to find H.

Let us start by considering the slightly easier problem of constructing a hiding process for the subgroup $HZ(G)$, a smaller group than $HZ_{k-1}(G)$ whenever $k > 2$. As we saw before, it would be sufficient to find a way to prepare a coset state $|gHZ(G)\rangle$ since the action of right multiplication on this state is then a hiding process.

Unfortunately, there appears to be no easy way to prepare this state. Instead, following [37], we construct a state that differs from this only by the presence of some extra phases.

We begin by preparing a random coset state, $|gH\rangle$, which, as we saw above, we can do using the hiding function for H. Then, we adjoin a new register and use an inverse Fourier transform to create a superposition on the elements of the center

$$(\mathcal{F}^\dagger_{Z(G)} \otimes \mathrm{I})|\mathrm{triv}\rangle \otimes |gH\rangle = \frac{1}{\sqrt{|Z(G)|}} \sum_{z \in Z(G)} |z\rangle \otimes |gH\rangle$$

Next, we perform a multiplication taking $|z\rangle \otimes |x\rangle$ to $|z\rangle \otimes |xz\rangle$. This leaves us with

$$|\psi\rangle := \frac{1}{\sqrt{|Z(G)|}} \sum_{z \in Z(G)} |z\rangle \otimes |gHz\rangle.$$

[8] Recall that, for $z \in Z_{k-1}(G)$ and any $g \in G$, we have $z^g \in Z_{k-2}(G) \leq Z_{k-1}(G)$.

Applying an Fourier transform on the first part turns this into

$$\begin{aligned}(\mathcal{F}_{Z(G)} \otimes \mathrm{I})|\psi\rangle &= \frac{1}{|Z(G)|} \sum_{z \in Z(G)} \sum_{\chi \in \widehat{Z(G)}} \chi(z)|\chi\rangle \otimes |gHz\rangle \\ &= \frac{1}{|Z(G)|} \sum_{\chi \in \widehat{Z(G)}} |\chi\rangle \otimes \sum_{z \in Z(G)} \chi(z)|gHz\rangle.\end{aligned}$$

If we let $|\psi_\chi\rangle := (|H||Z(G)|)^{-1/2} \sum_{h \in H} \sum_{z \in Z(G)} \chi(z)|ghz\rangle$, then we can see that $\langle \psi_\chi \,|\, \psi_\chi \rangle = (|H||Z(G)|)^{-1} \sum_{h \in H} \sum_{z \in Z} \overline{\chi(z)}\chi(z) = 1$. Hence, if we measure the first part, we find a uniformly random $\chi \in \widehat{Z(G)}$ and are left with the state $|\psi_\chi\rangle$.

Let us see how close this is to the hiding process that we desire. We consider the action of right multiplication on this state, defining $|u_x\rangle := \mathrm{reg}_R(x)|\psi_\chi\rangle$, and check whether these $|u_x\rangle$'s satisfy the orthogonality properties needed to hide $HZ(G)$.

First, we can see that $ghzx = gh'z'x'$ iff $hzx = h'z'x'$ iff $x' = h''z''x$ for some $h'' \in H$ and $z'' \in Z(G)$.[9] This shows that $\langle u_x \,|\, u_{x'} \rangle = 0$ unless $x' = h''z''x$ for some $h'' \in H$ and $z'' \in Z(G)$.

Next, we can see that $\langle ghz \,|\, \psi_\chi \rangle = \chi(z)/\sqrt{|H||Z(G)|}$, for all $h \in H$, and, since we have $ghz \cdot h' = ghh'z$, we can see that $\langle ghz \,|\, \mathrm{reg}_R(h')\psi_\chi \rangle = \langle gh(h')^{-1}z \,|\, \psi_\chi \rangle = \chi(z)/\sqrt{|H||Z(G)|}$. In other words, since multiplication on the right by h' only permutes states in the superposition with the same phase, we have $\mathrm{reg}_R(h)|\psi_\chi\rangle = |\psi_\chi\rangle$.

So far, this is exactly what we want. On the other hand, we can see that $\langle ghz \,|\, \mathrm{reg}_R(z')\psi_\chi \rangle = \langle ghz(z')^{-1} \,|\, \psi_\chi \rangle = \chi(z)\overline{\chi(z')}/\sqrt{|H||Z(G)|}$. Hence, multiplication on the right by $z' \in Z(G)$ differs from the desired state by a phase of $\overline{\chi(z')}$.

We can remove this extra phase with the following approach. Instead of using a single state $|\psi_\chi\rangle$, we will use a state of the form $|\psi_{\chi_1}\rangle \otimes \cdots \otimes |\psi_{\chi_k}\rangle$ and have $g \in G$ act by $\mathrm{reg}_R(g) \otimes \cdots \otimes \mathrm{reg}_R(g)$, i.e., by multiplication on all parts. If $g' \neq hzg$ for some $h \in H$ and $z \in Z$, then all parts will be orthogonal, so the combined state is orthogonal. Furthermore, multiplying by h on all parts leaves all parts unchanged, so the overall state is unchanged. Finally, multiplying by z on all parts picks up an overall phase of $\overline{\chi_1(z) \cdots \chi_k(z)}$. If we find χ_i's such that their pointwise product is 1

[9] Note that we have used the fact that z and z' are in the center.

everywhere, then we have the state that we need to find $HZ(G)$.

Each χ is a representation of $Z(G)$. We know from Theorem 2.1 that $Z(G) \cong \mathbb{Z}_{p_1^{t_1}} \times \cdots \times \mathbb{Z}_{p_\ell^{t_\ell}}$ for some p_i's (primes) and t_i's. We also know representations of such groups are of the form $\chi(x_1, \ldots, x_\ell) = \omega_{p_1^{t_1}}^{a_1 x_1} \cdots \omega_{p_\ell^{t_\ell}}^{a_\ell x_\ell}$, where each ω_M is an M-th root of unity. Thus, if we have a set of uniformly random representations $\chi_1, \ldots, \chi_m$, we can find a subset whose pointwise product is zero if we can solve the following problem in the special case $B = 0$.

Definition 4.6. *Given a set of values $A_1, \ldots, A_m$, each chosen uniformly at random from $X := \mathbb{Z}_{M_1} \times \cdots \times \mathbb{Z}_{M_\ell}$, and a number $B \in X$, the* Average-Case Subset Sum Problem *(*AvgSSP*) over X asks us to find a* nonempty *subset $A_{i_1}, \ldots, A_{i_k}$ that sums to B in X.*

Given an oracle that solves AvgSSP, we can see how to construct a hiding process for $HZ(G)$. Applying this idea iteratively to $Z(G), Z_2(G), \ldots, Z_{k-1}(G)$, gives us a hiding process for $HZ_{k-1}(G)$.

Lemma 4.7. *Given an efficient algorithm that solves* AvgSSP *when provided with $f(X)$ random values for each abelian group X, there is a hiding process for the group $HZ_{k-1}(G)/Z_{k-1}(G)$ running in time polynomial* $\log |G|$ *and in the quantity $f(Z(G)) \cdots f(Z_{k-1}(G)/Z_{k-2}(G))$.*

Proof. We start by preparing $f(Z(G))$ states of the form $|\psi_{\chi_j}\rangle$ with each χ_j chosen uniformly at random from the characters of $Z(G)$. Next, we invoke the oracle to find a subset of the χ_j's whose pointwise product is zero. As we saw above, acting on this set by simultaneous right multiplication gives a hiding process for $HZ(G)$. Preparing a superposition $\sum_{g \in G} |g\rangle \otimes |0\rangle$, applying the hiding process, and throwing away the second part of this state, gives a coset state $|gHZ(G)\rangle$, for a uniformly random g.[10]

Thus, we have moved from a hiding process for H to a hiding process for $HZ(G)$. Since $Z(G) \leq HZ(G)$, we can identify the latter subgroup with $HZ(G)/Z(G) \leq$

[10] Actually, this will give us a left coset of $HZ(G)$, but as we saw above, the two behave identically with respect to Fourier sampling. The only change is that, in the next round, we will use left multiplication rather than right multiplication to construct the hiding process for $HZ_2(G)$.

$G/Z(G)$. We can repeat the above process with H replaced by $HZ(G)/Z(G)$ and G replaced by $G/Z(G)$. That will gives us $HZ(G/Z(G)) \cong HZ_2(G)$. Repeating this for $i = 3, \ldots, k-1$ gives us the state $HZ_{k-1}(G)/Z_{k-1}(G)$.

It remains to analyze the total running time. Let $T(i)$ denote the time required for the hiding process of $HZ_i(G)/Z_i(G)$. For level $i+1$, the time to prepare each $|\psi_\chi\rangle$ state is poly$(\log|G|) + T(i)$. Since we repeat this for $m_{i+1} := f(Z_{i+1}(G)/Z_i(G))$ states, the total time is $T(i+1) = m_{i+1}[(\operatorname{poly}\log|G|) + T(i)] + g(m_{i+1})$, where $g(m)$ is the running time of the AvgSSP algorithm. The solution to this is bounded by $(\operatorname{poly}\log|G|)m_1 \cdots m_{k-1} + O(g(m_1 \cdots m_{k-1}))$, which shows that the running time is polynomial in $(\log|G|)m_1 \cdots m_{k-1}$. □

It is worth noting that the space requirement of this algorithm is not nearly so bad as the time requirement. At each point in time, we only need space to store the each of the $f(Z_{i+1}(G)/Z_i(G))$ states of the form $|\psi_\chi\rangle$ at each level $i = 1, \ldots, k-1$. Thus the overall space requirement is only $O(\log|G|)\sum_{i=1}^{k=1} f(Z_i(G)/Z_{i-1}(G))$.

4.3 Algorithms for Average-Case Subset Sum

The quantum hiding process described above depends on having an algorithm for solving AvgSSP. It is well-known that AvgSSP can be solved in $O(nM_1 \cdots M_\ell)$ time using dynamic programming [28]. We will be most interested in cases when $M_1 \cdots M_\ell$ is exponentially large. In that case, this standard algorithm takes exponential time.

Using techniques developed in [26], we can reduce the time requirement to subexponential. The following theorem will give us, the next section, an algorithm whose running time is subexponential (in $\log(M_1 \cdots M_\ell)$) provided that $\ell = \operatorname{poly}\log M$, where $M = \max_i M_i$.

Theorem 4.8. *There is a polynomial time algorithm for* AvgSSP *when* $B = 0$ *provided that* $m = \exp(\Omega(\sqrt{\log M_1}) + \cdots + \sqrt{\log M_\ell}))$.

Proof. The case $\ell = 1$ is proven in [26]. We extend to $\ell > 1$ by brute force. Let $m_i = \exp(C\sqrt{\log M_i})$, the number of inputs required for $\ell = 1$ [26]. We require the number of inputs m for the overall problem to be the product $m_1 \cdots m_\ell$.

To solve the problem for $\ell > 1$, we split the inputs into groups of m_1 elements and solve AvgSSP on each group using the algorithm of [26] applied to the the $\mathbb{Z}_{M_1}$ part of each number. This gives us, for each group of m_1 numbers, a nonempty subset that sums to zero in the $\mathbb{Z}_{M_1}$ part.

Replacing the solution for each group with the sum of those numbers in $\mathbb{Z}_{M_1} \times \cdots \times \mathbb{Z}_{M_\ell}$, gives us $m_2 \cdots m_\ell$ numbers with zeros in the $\mathbb{Z}_{M_1}$ part. Next, we group these numbers into groups of size m_2 and repeat as above but now using the $\mathbb{Z}_{M_2}$ part. For each group, we get a solution that sums to zero in this part, but furthermore, since each number was already zero in the $\mathbb{Z}_{M_1}$ part, we can see that each solution sums to zero in the $\mathbb{Z}_{M_1} \times \mathbb{Z}_{M_2}$ part.

Repeating this j times, leaves us with $m_{j+1} \cdots m_\ell$ numbers that sum to zero in the $\mathbb{Z}_{M_1} \times \mathbb{Z}_{M_j}$ part. Hence, if we repeat $j = \ell$ times, we get a single number that is zero in all copies. The set of inputs we added to produce this result is a solution. It is routine to verify that the total running time is still polynomial in m. □

4.4 Quantum Algorithms for the Nilpotent HSP

Applying the algorithm for AvgSSP discussed in the last section to the method of the previous section gives us a quantum hiding process for $HZ_{k-1}(G)/Z_{k-1}(G)$. Since $G/Z_{k-1}(G)$ is an abelian group, we can find $HZ_{k-1}(G)/Z_{k-1}(G)$ efficiently using the standard method. Identifying this subgroup of $G/Z_{k-1}(G)$ also identifies $HZ_{k-1}(G)$ (it's inverse image under the quotient map).[11]

This shows that we can reducing finding the subgroup $HZ_{k-1}(G)$ to an abelian HSP. But how does knowing this subgroup help us find H? The key idea comes from the following lemma.

Lemma 4.9. *Let H be a cyclic subgroup of a k-nilpotent group G, then $HZ_{k-1}(G)$ is a $(k-1)$-nilpotent group.*

[11] More concretely, simply adjoining a generating set for $Z_{k-1}(G)$ to the inverse image of the generators of $HZ_{k-1}(G)/Z_{k-1}(G)$ gives a generating set for $HZ_{k-1}(G)$.

Proof. Let $G' := HZ_{k-1}(G)$.

First, we will check that $Z_i(G') \supset Z_i(G)$ for $i = 0, \ldots, k$. If $z \in Z(G)$, then it commutes with all of G, which includes G', so $z \in Z(G')$. This is the case $i = 1$. For $i > 1$, if $z \in Z_i(G)$, then for all $g \in G$, we have $[z, g] \in Z_{i-1}(G) \subset Z_{i-1}(G')$ by induction. In particular this holds for those $g \in G'$, so we have $z \in Z_i(G')$.

Now, an element of G' can be written as $h^i z$ for some $i \in \mathbb{Z}_p$ and $z \in Z_{k-1}(G)$. Let $h^j z'$ be another element of G'. By the definition of $Z_{k-1}(G)$, both z and z' commute with everything, including $h \in H$, modulo $Z_{k-2}(G)$. Hence,

$$h^i z h^j z' = h^i h^j z z' = h^i h^j z' z = h^{i+j} z' z = h^j h^i z' z = h^j z' h^i z \pmod{Z_{k-2}(G)}.$$

Thus, every $h^i z \in G'$ is in the center of G' modulo $Z_{k-2}(G)$. Since $Z_{k-2}(G') \supset Z_{k-2}(G)$, the same is true modulo $Z_{k-2}(G')$. Thus, we see that $Z_{k-1}(G') = G'$, which shows that G' is $(k-1)$-nilpotent. □

Once we have found $HZ_{k-1}(G)$, we have reduced to the problem of finding H in $HZ_{k-1}(G)$. The lemma shows that the latter is a $(k-1)$-nilpotent group, which is one smaller in nilpotency class than the k-nilpotent group G. Thus, we have reduced to finding H inside of a simpler group.

Now, the lemma requires that H be a cyclic subgroup. However, as occurs in many HSPs (see, e.g., [42, 5]), we can efficiently reduce to this case. For nilpotent groups, this was shown in [37].[12] This reduction adds only a $O(\log^2 |G|)$ factor to the total running time.

The above lemma allows us to prove the existence of useful algorithms for the HSP on k-nilpotent groups by induction on k. We begin by showing that we can efficiently solve the HSP on groups of small nilpotency class assuming we can solve AvgSSP efficiently. This was previously shown for the dihedral group [52].

Theorem 4.10. *Given an oracle for* AvgSSP*, there is an efficient quantum algorithm for solving the HSP on k-nilpotent groups with $k = O(1)$.*

[12] Lemma 2 of [37] is stated only for p-groups. However, the technique works more generally by replacing the refined polycyclic representation with a chief series for the nilpotent group. On the other hand, we can also reduce to the p-group case here and then apply their Lemma 2 directly.

Proof. We prove this by induction on k. The case $k \leq 2$ is known (even without an oracle for AvgSSP) from [37].

For $k > 2$, we apply Lemma 4.7. An application of Chebyshev's inequality tells us that $m = 2\log|X|$ values are sufficient for the existence of a solution to this instance of AvgSSP with high probability. Thus, the hiding process given by Lemma 4.7 runs in time polynomial in $(\operatorname{poly}\log|G|)\log^{k+1}|G| = \operatorname{poly}\log|G|$ time. Hence, the standard results on abelian HSPs [46] tell us that we can find $HZ_{k-1}(G)$ time polynomial in $\log|G|$. By induction, we can find H within the subgroup $HZ_{k-1}(G)$, which we know is $(k-1)$-nilpotent by Lemma 4.9. □

Even with brute-force algorithms for AvgSSP, the above approach still gives a polynomial speedup for groups of small nilpotency class.

Corollary 4.11 (due to Aram Harrow[13]). *Let* $B_i := |Z_i(G)/Z_{i-1}(G)|$ *and* $B := \max_i B_i$. *Then the total time for solving subset sum instances in the above algortihm using a brute-force approach is* $O(B\operatorname{poly}\log|G|)$. *This is* $O(|G|^\alpha \operatorname{poly}\log|G|)$ *for* $\alpha = \log B / \sum_{i=1}^k \log B_i = \log B / \log|G|$.

For our example of k-nilpotent unitriangular matrices, with $k = O(1)$, the cost of classically solving the subset sum instances is $O^*(|G|^{2/(k+1)})$, and the cost of the quantum computation is only $O(\operatorname{poly}\log|G|)$.

Finally, we can use Theorem 4.8, which gives a substantial speedup over the brute-force algorithm, in order to get subexponential-time algorithms for the HSP on nilpotent groups whose order is divisible by only large primes. This includes $k = O(1)$ as a special case.

Theorem 4.12. *Let* G *be a nilpotent group such that, for some* $\epsilon > 0$, $|G|$ *is divisible only by primes* p *with* $p \geq \exp(\log^\epsilon|G|)$. *Then there is a quantum algorithm solving the HSP on* G *in* $\exp(O(\log^{1-\epsilon}|G|))$ *time.*

Proof. As we saw in Lemma 4.7 and above, the overall cost of the algorithm is $(\operatorname{poly}\log|G|)\prod_{i=1}^k f(B_i)$, where $B_i = |Z_i(G)/Z_{i-1}(G)|$ and $f(B_i)$ is the number of

[13] Personal communication.

samples needed to solve AvgSSP on the abelian group B_i.[14]

By Theorem 4.8, if $X = \mathbb{Z}_{p_1^{t_1}} \times \cdots \times \mathbb{Z}_{p_\ell^{t_\ell}}$, we can take $f(X)$ to be $\exp \Theta(\sqrt{t_1 \log p_1} + \cdots + \sqrt{t_\ell \log p_\ell})$. We can upper bound the latter by $\exp \Theta(t_1 \sqrt{\log p_1} + \cdots + t_\ell \sqrt{\log p_\ell})$. Multiplying these together for each of the factor groups $Z_{i+1}(G)/Z_i(G)$ gives a bound of $\exp \Theta(t_1 \sqrt{\log p_1} + \cdots + t_n \sqrt{\log p_n})$, where $|G|$ factors as $p_1^{t_1} \cdots p_n^{t_n}$.

Let $T := t_1 + \cdots + t_n$. By the concavity of square root, we can see that

$$\begin{aligned} t_1\sqrt{\log p_1} + \cdots + t_n\sqrt{\log p_n} &= T(\tfrac{t_1}{T}\sqrt{\log p_1} + \cdots + \tfrac{t_n}{T}\sqrt{\log p_n} \\ &\leq T(\tfrac{t_1}{T}\log p_1 + \cdots + \tfrac{t_n}{T}\log p_n)^{1/2} \\ &= \sqrt{T}\sqrt{\log |G|} \end{aligned}$$

This is $O(\log^{1-\epsilon} |G|)$ provided that $T = O(\log^{1-\epsilon'} |G|)$ for some $\epsilon' > 0$.

Let p be the smallest prime dividing $|G|$. Then we can see that $T = t_1 + \cdots + t_n \leq t_1 \frac{\log p_1}{\log p} + \cdots + t_n \frac{\log p_n}{\log p} = \log |G| / \log p$. Hence, if $p \geq \exp(\log^\epsilon |G|)$ for some $\epsilon > 0$, then $T \leq \log |G| / \log^\epsilon |G| = \log^{1-\epsilon} |G|$. □

4.5 Conclusion

In this chapter, we saw that there are improved algorithms for solving the HSP on non-abelian groups when those groups contain a well-organized set of abelian subgroups, namely, when those groups are nilpotent.

We saw, in section 4.1, that the nilpotent groups are precisely those groups for which the HSP can be solved for every $H \leq G$ using just the information provided by weak Fourier sampling applied recursively.

We also saw, in section 4.4, that there are subexponential time algorithms for solving the HSP on nilpotent groups whose orders are divisible by only large primes. The algorithm worked by using the supplied hiding function for H in order to construct a hiding process for the subgroup $HZ_{k-1}(G)/Z_{k-1}(G)$. The latter is a subgroup of the *abelian* group $G/Z_{k-1}(G)$, so we can use the standard abelian HSP algorithms to find

[14] Since the subgroup $HZ_{k-1}(G)$ on which we recurse is strictly smaller than G, the overall time is dominated by the work done to find $HZ_{k-1}(G)$.

this group. Once $HZ_{k-1}(G)$ is found, we search for H recursively within this group of smaller nilpotency class. This approach takes advantage of the well organized set of groups $Z_1(G) \leq \cdots \leq Z_{k-1}(G)$, each of which is abelian modulo the previous one, in order to reduce the HSP on G to a sequence of HSPs on the abelian factor groups.

In the next chapter, we will see an entirely different way of using abelian algebraic structures to solve HSPs on non-abelian groups.

Chapter 5

An Application of Translation Into Abelian Structures

In this chapter, we look at a second way of leveraging the ability of quantum computers to solve problems on abelian algebraic structures in order to solve problems on non-abelian ones. This time, we will translate the non-abelian structure into an abelian one that has nicer computational properties and preserves sufficient information about the original structure in order to solve the problem of interest.

Our example in this chapter comes from the hidden kernel problem (HKP), first described in section 3.2.1, which is a special case of the HSP where the hidden subgroup is the kernel of a group homomorphism. Any such subgroup is normal (and any normal subgroup is the kernel of some homomorphism). The values of the hiding function now lie in a known group (rather than being opaque labels), which gives extra information to the algorithms. However, as discussed in section 3.2.1, the HKP is still at least as hard as discrete logarithm.

As we mentioned in the previous chapter, the groups in which all subgroups are normal, the Hamiltonian groups, are 2-nilpotent, so the fact that we can solve the HKP in such groups is already explained by the efficient quantum algorithm of Ivanyos et al. [37] for the HSP on 2-nilpotent groups. Hallgren et al. showed, however, that we can also find hidden normal subgroups and, hence, solve the HKP in any group, in principle, even those that are not nilpotent [32]. This is an important result, yet it remains poorly explained in the sense that normality—the necessary condition for their algorithm—seems specific to groups and does not connect to any broader characterization of why this problem should be solvable by quantum computers.

In this chapter, we provide an alternative explanation of why this problem can be solved efficiently on quantum computers, namely, that the HKP can be translated into a related problem on an abelian algebraic structure. In accordance with our general thesis, we will demonstrate that the latter problem can be solved efficiently on a quantum computer under reasonable assumptions. All together, this provides both a broader characterization of why the HKP is easy and also an example of how a problem on a non-abelian algebraic structure (the HKP) can be solved by translation into a related problem on an abelian one.

The chapter is organized as follows. We start in section 5.1 by reviewing the basic properties of hypergroups. Next, in section 5.2, we show how to reduce the HKP to the Hidden Subhypergroup Problem (HSHP). In section 5.3, we review some background on Fourier analysis of hypergroups.

We then give algorithms for solving the HSHP. In section 5.4, we give an efficient quantum algorithm for finding a hidden subhypergroup $\mathcal{N} \subset \mathcal{T}$ when $\mathcal{T}/\mathcal{N}$ is a group (Theorem 5.4). Combined with our result from section 5.2, this shows that the HSP is easy for class functions (Corollary 5.5).

Next, in section 5.5, we give an efficient quantum algorithm for the HSHP on strongly integral hypergroups assuming that we can implement QFTs on normal subgroups (Theorem 5.9). Combined with our results from section 5.2, this gives a new algorithm for the HKP and an alternative proof that the HKP is easy, assuming the existence of QFTs on normal subgroups (Corollary 5.10).

In section 5.6, we give an improved algorithm for a special class of hypergroups called ultragroups. We first give an efficient quantum algorithm for the HSHP on ultragroups assuming that we can implement certain group operations efficiently for the hypergroup (Theorem 5.13), which is potentially more plausible than assuming QFTs on normal subgroups. Combined with our results from section 5.2, this gives an efficient quantum algorithm with much weaker assumptions for the HKP on nilpotent groups (Corollary 5.15). Finally, this also allows us to show, under plausible assumptions, that we can solve the HSHP for the hypergroups of characters of nilpotent groups (Corollary 5.16), most of which are instances of the HSHP that cannot arise by reduction from the HKP (because the hypergroups are usually not strongly integral, for example) and, hence, cannot be solved by previously known algorithms.

We conclude in section 5.7.

5.1 Hypergroup Background

In this section, we review the basic properties of hypergroups, which were introduced by Dunkl [25], Jewett [38], and Spector [60]. (See [54, 63, 44] for introductions to hypergroups.) Although we will focus on applications of hypergroups to the study of groups, hypergroups arise in many other areas such as combinatorial optimization [23], cryptography [22], error correcting codes [22], particle physics [50], and even the study of non-abelian anyons in quantum computation [41, Appendix E]. (There are likely other connections to quantum computation beyond this last result and our own below since, as Litvinov noted, any space supporting a unitary Fourier transform "is necessarily connected with the existence of a certain hypergroup" [44].)

Hypergroups generalize groups by removing the restriction that the product of two element be *one* other element. Instead, they allow the product to be a *probability distribution* over elements.

More formally, let $\mathcal{T} = \{a_0, a_1, \dots, a_n\}$ be a set of elements. If $\mathcal{T}$ is a hypergroup, then, for any $i, j \in \{0, \dots, n\}$, we have an associative product[1]

$$a_i a_j = \sum_{k=0}^{n} n_{i,j}^{k} a_k, \tag{5.1}$$

where the numbers $n_{i,j}^{0}, \dots, n_{i,j}^{n}$ are non-negative reals summing to 1. The $n_{i,j}^{k}$'s are called the **structure constants** of $\mathcal{T}$ and uniquely define the hypergroup.

Like groups, every hypergroup has a (unique) identity element, conventionally

[1] Rather, this product must be associative once it is extended linearly to $\mathbb{R}\mathcal{T}$.

chosen to be element a_0. This satisfies $a_0 a_i = a_i = a_i a_0$ for $i \in \{0, \ldots, n\}$.

Hypergroups also have a generalization of the inverse element x^{-1} of a group, called the **anti-element**, which is unique and denoted by $\overline{x}$. The group inverse property, that $xx^{-1} = e$, is generalized to the anti-element property: $a_j = \overline{a_i}$ iff $n^0_{i,j} > 0$. In other words, element a_j is the anti-element of a_i iff the probability distribution for $a_i a_j$ (or $a_j a_i$) has the identity, a_0, in its support.

In summary, a hypergroup is a set of elements with (1) an associative product, yielding a probability distribution on elements; (2) an identity element; and (3), for each element, an anti-element with which the product has non-zero probability of producing the identity element. It is not hard to show that the identity and anti-elements are (as usual) always unique [54].

Many of our formulas will make use of the **weight** of an element $a \in \mathcal{T}$, which is defined to be $w_a := 1/n^0_{a,\overline{a}}$ and is always at least 1. In particular, $\overline{a}$ is a true inverse of a iff we have $a\overline{a} = a_0$, which holds iff $w_a = w_{\overline{a}} = 1$. In fact, the subset of elements of weight 1 not only all have inverses but they in fact form a group, called the group of **scalars**, which is the largest subset of the hypergroup that forms a group [54].

We can use the set $\mathcal{T}$ as the basis of a vector space, $\mathbb{C}\mathcal{T}$, which becomes a $\mathbb{C}$-algebra with product operation $\mathcal{T} \times \mathcal{T} \to \mathbb{C}\mathcal{T}$ as defined above for basis elements and extended to all of $\mathbb{C}\mathcal{T}$ by linearity. This is called the **hypergroup algebra**, denoted $\mathcal{A}(\mathcal{T})$, and is the natural generalization of a group algebra $\mathbb{C}G$, for a group G.

As with groups, a hypergroup $\mathcal{T}$ is called **abelian** if $ab = ba$ for all $a, b \in \mathcal{T}$. Every group G is also a hypergroup with algebra $\mathbb{C}G$, but this hypergroup is not abelian when G is a non-abelian group. Next, we will see two examples of different hypergroups associated to a group G that are always abelian, even when G is not.

5.1.1 The Hypergroup of Conjugacy Classes

In this section (and the next), we look at a class of hypergroups arising from groups. These will be especially important for us below. See [63] for a number of examples of hypergroups that arise in ways other than from groups.

Let G be a finite group. For any $g \in G$, recall that $C_g := \{g^a \mid a \in G\}$ is the conjugacy class of g. Let $\overline{G}$ be the set of distinct conjugacy classes of G.

Let $C = \{g_1, g_2, \dots\}$ and $D = \{h_1, h_2, \dots\}$ be two conjugacy classes. Then, for any product, $g_i h_j$, its conjugate $(g_i h_j)^a = g_i^a h_j^a$ is a product of conjugates, so it can be written as $g_k h_\ell$ for some k and ℓ. Furthermore, if there are M distinct products $g_{i_1} h_{j_1}, \dots, g_{i_M} h_{j_M}$ producing some element x, then the distinct products $g_{i_1}^a h_{j_1}^a, \dots, g_{i_M}^a h_{j_M}^a$ all produce x^a. Thus, for each conjugacy class E arising in the product of elements of C and D, we get a well defined number of "how many times" that class arises, which we denote $M_{C,D}^E$. Thus, we have something resembling a hypergroup product.[2]

To get a probability distribution, though, we must make a minor change. For each $C_g \in \overline{G}$, define c_g to be $(1/|C_g|)\, C_g$ and take $\{c_g \mid C_g \in \overline{G}\}$ to be a new set of elements. The structure constants become $m_{C,D}^E := M_{C,D}^E |E|/|C||D|$. Since the total number of products of elements formed multiplying C by D is $|C||D|$, we can see that $\sum_{E \in \overline{G}} M_{C,D}^E |E| = |C||D|$, which means that these new structure constants, $m_{C,D}^E$, define a probability distribution (indexed by E).

The identity element for this product is $c_e = \{e\}$. Furthermore, we can see that C_e arises in a product $C_g C_h$ iff C_h contains g^{-1}, which occurs iff $C_h = C_{g^{-1}}$. Thus, for each c_g, we have an anti-element $\overline{c_g} := c_{g^{-1}}$. Hence, the set of conjugacy classes defines a hypergroup with the c_g's as elements.

A quick calculation shows that the weight of c_g is $w_{c_g} = |C_g|$ for each $C_g \in \overline{G}$.

Finally, we note that this hypergroup is abelian, even if the underlying group is not abelian. To see this, we calculate $gh = hh^{-1}gh = hg^h = (hg)^h$ (since $h^h = h$), which shows that gh and hg are in the same conjugacy class. Hence, if we are multiplying conjugacy classes instead of elements, we do not distinguish between gh and hg, so we get an abelian structure.

5.1.2 The Hypergroup of Characters

Let $\mu, \sigma \in \widehat{G}$ be two irreducible representations of G. Recall that the pointwise product of their characters $\chi_\mu \chi_\sigma$ is the character of the representation $\mu \otimes \sigma$. This repre-

[2] Another way to think of this is to define C to be the sum $g_1 + g_2 + \dots$ in the group ring $\mathbb{Z}G$.

sentation is often not irreducible; however, it is equal to a direct sum of irreducible representations. Thus, we have $\chi_\mu \chi_\sigma = \sum_{\tau \in \widehat{G}} N^\tau_{\mu,\sigma} \chi_\tau$, where $N^\tau_{\mu,\sigma}$ is the number of copies of τ appearing in $\mu \otimes \sigma$, giving us something resembling a hypergroup product.

As above, we can scale these elements to get a probability distribution on outcomes. Let us define $\mathcal{X}_\mu = (1/d_\mu)\chi_\mu$ and take $\{\mathcal{X}_\mu \,|\, \mu \in \widehat{G}\}$ as the new set of elements. The structure constants become $n^\tau_{\mu,\sigma} := N^\tau_{\mu,\sigma} d_\tau / d_\mu d_\sigma$. Since the dimension of $\mu \otimes \sigma$ is $d_\mu d_\sigma$, we must have $\sum_{\tau \in \widehat{G}} N^\tau_{\mu,\sigma} d_\tau = d_\mu d_\sigma$. This implies that these new structure constants, $n^\tau_{\mu,\sigma}$, form a probability distribution (indexed by τ).

The identity element of this product is χ_{triv} since $\chi_{\text{triv}} \equiv 1$.

To find the anti-element, first recall that the coefficient $N^\tau_{\mu,\sigma}$ is explicitly given by $\langle \chi_\mu \chi_\sigma \,|\, \chi_\tau \rangle$, where $\langle \cdot \,|\, \cdot \rangle$ denotes the usual inner product on characters. From this, we can see that $N^{\text{triv}}_{\mu,\sigma} = \langle \chi_\mu \chi_\sigma \,|\, \chi_{\text{triv}} \rangle = \langle \chi_\mu \,|\, \overline{\chi_\sigma} \chi_{\text{triv}} \rangle = \langle \chi_\mu \,|\, \overline{\chi_\sigma} \rangle$. Since χ_μ and $\overline{\chi_\sigma}$ are both irreducible, this is 1 if $\chi_\sigma = \overline{\chi_\mu}$ and 0 otherwise, which shows that χ_{triv} arises in the product $\chi_\mu \chi_{\overline{\mu}}$ but not in any other product $\chi_\mu \chi_\sigma$. Hence, we can see $\overline{\mathcal{X}_\mu} := \mathcal{X}_{\overline{\mu}}$ is the anti-element of $\mathcal{X}_\mu$.

All together, this shows that the set of irreducible representations defines a hypergroup with the $\mathcal{X}_\mu$'s as its elements.

A quick calculation shows that the weight of $\mathcal{X}_\mu$ is $w_{\mathcal{X}_\mu} = d^2_\mu$.

Finally, we note that, in this case, our product is manifestly abelian since $\chi_\mu \chi_\sigma$ denotes the pointwise product of these functions and each value lies in $\mathbb{C}$, which is an abelian group.

5.1.3 Dual Hypergroups

For any abelian hypergroup $\mathcal{T}$, it is possible to define a dual hypergroup, $\mathcal{T}^*$, the elements of which are the **characters of $\mathcal{T}$**. These are the functions $\mathcal{X} : \mathcal{T} \to \mathbb{C}$ with the property that, once extended linearly to all of $\mathcal{A}(\mathcal{T})$, satisfy $\mathcal{X}(ab) = \mathcal{X}(a)\mathcal{X}(b)$ for every $a, b \in \mathcal{T}$.[3]

This construction can be performed for an arbitrary abelian hypergroup (see [54,

[3] Throughout this chapter, we will assume that all hypergroups are **strong**, which means that the structure constants of the characters are nonnegative and, hence, form a hypergroup.

65] for details). In the case of the class hypergroup $\overline{G}$, defined in the previous section, it can be shown that the dual hypergroup is precisely $\widehat{G}$ [54].

Since this construction can be performed for any abelian hypergroup, it can be applied to $\mathcal{T}^*$ to get $\mathcal{T}^{**}$. As is shown in [54, 65], this double dual is isomorphic to $\mathcal{T} \cong \mathcal{T}^{**}$. This isomorphism can be constructing canonically by taking $a \in \mathcal{T}$ to a function $\widetilde{\chi}_a$ defined by $\widetilde{\chi}_a(\mathcal{X}) = \mathcal{X}(a)$. In particular, this shows that both $\mathcal{A}(\mathcal{T})$ and $\mathcal{A}(\mathcal{T}^*)$ are vector spaces of the same dimension and, hence, the hypergroups $\mathcal{T}$, $\mathcal{T}^*$ have the same number of elements. In our case above, gives another proof that the number of conjugacy classes is equal to the number of irreducible characters.

For any subset $\mathcal{S} \subset \mathcal{T}$, we define $\varpi_{\mathcal{S}} = \sum_{a \in \mathcal{S}} w_a$ to be the sum of the weights of the elements in the subset. In particular, $\varpi_{\mathcal{T}}$ is the total weight of all the elements in the hypergroup.

For any hypergroup $\mathcal{T}$, it is always the case that $\varpi_{\mathcal{T}} = \varpi_{\mathcal{T}^*}$ [65]. For our examples above, this tells us that $\varpi_{\overline{G}} = \varpi_{\widehat{G}}$. We can see that the former is $\sum_{C_g \in \overline{G}} |C_g| = |G|$ and the latter is $\sum_{\mu \in \widehat{G}} d_\mu^2$, so this gives a proof of the familiar fact that $\sum_{\mu \in \widehat{G}} d_\mu^2 = |G|$.

As with groups, the characters of abelian hypergroups are orthogonal under a natural inner product. For any $\mathcal{X}_\mu, \mathcal{X}_\nu \in \mathcal{T}^*$, we define

$$\langle \mathcal{X}_\mu \,|\, \mathcal{X}_\nu \rangle := \sum_{a \in T} \frac{w_{\mathcal{X}_\nu} w_a}{\varpi_{\mathcal{T}}} \overline{\mathcal{X}_\mu}(a) \mathcal{X}_\nu(a), \tag{5.2}$$

Then, it is possible to show that $\langle \mathcal{X}_\mu \,|\, \mathcal{X}_\nu \rangle = \delta_{\mu,\nu}$ [65].

The fact that characters are orthogonal allows us to also define a quantum Fourier transform (QFT) for any hypergroup $\mathcal{T}$ by

$$\mathcal{F}_{\mathcal{T}} \,|a\rangle := \sum_{\mathcal{X}_\mu \in \mathcal{T}^*} \sqrt{\frac{w_{\mathcal{X}_\mu} w_a}{\varpi_{\mathcal{T}^*}}} \mathcal{X}_\mu(a) |\mathcal{X}_\mu\rangle.$$

The character orthogonality relation, equation (5.2), implies that this is unitary.

(In addition to the QFT on $\mathcal{T}$, it is possible to define hypergroup analogues of other natural classes of gates defined over abelian groups. Furthermore, Van Den Nest's vast generalization of the Gottesman-Knill Theorem to arbitrary abelian groups [19] can also be extended to hypergroups. See [12] for details.)

5.1.4 Subhypergroups and Quotient Hypergroups

Analogous to a subgroup, a **subhypergroup** is a subset of the hypergroup that is closed under taking products and anti-elements.

If $\mathcal{N} \subset \mathcal{T}$ is a subhypergroup, then, for each $a \in \mathcal{T}$, we can define a coset $a\mathcal{N} := \{x \in \mathcal{T} \,|\, x \in ab \text{ for some } b \in \mathcal{N}\}$.[4] It is possible to show that, for every $a, b \in \mathcal{T}$, the cosets $a\mathcal{N}$ and $b\mathcal{N}$ are either equal or disjoint [54]. This gives some hint that it should be possible to define a product on hypergroup cosets like we can do on group cosets.

To do this, we identify $a\mathcal{N}$ with the element of $\mathcal{A}(\mathcal{T})$ given by $\sum_{x\in a\mathcal{N}}(w_x/\varpi_{a\mathcal{N}})x$. It is possible to show that the products of these elements in $\mathcal{A}(\mathcal{T})$ define a hypergroup with structure constants

$$r_{a\mathcal{N},b\mathcal{N}}^{c\mathcal{N}} := \sum_{d\in c\mathcal{N}} n_{a,b}^{d}, \tag{5.3}$$

for each $a, b, c \in \mathcal{T}$, and where the anti-element of $a\mathcal{N}$ is $\bar{a}\mathcal{N}$ [54]. This is called the **quotient hypergroup** and denoted $\mathcal{T}/\mathcal{N}$.

It is also possible to show that the weight of $a\mathcal{N}$ is $w_{a\mathcal{N}} = \varpi_{a\mathcal{N}}/\varpi_{\mathcal{N}}$ [54], a formula that will be useful to us later on.

5.2 Reducing the HKP to the HSHP

Hallgren et al. showed that any normal subgroup N is the intersection of the kernels of those irreducible representations whose kernels contain N [32]. This means that the kernels of irreducible representations contain all information about the normal subgroups of the group. Hence, it should not be surprising that we can find hidden normal subgroups (the kernels of homomorphisms) using only information about the hypergroup $\overline{G}$, a fact that we will prove computationally in this section. As $\overline{G}$ is an *abelian* hypergroup, this will demonstrate that we can find normal subgroups by solving a related problem in the abelian algebraic structure $\overline{G}$.

[4] Here and below, when we treat a probability distribution on elements as a set, we are referring to the set of elements in the support of that probability distribution.

A natural problem on hypergroups is the hidden subhypergroup problem (HSHP), first considered in [2]. In the HSHP, we are given, as an oracle, a hiding function $f : \mathcal{T} \to \{0,1\}^*$ assigning labels to elements of the hypergroup that is promised to hide some subhypergroup $\mathcal{N} \subseteq \mathcal{T}$, meaning that $f(x') = f(x)$ iff $n_{x,h}^{x'} > 0$ (i.e., xh can produce x') for some $h \in \mathcal{N}$. The problem asks us to determine $\mathcal{N}$ via oracle queries and quantum computation.

A subgroup is normal precisely when it is the union of a set of conjugacy classes (whereas a non-normal subgroup is one that contains part of some conjugacy class but not all of it), so a normal subgroup $N \lhd G$ consists of a set of conjugacy classes of G, which we denote $\overline{N}_G$. Since normal subgroups are, in particular, subgroups, we know that they contain the identity and inverses and are closed under products. Thus, the set of conjugacy classes $\overline{N}_G$ must contain c_e, must contain $c_{n^{-1}}$ for every $n \in N$, and must be closed under products. In other words, the set $\overline{N}_G$ also defines a hypergroup, which is a subhypergroup of $\overline{G}$. This means that a normal subgroup N can be not only the solution to some HKP instance but also, in the form of $\overline{N}_G$, the solution to some HSHP instance.

Below, we will formalize this argument into a reduction of the HKP to the HSHP. In order to perform such a reduction, we must provide an oracle for the HSHP on the hypergroup $\overline{G}$. Our construction requires us to assume that we can perform certain computations with conjugacy classes, as laid out in the following definition.

Definition 5.1. *Consider the following operations for working with conjugacy classes:*

- *Given a conjugacy class C, produce the size of this class, $|C|$.*
- *Given an $x \in G$, produce the pair (C, j), where $x = x_j$ in class $C = \{x_1, \ldots, x_t\}$.*
- *Given a pair (C, j), produce the element x_j from C.*

If each of these operations can be performed efficiently, then we say that we can ***compute efficiently with conjugacy classes*** *of G.*

We note that this assumption is trivial for abelian groups since each element is in its own conjugacy class. For some common examples of non-abelian groups, such as the dihedral and Heisenberg groups (and their higher nilpotent generalizations), elements are normally encoded in this manner already, so no additional assumption is actually required. For other common examples like the symmetric group, while elements are not always encoded directly in this manner, it is not hard to show that the above calculations can be performed efficiently. In general, while we must formally make this assumption, we are not aware of any group for which these calculations cannot be performed efficiently.

With this definition in hand, we can now state and prove our reduction.

Theorem 5.2 (HKP $\leq$ HSHP). *Let G be a group. Suppose that $f : G \to K$ is a homomorphism into a group K. If we can compute efficiently with conjugacy classes of G and K, then we can efficiently reduce this HKP to the HSHP on $\overline{G}$.*

Proof. Consider any element $x \in G$. For any conjugate x^g, for some $g \in G$, we see that $f(x^a) = f(a^{-1}xa) = f(a^{-1})f(x)f(a)$ since f is a homomorphism. Furthermore, since $a^{-1}a = e$, we see that $f(a^{-1})f(a) = f(e) = e$, which shows that $f(a^{-1}) = f(a)^{-1}$. Putting these together, we have $f(x^a) = f(a)^{-1}f(x)f(a) = f(x)^{f(a)}$. This means that the function $\tilde{f}$ taking x to the conjugacy class label of $f(x)$ is a well-defined class function, $\overline{G} \to \overline{K}$, whose kernel is the same as that of f (since $x \in C_e \Rightarrow x = e$).

Let N be the kernel of $\tilde{f}$. Using the facts that $\tilde{f}$ is a class function, that f hides N, and that N is normal, we can see that $\tilde{f}(xn) = \tilde{f}((xn)^a) = \tilde{f}(x^a n^a) = \tilde{f}(x^a) = \tilde{f}(x^a n')$ for any $n, n' \in N$.[5] This shows that $\tilde{f}$ is constant on CN, for any $C \in \overline{G}$, which is a coset of the subhypergroup $\overline{N}_G$. This shows that $\tilde{f}$ is a hiding function for $\overline{N}_G$.

It remains only to show that we can compute $\tilde{f}$ efficiently. Given a conjugacy class $C \in \overline{G}$, since we can efficiently compute with conjugacy classes of G, we can translate the pair $(C, 1)$ into an element $x \in C$. Then, we invoke f to get an element $f(x) \in K$. Since we can efficiently compute with conjugacy classes of K, we can translate $f(x)$ into a conjugacy class $D \in \overline{K}$ with $f(x) \in D$. The value of $\tilde{f}(C)$ is D, and we have seen that this be computed efficiently. □

While the HKP is perhaps the most natural setting for the problem of finding

[5] Note that $f(xn) = f(x)$ implies that $\tilde{f}(xn) = \tilde{f}(x)$ since $\tilde{f}$ only makes more elements equal.

hidden normal subgroups, since the kernel of a homomorphism is always normal, it is also possible to pose such problems as instances of the HSP if the hidden subgroup N is *promised* to be normal. While these require a promise, they have the advantage of being provably hard classically, whereas the HKP requires assumptions about the hardness of certain cryptographic primitives. The next theorem gives a reduction that applies in the HSP setting.

The theorem applies to the HSP with the assumption that the hiding function f is a class function. This will occur iff G/N is abelian, where N is the hidden subgroup. This problem is provably hard classically since it includes the abelian HSP, which is known to be hard, as a special case.

Theorem 5.3 (class function HSP $\leq$ HSHP). *Let G be a group. Suppose that we are given a hiding function $f : G \to \{0,1\}^*$ that is also a class function. If we can compute efficiently with conjugacy classes of G, then we can efficiently reduce this HSP to the HSHP on the hypergroup $\overline{G}$.*

Proof. As we saw in the previous proof, the assumptions about computing efficiently with conjugacy classes of G imply that, given a conjugacy class C, we can efficiently find an element $x \in C$ and apply f to get a label. Since f is a class function, the label would be the same for any $x' \in C$, so we have a well-defined function $\overline{G} \to \{0,1\}^*$. The kernel of this map is the same as that of f, and, by the same argument as in the previous proof, this is a hiding function for the subhypergroup $\overline{N}_G$, which we have seen can be computed efficiently. □

5.3 Abelian Hypergroup Duality

In the last section, we saw that the HKP, even for non-abelian groups, is reducible to a problem on an *abelian* hypergroup. We will see in this section that, even though hypergroup product operations are more complex, we gain much from the fact that they are commutative. In particular, an abelian hypergroup and its dual hypergroup of characters exhibit the same relationships that were used in solving the HSP for abelian groups. Let us now describe these in more detail.

For any subhypergroup $\mathcal{N} \subset \mathcal{T}$, the **annihilator** is the subgroup of $\mathcal{T}^*$ given by

$$\mathcal{N}^\perp := \{\mathcal{X}_\mu \in \mathcal{T}^* \,|\, \mathcal{X}_\mu(a) = 1 \;\forall a \in \mathcal{N}\}.$$

As with abelian groups, the annihilator $\mathcal{N}^\perp$ is isomorphic to the hypergroup of characters of $\mathcal{T}/\mathcal{N}$.[6] More importantly, the hypergroup of characters of $\mathcal{N}$ is isomorphic to $\mathcal{T}^*/\mathcal{N}^\perp$. As with abelian groups, knowledge of $\mathcal{N}^\perp$ is sufficient to identify $\mathcal{N}$, as we have $(N^\perp)^\perp \cong N$.[7]

Just as with abelian groups, the annihilator arises when we apply the Fourier transform to a coset state. For hypergroups, the latter is a state of the form

$$|a\mathcal{N}\rangle := \sum_{x \in a\mathcal{N}} \sqrt{w_x/\varpi_{a\mathcal{N}}}\,|x\rangle, \tag{5.4}$$

for some $a \in \mathcal{T}$. It follows from [12, Theorem 4] that

$$\mathcal{F}_\mathcal{T}\,|a\mathcal{N}\rangle = \sum_{\mathcal{X}_\mu \in \mathcal{N}^\perp} \sqrt{\frac{w_{\mathcal{X}_\mu} w_{a\mathcal{N}}}{\varpi_{\mathcal{N}^\perp}}}\mathcal{X}_\mu(a)|\mathcal{X}_\mu\rangle. \tag{5.5}$$

Note the sum is *only* over those $\mathcal{X}_\mu \in \mathcal{N}^\perp$.

The aforementioned theorem from [12] applies to a hypergroup generalization of so-called CSS stabilizer states. These are states that are uniquely described by two mutually-commuting *hypergroups* of operators that act on the state in a simple way. The theorem describes how the Fourier transform maps one CSS stabilizer state to another. In this case, a coset state is uniquely described by its behavior under the hypergroup of operators that perform multiplication by elements of $\mathcal{N}$ and the hypergroup of operators that apply a fixed character function from $\mathcal{N}^\perp$. In particular, it is stabilized by the former hypergroup. After the Fourier transform, we have a state that stabilized by the Fourier transform of the former hypergroup operators, which are operators that evaluate characters at elements of $\mathcal{N}$. This means that the resulting

[6] The quotient can be defined in any hypergroup [54]. The elements of this hypergroup are the distinct cosets of the form $a\mathcal{N}$, for $a \in \mathcal{T}$.

[7] This is an important property of abelian groups that is maintained by abelian hypergroups but not by non-abelian groups.

state is supported only on elements of $\mathcal{N}^\perp$.

5.4 Solving the HSHP: Easy Case

Given the relationships outlined in the previous section, the first approach that comes to mind for solving the HSHP is to mirror the Fourier sampling technique that works for abelian groups. Let us now describe that approach, which we call Hypergroup Fourier Sampling (HFS), in detail.

We will work in a Hilbert space of the form $V_\mathcal{T} \otimes W$, where the elements of $\mathcal{T}$ index a basis for $V_\mathcal{T}$ and those of $\{0,1\}^m$ (where our hiding function maps $f : \mathcal{T} \to \{0,1\}^m$) index a basis for W. If we assume that we can implement the QFT $\mathcal{F}_\mathcal{T} : V_\mathcal{T} \to V_{\mathcal{T}^*}$, then it is reasonable to assume that we can also work with the Hilbert space $V_{\mathcal{T}^*}$.

We perform the following steps.

1. Initialize the state to $|\mathcal{X}_{\text{triv}}\rangle \otimes |0^m\rangle$.

2. Apply the inverse QFT to prepare a superposition over elements of $\mathcal{T}$.

$$\begin{aligned}(\mathcal{F}_\mathcal{T}^\dagger \otimes I)|\mathcal{X}_{\text{triv}}\rangle \otimes |0^m\rangle &= \sum_{a\in\mathcal{T}} \sqrt{\frac{w_a w_{\mathcal{X}_{\text{triv}}}}{\varpi_\mathcal{T}}}\,\overline{\mathcal{X}_{\text{triv}}(a)}|a\rangle \otimes |0^m\rangle \\ &= \sum_{a\in\mathcal{T}} \sqrt{\frac{w_a}{\varpi_\mathcal{T}}}|a\rangle \otimes |0^m\rangle\end{aligned}$$

3. Apply the oracle for f, giving us the state

$$\sum_{a\in T} \sqrt{\frac{w_a}{\varpi_\mathcal{T}}}|a\rangle \otimes |f(a)\rangle.$$

4. Measure the second Hilbert space.

   This yields a coset state $|a\mathcal{N}\rangle$, as in equation (5.4), where $\mathcal{N} \subset \mathcal{T}$ is the subhypergroup hidden by f. The probability of seeing the particular state $|a\mathcal{N}\rangle$ is equal to $\varpi_{a\mathcal{N}}/\varpi_\mathcal{T} = w_{a\mathcal{N}}$.

5. Apply the QFT to the remaining state.

The result is the state in equation (5.5), supported only on characters of $\mathcal{N}^\perp$.

6. Measure the remaining state.

   From equation (5.5), the probability of measuring $\mathcal{X}_\mu \in \mathcal{N}^\perp$ is proportional to $w_{\mathcal{X}_\mu}|\mathcal{X}_\mu(a)|^2$ for a fixed choice of $a\mathcal{N} \in \mathcal{T}/\mathcal{N}$. Combining with the distribution on $a\mathcal{N}$ from step 4, we can see that the overall probability of measuring $\mathcal{X}_\mu$ is

$$\begin{aligned}\Pr[\mathcal{X}_\mu \in \mathcal{N}^\perp] &= \sum_{a\mathcal{N}\in\mathcal{T}/\mathcal{N}} \frac{\varpi_{a\mathcal{N}}}{\varpi_{\mathcal{T}}} \frac{w_{\mathcal{X}_\mu} w_{a\mathcal{N}}}{\varpi_{\mathcal{N}^\perp}} |\mathcal{X}_\mu(a)|^2 \\ &= w_{\mathcal{X}_\mu} \sum_{a\mathcal{N}\in\mathcal{T}/\mathcal{N}} \frac{w_{a\mathcal{N}}^2}{\varpi_{\mathcal{T}/\mathcal{N}}^2} |\mathcal{X}_\mu(a)|^2, \qquad (5.6)\end{aligned}$$

   where we have used the fact that $N^\perp \cong (\mathcal{T}/\mathcal{N})^*$ and $\varpi_{\mathcal{T}} = \varpi_{\mathcal{T}^*}$.

This approach was first proposed by Amini et al. [2], though the analysis above is from [12]. Below, we will see that, unlike what happens with groups, we cannot always find $\mathcal{N}$ simply by intersecting the kernels of poly log $|\mathcal{T}|$ random samples of $\mathcal{X}_\mu \in \mathcal{N}^\perp$ drawn from the above distribution. However, we will begin here with a positive case.

Theorem 5.4. *If there is an efficient quantum circuit implementing the QFT of $\mathcal{T}$, then the HSHP on $\mathcal{T}$ can be solved efficiently whenever the hidden subgroup $\mathcal{N}$ is such that $\mathcal{T}/\mathcal{N}$ is a group.*

Proof. If $\mathcal{T}/\mathcal{N}$ is a group, then $w_{a\mathcal{N}} = 1$ since all weights in a group are 1. The same is true of $w_{\mathcal{X}_\mu}$ since $(\mathcal{T}/\mathcal{N})^*$ is then also a group. Thus, by equation (5.6), we get a uniform distribution over $\mathcal{N}^\perp$, just as occurs in the HSP for abelian groups.

Hence, the standard results on the abelian HSP (discussed in section 3.3.1) tell us not only that $O(\log|\mathcal{T}/\mathcal{N}|) = O(\log|\mathcal{T}|)$ samples are sufficient to uniquely determine $\mathcal{N}$ (as the intersection of the kernels of the measured characters) but also that this intersection can be computed efficiently.

Specifically, the kernel of every measured character must include the **core of $\mathcal{T}$** [54], which is the (unique) smallest subhypergroup $\mathcal{K} \subset \mathcal{T}$ such that $\mathcal{T}/\mathcal{K}$ is a group:

since $\mathcal{T}/\mathcal{N}$ is a group, by assumption, we must have $\mathcal{K} \subset \mathcal{N}$ (since $\mathcal{K}$ is the smallest subhypergroup for which this holds), which means $\mathcal{N}^{\perp} \subset \mathcal{K}^{\perp}$. Hence, we may use the abelian group techniques to find $\mathcal{N}/\mathcal{K}$ inside of the group $\mathcal{T}/\mathcal{K}$. Adjoining generators of $\mathcal{K}$ to (the inverse images of) those of $\mathcal{N}/\mathcal{K}$ gives a generating set for $\mathcal{N}$. □

Putting this together with our reduction from Theorem 5.3 gives an alternative proof that we can efficiently solve the HSP when the hiding function is a class function.

Corollary 5.5. *(class function HSP is easy) Let G be a group. Suppose that we are given a hiding function $f : G \rightarrow \{0,1\}^*$ that is also a class function. If there is efficient quantum circuit implementing the QFT of G and we can compute efficiently with conjugacy classes of G, then there is an efficient quantum circuit for the HSP.*

Proof. This follows immediately from Theorems 5.3 and 5.4 provided that we have an efficient quantum circuit implementing the QFT for $\overline{G}$. The latter exists, since we have an efficient quantum circuit for the QFT of G, by the following lemma. □

Lemma 5.6. *If there is an efficient QFT for a group G, then, with the appropriate choice of basis for $\overline{G}$ and $\widehat{G}$, there is an efficient QFT for the hypergroup $\overline{G}$ (and $\widehat{G}$).*

Proof. Let us identify the state $|C_x\rangle$ with the superposition $|C_x|^{-1/2}\sum_{g\in C_x}|g\rangle$. The Fourier transform of such a state is

$$\begin{aligned}
\mathcal{F}_G\,|C_x\rangle &= \frac{1}{\sqrt{|C_x|}}\sum_{g\in C_x}\mathcal{F}_G\,|g\rangle \\
&= \frac{1}{\sqrt{|C_x|}}\sum_{g\in C_x}\frac{1}{\sqrt{|G|}}\sum_{\rho\in\widehat{G}} d_\rho|\rho\rangle\otimes\sum_{i,j=1}^{d_\rho}\frac{[\rho(g)]_{i,j}}{\sqrt{d_\rho}}|i,j\rangle \\
&= \frac{1}{\sqrt{|C_x||G|}}\sum_{\rho\in\widehat{G}} d_\rho|\rho\rangle\otimes\sum_{i,j=1}^{d_\rho}\frac{[\sum_{g\in C_x}\rho(g)]_{i,j}}{\sqrt{d_\rho}}|i,j\rangle
\end{aligned}$$

Now, by the Orbit-Stabilizer Theorem [24], the sum $\sum_{g\in C_x}\rho(g)$ can be written as $(|C_x|/|G|)\sum_{h\in G}\rho(g^h)$. By Corollary A.4, the latter sum is $(|C_x|\operatorname{tr}\rho(g)/d_\rho)\,\mathrm{I} =$

$(|C_x|\chi_\rho(g)/d_\rho)\,\mathrm{I}$. Thus, we can see that

$$\begin{aligned}\mathcal{F}_G\,|C_x\rangle &= \frac{1}{\sqrt{|C_x|}}\sum_{g\in C_x}\frac{1}{\sqrt{|G|}}\sum_{\rho\in\widehat{G}} d_\rho|\rho\rangle\otimes\sum_{i=1}^{d_\rho}\frac{|C_x|\chi_\rho(g)}{d_\rho\sqrt{d_\rho}}|i,i\rangle\\ &= \sqrt{\frac{|C_x|}{|G|}}\sum_{\rho\in\widehat{G}} d_\rho\frac{\chi_\rho(g)}{d_\rho}|\rho\rangle\otimes\sum_{i=1}^{d_\rho}\frac{1}{\sqrt{d_\rho}}|i,i\rangle,\end{aligned}$$

which can be written in hypergroup notation as

$$\mathcal{F}_G\,|C_x\rangle = \sum_{\mathcal{X}_\rho\in\widehat{G}}\sqrt{\frac{w_{C_x}w_{\mathcal{X}_\rho}}{w_{\overline{G}}}}\mathcal{X}_\rho(C_x)\left(\frac{1}{\sqrt{d_\rho}}\sum_{i=1}^{d_\rho}|\rho,i,i\rangle\right).$$

This matches the definition of $\mathcal{F}_{\overline{G}}$ if we identify $|\mathcal{X}_\rho\rangle$ with $d_\rho^{-1/2}\sum_{i=1}^{d_\rho}|\rho,i,i\rangle$.

Thus, we can see that, if we identify $|c_x\rangle$ with $|C_x|^{-1/2}\sum_{g\in C_x}|g\rangle$ and $|\mathcal{X}_\rho\rangle$ with $d_\rho^{-1/2}\sum_{i=1}^{d_\rho}|\rho,i,i\rangle$, the Fourier transform of G implements that of $\overline{G}$. □

It is worth noting that, if we identify the states $|c_x\rangle$ and $|\mathcal{X}_\rho\rangle$ as in the previous lemma, then the intermediate states of the algorithm considered above for the HSHP are actually identical to those of the algorithm of Hallgren et al. for the HSP when the hidden subgroup is normal.

We should also verify that we can efficiently prepare and measure in these bases. This is possible for the $|\mathcal{X}_\rho\rangle$'s under standard assumptions. (See [12] for details.) Although not needed for the above algorithm, this is also possible for the $|c_x\rangle$'s under the assumption that we can compute efficiently with conjugacy classes. (Indeed, the assumption that we can compute efficiently with conjugacy classes largely amounts to assuming that we can efficiently perform the same operations in the hypergroup of conjugacy classes that are typically assumed for the dual hypergroup of characters.)

5.5 Solving the HSHP: General Case

Let us now demonstrate that the above approach does not work for all hypergroups. Indeed, it does not even work for all hypergroups arising from groups.

Example 5.7. *Our example will be the hypergroup of conjugacy classes of the Heisenberg group defined in section 2.1.4. This group is 2-nilpotent. Its center consists of elements of the form* $(0, y, 0)$ *for any* $y \in \mathbb{Z}_p$. *For any* $(x, z) \neq (0, 0)$, *the conjugacy class of* (x, y, z) *is* $C_{(x,y,z)} = (x, 0, z)Z(G)$, *which contains* p *elements. Each element of the center must be in its own conjugacy class.*

Now, suppose that the hidden subhypergroup is $\{c_e\}$, *and let us compute the probability of measuring* $\mathcal{X}_{\chi_{\alpha,\gamma}}$, *where* $\chi_{\alpha,\gamma} : G \to \mathbb{C}$ *is a 1-dimensional representation of* G *of the form* $\chi_{\alpha,\gamma}(x, y, z) = \omega_p^{\alpha x + \gamma z}$. *By equation (5.6), since* $|\mathcal{X}_{\chi_{\alpha,\gamma}}(a)| = 1$, *we can see that the probability of measuring* $\mathcal{X}_{\chi_{\alpha,\gamma}}$ *is just* $\sum_{c \in \overline{G}} w_c^2 / \varpi_{\overline{G}}^2 = (\sum_{c \in \overline{G}} |C|^2)/|G|^2$. *Since there are* $p^2 - 1$ *classes of size* p *and* p *class of size* 1, *we can see that the above sum is* $((p^2 - 1) \cdot p^2 + p \cdot 1^2)/p^6 = 1/p^2 - O(1/p^4)$.

All together, there are p^2 *1-dimensional representations of the form* $\chi_{\alpha,\gamma}$, *so the probability of measuring any 1-dimensional representation is* $p^2(1/p^2 - O(1/p^4)) = 1 - O(1/p^2)$. *Now, since* $|G|$ *is exponentially large,* p *must be exponentially large, which means that we will only ever measure such representations. However, every such representation contains* $Z(G)$ *in its kernel, so this shows that the intersection of the kernels of* poly log $|G|$ *samples will still contain* $Z(G)$ *with high probability, which is strictly larger than the hidden subgroup* $\{c_e\}$.

While this rules out the idea of simply intersecting the kernels of the characters that we measure, it seems possible that an algorithm similar to those we considered in an earlier section may work here. We will see below that this is indeed the case. The key to proving this is the following lemma.

The lemma applies to a class of hypergroups that includes the conjugacy class hypergroup of any group. We will say that a hypergroup $\mathcal{T}$ is **strongly integral** if the weight w_a, for any $a \in \mathcal{T}$, is not only an integer but also divides $\varpi_{\mathcal{T}}$.

Lemma 5.8. *Let* $\mathcal{N} \subsetneq \mathcal{T}$ *be the hidden subgroup. If* $\mathcal{T}/\mathcal{N}$ *is strongly integral, then the probability of measuring* $\mathcal{X}_{\text{triv}}$ *in HFS is at most* $1/2$.

Proof. Using the facts that $w_{\mathcal{X}_{\text{triv}}} = 1$ and $\mathcal{X}_{\text{triv}} \equiv 1$, we can see by equation (5.6) that the probability of measuring $\mathcal{X}_{\text{triv}}$ is equal to $\sum_{a\mathcal{N} \in \mathcal{T}/\mathcal{N}} w_{a\mathcal{N}}^2 / \varpi_{\mathcal{T}/\mathcal{N}}^2$. Defining

$\mathcal{T}' := \mathcal{T}/\mathcal{N}$, we can rewrite this as $\sum_{a \in \mathcal{T}'} w_a^2/\varpi_{\mathcal{T}'}^2$.

Let $s_a := w_a/\varpi_{\mathcal{T}'}$. By assumption, w_a divides $\varpi_{\mathcal{T}'}$, so $s_a \leq 1/2$. We also know that $\sum_{a \in \mathcal{T}'} s_a = 1$ by the definition of $\varpi_{\mathcal{T}'}$. Hence, we can upper bound this probability by the maximum value of $\sum_{i=1}^n s_i^2$ subject to the constraints $\sum_{i=1}^n s_i = 1$ and $s_i \leq 1/2$ for all $i \in \{1, \dots n\}$, where each s_i is now an arbitrary nonnegative real. We can see that the latter is at most $\sum_{i=1}^n s_i^2 \leq \sum_{i=1}^n s_i \frac{1}{2} = \frac{1}{2} \sum_{i=1}^n s_i = \frac{1}{2}$.

Thus, the probability of measuring $\mathcal{X}_{\text{triv}}$ is bounded by $1/2$. □

With that in hand, we can now prove the following result, which applies, in particular, whenever $\mathcal{T}$ is the conjugacy class hypergroup of a group.

Theorem 5.9. *Let $\mathcal{T}$ be a hypergroup such that the weights of both $\mathcal{T}$ and $\mathcal{T}^*$ are integral. If, for any subhypergroup $\mathcal{K} \subset \mathcal{T}$ containing the hidden subgroup $\mathcal{N}$, there is an efficient quantum circuit implementing the QFT of $\mathcal{K}$ and the quotient hypergroup $\mathcal{K}/\mathcal{N}$ is strongly integral, then there is an efficient quantum circuit solving the HSHP.*

Proof. By the previous lemma, if $\mathcal{N} \neq \mathcal{T}$, then we measure $\mathcal{X}_{\text{triv}}$ with probability at most $1/2$. Hence, after $O(\log |\mathcal{T}|)$ samples, we will draw some $\mathcal{X}_\rho$ with $\rho \neq$ triv with high probability. We can then recurse on the subhypergroup $\mathcal{K} = \operatorname{Ker} \mathcal{X}_\rho$ since the hiding function for $\mathcal{T}$ becomes a hiding function for $\mathcal{K}$ by restriction and since, by assumption, there is an efficient QFT for $\mathcal{K}$ as well.

By Lagrange's Theorem for hypergroups [64], the weight of $\mathcal{K}$ divides $\mathcal{T}$, so, each time we recurse on a subhypergroup, it is smaller in weight by at least a factor of $1/2$. Thus, it takes at most $\log |\mathcal{T}|$ iterations to reach $\mathcal{K} = \mathcal{N}$.

At that point, the above algorithm will only measure $\mathcal{X}_{\text{triv}}$. Hence, after $O(\log |\mathcal{T}|)$ samples, all of which are $\mathcal{X}_{\text{triv}}$, we know with high probability that the current subhypergroup $\mathcal{K}$ is $\mathcal{N}$. □

Putting this together with our reduction from Theorem 5.2 gives an alternative proof that we can efficiently solve the HKP.

Corollary 5.10 (HKP is easy). *Let G be a group. Suppose that we are given a homomorphism $f : G \to H$ hiding a subgroup $N \lhd G$. If we can efficiently compute* *the QFT of any subgroup $K \lhd G$ with $N \leq K$ and compute efficiently with conjugacy classes of G and K, then there is an efficient quantum circuit solving the HKP.*

Proof. The restriction $f|_K : K \to H$ is itself a group homomorphism (since f is), so by our assumptions, Theorem 5.2, and Lemma 5.6, we can efficiently reduce to an instance of the HSHP covered by Theorem 5.9 above. □

As noted above, the assumptions about computing with conjugacy classes are analogues for the conjugacy class hypergroup of facts that are frequently assumed for the hypergroup of characters—plus, we are unaware of any group for which these assumptions do not hold—so we do not see these assumptions as particularly onerous.

Recall that the algorithm of Hallgren et al., discussed in section 3.3.2, performs weak Fourier sampling only in the group G. Then, to find $\operatorname{Ker} f$, it computes the intersection of the kernels of all characters that appeared as measurement outcomes. In contrast, by recursing on smaller subgroups, our algorithm avoids having to compute intersections of kernels, replacing this with the assumption that we can perform QFTs over each of these subgroups.

In the next section, we will see that these assumptions can be even further reduced for nilpotent groups.

5.6 Solving the HSHP: Ultragroups

The algorithm of the previous section has two limitations that may worry us. First, it requires us to assume that we can implement the QFT for every subhypergroup $\mathcal{K} \subset \mathcal{T}$. That is not a real obstacle in principle (as we will see), but it could present some practical difficulties. Second, the previous algorithm requires all of the quotient hypergroups $\mathcal{K}/\mathcal{N}$ that arise in the algorithm to be strongly integral. The latter holds for conjugacy class hypergroups, but, as far as we know, those might be the only hypergroups for which it holds. For example, this assumption does not hold for the hypergroup of characters of a group.

In this section, we will describe an algorithm that removes both of these limitations. In particular, our algorithm will work for solving the HSHP on hypergroups that are not conjugacy classes of groups. Hence, this algorithm can be used to solve instances of the HSHP that do not arise from the HKP on groups. In order to do this, however, we will need to make some other assumptions about the hypergroup, which we will discuss later on.

First, let us show how we can remove the assumption that there is an efficient circuit for the QFT of every subhypergroup $\mathcal{K} \subset \mathcal{T}$.

Lemma 5.11. *Let $\mathcal{K} \subset \mathcal{T}$ be a subhypergroup. If there is an efficient QFT for $\mathcal{T}$, then, with an appropriate choice of bases, there is an efficient QFT for $\mathcal{K}$.*

Proof. As noted above, we always have $\mathcal{K}^* \cong \mathcal{T}^*/\mathcal{K}^\perp$, so the cosets of $\mathcal{K}^\perp$ are a natural basis for the characters of $\mathcal{K}$. In detail, this basis is

$$|\mathcal{X}_\nu\mathcal{K}^\perp\rangle := \sum_{\mathcal{X}_\mu\in\mathcal{X}_\nu\mathcal{K}^\perp} \sqrt{\frac{w_\mu}{\varpi_{\mathcal{X}_\nu\mathcal{K}^\perp}}}|\mathcal{X}_\mu\rangle = \sum_{\mathcal{X}_\mu\in\mathcal{X}_\nu\mathcal{K}^\perp} \sqrt{\frac{w_{\mathcal{X}_\mu}\varpi_{\mathcal{K}}}{w_{\mathcal{X}_\nu\mathcal{K}^\perp}\varpi_{\mathcal{T}}}}|\mathcal{X}_\mu\rangle,$$

where, in the second equality, we have used the definition $w_{\mathcal{X}_\mu\mathcal{K}^\perp} := \varpi_{\mathcal{X}_\mu\mathcal{K}^\perp}/\varpi_{\mathcal{K}^\perp}$ and the relation above, which tells us that $\varpi_{\mathcal{T}}/\varpi_{\mathcal{K}^\perp} = \varpi_{\mathcal{K}^*} = \varpi_{\mathcal{K}}$.

Now, since any $\mathcal{X}_\mu \in \mathcal{X}_\nu\mathcal{K}^\perp$ has the same value on $a \in \mathcal{K}$, we can calculate

$$\begin{aligned}
\mathcal{F}_{\mathcal{T}}\,|a\rangle &= \sum_{\mathcal{X}_\nu\in\mathcal{T}^*} \sqrt{\frac{w_a w_{\mathcal{X}_\nu}}{\varpi_{\mathcal{T}}}}\mathcal{X}_\nu(a)|\mathcal{X}_\nu\rangle \\
&= \sqrt{\frac{w_a}{\varpi_{\mathcal{T}}}} \sum_{\mathcal{X}_\nu\mathcal{K}^\perp\in\mathcal{T}^*/\mathcal{K}^\perp} (\mathcal{X}_\nu\mathcal{K}^\perp)(a) \sum_{\mathcal{X}_\mu\in\mathcal{X}_\nu\mathcal{K}^\perp} \sqrt{w_{\mathcal{X}_\mu}}|\mathcal{X}_\mu\rangle \\
&= \sqrt{\frac{w_a}{\varpi_{\mathcal{T}}}} \sum_{\mathcal{X}_\nu\mathcal{K}^\perp\in\mathcal{T}^*/\mathcal{K}^\perp} (\mathcal{X}_\nu\mathcal{K}^\perp)(a)\sqrt{\frac{w_{\mathcal{X}_\nu\mathcal{K}^\perp}\varpi_{\mathcal{T}}}{\varpi_{\mathcal{K}}}}|\mathcal{X}_\mu\mathcal{K}^\perp\rangle \\
&= \sum_{\mathcal{X}_\nu\mathcal{K}^\perp\in\mathcal{T}^*/\mathcal{K}^\perp} \sqrt{\frac{w_a w_{\mathcal{X}_\mu\mathcal{K}^\perp}}{\varpi_{\mathcal{K}}}}(\mathcal{X}_\nu\mathcal{K}^\perp)(a)|\mathcal{X}_\mu\mathcal{K}^\perp\rangle
\end{aligned}$$

The last line is, by definition, the QFT for $\mathcal{K}$ since the elements of $\mathcal{T}^*/\mathcal{K}^\perp$ are the characters of $\mathcal{K}$. □

This QFT uses the usual basis vectors, $|a\rangle$, for $a \in \mathcal{K}$, but uses the superpositions $|\mathcal{X}_\nu\mathcal{K}^\perp\rangle$ as the basis vectors for $\mathcal{K}^*$. By assumption, we can measure and prepare the basis states for $\mathcal{K}$, but the situation is less clear for $\mathcal{K}^*$. We can measure a state $\mathcal{X}_\nu\mathcal{K}^\perp$

just by measuring in the basis of $\mathcal{T}^*$: whichever $|\mathcal{X}_\mu\rangle$ is found lies in precisely one coset $\mathcal{X}_\nu\mathcal{K}^\perp$, which identifies an element of $\mathcal{K}^*$. On the other hand, it is not obvious how we can prepare states of the form $|\mathcal{X}_\nu\mathcal{K}^\perp\rangle$.

In order to accomplish this, we will make an assumption about the hypergroup $\mathcal{T}$, namely, that it is an **ultragroup**. This means that there exists a tower of subhypergroups $\{a_0\} = \mathcal{T}_0 \subset \cdots \subset \mathcal{T}_k = \mathcal{T}$, where each factor $\mathcal{T}_{i+1}/\mathcal{T}_i$ is a group. For the hypergroup of conjugacy class of G, this holds iff G is nilpotent [54].[8]

Ultragroups are especially useful for quantum computation because many of the hypergroup product operations become unitary. In general, the fact that the product is a probability distribution over elements tells us that the product operation preserves the ℓ_1-norm instead of the ℓ_2-norm. However, in an ultragroup, since multiplication by any $a \in \mathcal{T}_1$ is a permutation, the hypergroup product operation, which we denote $X(a) : \mathbb{C}\mathcal{T} \to \mathbb{C}\mathcal{T}$, is a permutation and, hence, unitary. Furthermore, for any $a \in \mathcal{T}_{i+1}$, the product operations $X(a\mathcal{T}_i)$ in the quotient hypergroup $\mathcal{T}/\mathcal{T}_i$ are likewise a permutation, so, if we identify cosets with superpositions $|b\mathcal{T}_i\rangle$, then the product operation $X(a)$ is a permutation on such coset states.

We will say that an ultragroup $\mathcal{T}$ has **computable scalars** if it meets the following criteria for every subhypergroup tower $\mathcal{K} \subset \mathcal{S} \subset \mathcal{T}$:

1. We can efficiently find generators for the scalars of $\mathcal{S}/\mathcal{K}$, that is, find elements $a_1, \ldots, a_\ell \in \mathcal{S}$ such that $a_1\mathcal{K}, \ldots, a_\ell\mathcal{K}$ generate the scalars of $\mathcal{S}/\mathcal{K}$.

2. We can efficiently implement the group operations $X(a)|b\mathcal{K}\rangle = |ab\mathcal{K}\rangle$ for every a such that $a\mathcal{K}$ is a scalar in $\mathcal{S}/\mathcal{K}$.[9]

For the hypergroup of conjugacy classes of a nilpotent group, both conditions hold. The first holds by corollary 3.2 assuming that we can compute efficiently with conjugacy classes.[10] For the second, we can implement the described operation just by multiplying by an arbitrary element from the conjugacy class (all behave identically

[8] It is interesting to note that nilpotency translates naturally into hypergroups but neither solvability nor super-solvability do the same.

[9] Here, it is assumed that a is an arbitrary element given by its label. Thus, we would be able to implement the conditional multiplication $|a\rangle \otimes |b\mathcal{K}\rangle \mapsto |a\rangle \otimes X(a)|b\mathcal{K}\rangle = |a, ab\mathcal{K}\rangle$.

[10] We need the latter condition to turn elements in G into classes in $\overline{G}$.

on such cosets), which we can, again, implement provided that we can compute efficiently with conjugacy classes.

The next lemma tells us that, if $\mathcal{T}$ is an ultragroup with computable scalars, then we can efficiently prepare superpositions over subhypergroups. Although we will not need it for the algorithm below, this also answers the question about how to prepare states of the form $|\mathcal{X}_\nu \mathcal{K}^\perp\rangle$, at least whenever $\mathcal{X}_\nu$ acts as a group on $\mathcal{T}^*/\mathcal{K}^\perp$, provided that $\mathcal{T}^*$ has computable scalars.

For the lemma, we need one more definition. We will say that a generating set $g_1, \dots, g_\ell$ for a subhypergroup $\mathcal{K} \subset \mathcal{T}$ of an ultragroup is **nice** if, for each $i \in \{1, \dots, k\}$, a prefix of the list of generators, $g_1, \dots, g_{j_i}$ for some j_i, generates $\mathcal{K} \cap \mathcal{T}_i$.[11] Such a generating set always exist for an ultragroup.

Lemma 5.12. *Let $\mathcal{T}$ be an ultragroup with an efficient product. If $\mathcal{K} \subset \mathcal{T}$ is a subhypergroup given by a nice generating set, then we can efficiently prepare $|\mathcal{K}\rangle$.*

Proof. The algorithm of Watrous [62, section 3.2] accomplishes this task if we replace its group multiplication M_{g^i} operation by the operation $X(g^i)$, which can be performed efficiently by assumption. (Note that, even though we have subhypergroups instead of subgroups, all of the operations used to construct these states for groups can be performed in our context as well, by our assumptions, so the same algorithm works here as well.) □

In the previous chapter, we saw that a well-organized set of abelian substructures allows us to perform certain computations more efficiently in groups. This lemma gives an example where the same is true in hypergroups.

With these pieces, we can how show how to solve the HSHP for ultragroups.

Theorem 5.13. *Let $\mathcal{T}$ be an ultra-group with computable scalars. If we can efficiently produce nice generating sets for the kernels of elements in $\mathcal{T}^*$ and there is an efficient quantum circuit implementing the QFT of $\mathcal{T}$, then there is an efficient quantum circuit solving the HSHP.*

[11] We will assume that j_i is known for each i as well.

Proof. We apply the same approach as in the previous two sections, finding a descending sequence of subhypergroups $\mathcal{T} = \mathcal{K}^{(0)} \supset \mathcal{K}^{(1)} \supset \cdots \supset \mathcal{N}$ except that, rather than recursing on each subhypergroup, we apply Lemma 5.11 to compute the QFT for each $\mathcal{K}$ within the Hilbert space $\mathbb{C}\mathcal{T}$.

We can apply each of the steps 1–6 as before except for steps 1–2, which prepare the state $|\mathcal{K}\rangle$. Instead, we prepare this state directly. Since $\mathcal{T}$ has computable scalars, then we can prepare this state directly by Lemma 5.12 provided that our generating set for $\mathcal{K}$ is nice. The latter holds for $\mathcal{K}^{(0)} = \mathcal{T}$ by assumption since this is the kernel of the trivial character. We will see below that this property can be maintained inductively.

Next, we must show that, for each $\mathcal{K}^{(i)}$, we have good probability of measuring a non-trivial character, which will give us a smaller subhypergroup (assuming one exists) for the next iteration. By (5.6) with $\mathcal{T} = \mathcal{K}^{(i)}$, we can see that the probability of measuring $\mathcal{X}_\mu \in \mathcal{K}^{(i)*}$ is now

$$\Pr[\mathcal{X}_\mu \in \mathcal{N}^\perp] = w_{\mathcal{X}_\mu} \sum_{a\mathcal{N} \in \mathcal{K}^{(i)}/\mathcal{N}} \frac{w_{a\mathcal{N}}^2}{\varpi_{\mathcal{K}^{(i)}/\mathcal{N}}^2} |\mathcal{X}_\mu(a)|^2.$$

We can see that this probability is the same for every $\mathcal{X}_\mu$ such that $w_{\mathcal{X}_\mu} = 1$ since every such character has $|\mathcal{X}_\mu(a)|^2 = 1$ for every $a \in \mathcal{T}/\mathcal{N}$ [54]. In fact, this must hold for every $\mathcal{X}_\mu \in (\mathcal{K}^{(i)}/\mathcal{N})_{k-1}^\perp$, where k is the length of the ultra-series for $\mathcal{K}^{(i)}/\mathcal{N}$, as all such characters are scalars. Hence, we conclude that the probability of measuring the trivial character is at most $1/2$ unless $\mathcal{K}^{(i)} = \mathcal{N}$ since $\mathcal{K}^{(i)}/\mathcal{N}$ is also an ultragroup.[12]

If we measure a nontrivial character $\mathcal{X}_\mu$, then we take $\mathcal{K}^{(i+1)} = \mathcal{K}^{(i)} \cap \operatorname{Ker} \mathcal{X}_\mu$. By induction, we have a nice generating set for $\mathcal{K}^{(i)}$, and, by assumption, we can produce a nice generating set for $\operatorname{Ker} \mathcal{X}_\mu$. Thus, we can get a nice generating set for $\mathcal{K}^{(i+1)}$ by the following lemma. □

[12] As we might expect from groups, both subhypergroups and quotients of ultragroups are themselves ultragroups. If $\mathcal{N} \lhd \mathcal{T}$ is a subhypergroup, then it is not hard to check that $\mathcal{N} \cap \mathcal{T}_1 \leq \cdots \leq \mathcal{N} \cap \mathcal{T}_k = \mathcal{N}$ is an ultraseries for $\mathcal{N}$ and $\mathcal{N}\mathcal{T}_1 \leq \cdots \leq \mathcal{N}\mathcal{T}_k$ is an ultraseries for $\mathcal{T}/\mathcal{N}$ once we quotient by $\mathcal{N}$. For the latter, note that, if $n \in \mathcal{N}$ and $t \in \mathcal{T}_{i+1}$, then $(nt)(\overline{nt}) = n\overline{n}t\overline{t} \subset \mathcal{N}\mathcal{T}_i$. For every $x \in nt$, the product $x\overline{x}$ appears in $(nt)(\overline{nt})$ with some positive probability, so this calculation shows that $x\overline{x} \subset \mathcal{N}\mathcal{T}_i$. Hence, we can see that, once we quotient by $\mathcal{N}\mathcal{T}_i$, every element has an inverse, so $\mathcal{N}\mathcal{T}_{i+1}/\mathcal{N}\mathcal{T}_i$ is a group.

Lemma 5.14. *Let $\mathcal{T}$ be an ultragroup with computable scalars. If $\mathcal{A}, \mathcal{B} \subset \mathcal{T}$ be subhypergroups given by nice generating sets, then we can efficiently produce a nice generating set for $\mathcal{A} \cap \mathcal{B}$.*

Proof. We will compute generators of $(\mathcal{A} \cap \mathcal{B}) \cap \mathcal{T}_i$ starting from $i = 1$ and working our way up to $i = k$.

In the case $i = 1$, have generators for $A = \mathcal{A} \cap \mathcal{T}_1$ and $B = \mathcal{B} \cap \mathcal{T}_1$ directly from their nice generating sets. These lie within the abelian group $\mathcal{T}_1$, so this is really an instance of abelian group intersection. The latter problem can be solved by noting that $A \cap B = (A^\perp \cup B^\perp)^\perp$: we can compute the annihilator $H^\perp$ efficiently in any abelian group [46] and the union simply requires concatenating the lists of generators.

For $i > 1$, we have, inductively, a nice generating set for $\mathcal{C} := \mathcal{A} \cap \mathcal{B} \cap \mathcal{T}_{i-1}$. Since $\mathcal{T}$ has computable scalars, we can find a nice generating set for the scalars of $\mathcal{T}/\mathcal{C}$, which we denote $\mathcal{T}'$. Since $\mathcal{C}$ is a subgroup of $\mathcal{A}$ and $\mathcal{B}$ as well, we can also find nice generating sets for the scalars of $\mathcal{A}'$ of $\mathcal{A}/\mathcal{C}$ and $\mathcal{B}'$ of $\mathcal{B}/\mathcal{C}$, both of which are subgroups of the *abelian* group $\mathcal{T}'$. Hence, by the same argument as in the case $i = 1$, we can compute a generating set for $\mathcal{A}' \cap \mathcal{B}'$ in $\mathcal{T}'$.[13]

Now, if $x\mathcal{C}$ is a scalar of $\mathcal{T}/\mathcal{C}$, then we must have $x\overline{x} \subset \mathcal{C}$.[14] In particular, this means that $x\overline{x} \subset \mathcal{T}_{i-1}$, so we must have $x \in \mathcal{T}_i$.[15] Thus, we can conclude that $\mathcal{A}' \cap \mathcal{B}' \subset (\mathcal{A} \cap \mathcal{B} \cap \mathcal{T}_i)/\mathcal{C}$. However, if $x \in \mathcal{A} \cap \mathcal{B} \cap \mathcal{T}_i$, then we have $x\overline{x} \subset \mathcal{A} \cap \mathcal{B}$ and $x\overline{x} \subset \mathcal{T}_{i-1}$, which means that $x\overline{x} \subset \mathcal{C}$. Thus, $\mathcal{A}' \cap \mathcal{B}'$ is precisely $(\mathcal{A} \cap \mathcal{B} \cap \mathcal{T}_i)/\mathcal{C}$.

Our generating set for $\mathcal{A}' \cap \mathcal{B}'$ is a set of elements $g_1, \ldots, g_\ell \in \mathcal{T}_i$ such that $g_1\mathcal{C}, \ldots, g_\ell\mathcal{C}$ generate $(\mathcal{A} \cap \mathcal{B} \cap \mathcal{T}_i)/\mathcal{C}$. Since $\mathcal{C} = \mathcal{A} \cap \mathcal{B} \cap \mathcal{T}_{i-1}$, adding our generators for $\mathcal{C}$ itself gives us a generating set for $\mathcal{A} \cap \mathcal{B} \cap \mathcal{T}_i$. Furthermore, we can preserve that this is a nice generating set by adding the new elements $g_1, \ldots, g_\ell$ after the generators for $\mathcal{C}$. □

Combining the theorem with the facts mentioned above about conjugacy class hypergroups gives us the following corollary.

[13] Note that, even though $\mathcal{T}'$ may not be a well-known group like $\mathcal{T}_1$ in the $i = 1$ case, we can always use abelian decomposition to write it as a product of cyclic groups.

[14] An element x is a scalar precisely when $x\overline{x}$ is the identity. See [54] for a proof.

[15] This is the definition of $\mathcal{T}_i$.

Corollary 5.15. *Let G be a nilpotent group. If we can compute efficiently with conjugacy classes of G and efficiently compute the QFT of $\overline{G}$, then there is an efficient quantum circuit solving the HSHP on $\overline{G}$.*

It is interesting to contrast this result with Theorem 5.9, which also covers these hypergroups, at least in principle. In that algorithm, when we identified a smaller subgroup $K \lhd G$ containing the hidden subgroup, we recursed on $\overline{K}_G$.

While that approach is workable in principle, it requires us to assume that we can implement QFTs for each of these subhypergroups (corresponding to normal subgroups of G), of which there could be exponentially many. The above corollary, in contrast, only requires that have a QFT for $\overline{G}$, making it probably more practical.

Finally, we note that our last theorem also gives us an algorithm solving the HSHP on the character hypergroup of a nilpotent group. Theorem 5.9 does not apply to these hypergroups as they are typically not strongly integral.

Corollary 5.16. *Let G be a nilpotent group such that $\widehat{G}$ has computable scalars. If, for every $\mathcal{X}_\mu \in \widehat{G}$, we can efficiently produce a nice generating set for $\langle \mathcal{X}_\mu \rangle^\perp$, and there is an efficient QFT for $\overline{G}$, then there is an efficient quantum circuit solving the HSHP on $\overline{G}$.*

Proof. The conditions in the statement of the corollary cover all of the conditions of Theorem 5.13 except that we have not stated that $\widehat{G}$ is an ultragroup. However, this follows from the general fact that $\mathcal{T}^*$ is an ultragroup whenever $\mathcal{T}$ is.

To see this, note that $\mathcal{T}_k^\perp \subset \cdots \subset \mathcal{T}_1^\perp \subset \mathcal{T}_0^\perp = \mathcal{T}^*$ is a tower of subhypergroups in $\mathcal{T}^*$. To see that each factor is a group, we can calculate that

$$\begin{aligned} \mathcal{T}_i^\perp/\mathcal{T}_{i+1}^\perp &\cong (\mathcal{T}/\mathcal{T}_i)^*/(\mathcal{T}/\mathcal{T}_{i+1})^* \\ &\cong (\mathcal{T}/\mathcal{T}_i)^*/((\mathcal{T}/\mathcal{T}_i)/(\mathcal{T}_{i+1}/\mathcal{T}_i)^* \\ &\cong (\mathcal{T}/\mathcal{T}_i)^*/(\mathcal{T}_{i+1}/\mathcal{T}_i)^\perp \\ &\cong (\mathcal{T}_{i+1}/\mathcal{T}_i)^* \end{aligned}$$

where we have used standard facts about $\mathcal{N}^\perp$ mentioned earlier and (in the second line) a standard isomorphism theorem about groups [43]. Since we know that $\mathcal{T}_{i+1}/\mathcal{T}_i$ is a group by assumption, we know that $(\mathcal{T}_{i+1}/\mathcal{T}_i)^*$ is a group, which shows that this is an ultraseries for $\mathcal{T}^*$, proving that it is an ultragroup. □

This last corollary shows, under plausible assumptions, that we can efficiently solve the HSHP for hypergroup of characters of a nilpotent group. This is an instance of the HSHP that does not arise from the HKP, providing good evidence that the solvable instances of the HSHP includes problems that are not simply HKPs in disguise.

5.7 Conclusion

In this chapter, we have seen a second way in which the ability of quantum computers to understand abelian structures can be leveraged to solve problems on non-abelian ones. Specifically, we saw that we can solve the HKP on a non-abelian group G by translating G into the *abelian* hypergroup $\overline{G}$ and solving a related instance of the HSHP on $\overline{G}$. While it was previously known that the HKP could be solved efficiently (due to the normality of the hidden subgroup), this gives us an alternative explanation for why the HKP is easy that generalizes beyond groups: it is the fact that the hypergroup is *abelian* that gives us a precise duality between $\overline{G}$ and $\widehat{G}$ and allows us to solve this problem efficiently.

In the previous section, we also saw a second instance where having a nicely structured set of abelian substructures allows problems to be solved efficiently. In the previous chapter, we saw how the abelian factor groups of a nilpotent group allow us to solve the HSP more efficiently. Here, we saw how the abelian factor groups of an ultragroup allow us to solve the HSHP more efficiently. This last algorithm also demonstrated that the HSHP can be solved on instances other than those arising from the HKP.

In the next chapter, we will see a third way in which the ability of quantum computers to understand abelian structures can be leveraged to solve problems on non-abelian ones.

Chapter 6

Applications of Cohomology

In this chapter, we look at our final way of leveraging the ability of quantum computers to solve problems on abelian algebraic structures in order to solve problems on non-abelian ones and, as we will see, even some problems not defined on algebraic structures. This approach, which is, in some ways, the most mathematically sophisticated of those we have seen, uses a technique from homological algebra known as **cohomology**. We will define this in some generality later in the chapter.

Cohomology can be applied to a wide variety of problems. It is used in algebra to study groups, rings, modules, and more [47]. It is used in topology to find invariants of general topological spaces [34] and of specific subtypes such as smooth manifolds [14]. Cohomology has other applications to quantum computing (e.g., [1]), which we will discuss in more detail at the end.

We will see examples from both algebra and topology in this chapter. We start small, with a concrete example, motivated by a practical problem, and develop just enough tools to solve it. Later in the chapter, we will consider cohomology in a broader setting.

In section 6.1, we give efficient quantum algorithms for testing equivalence of group extensions of A by K, when A is given as a black-box group and K is either given as a multiplication table (Theorem 6.1) or is given as a black-box group and is *abelian* (Theorem 6.6). We also show that both problems are classically hard under standard cryptographic assumptions (Theorem 6.7).

In section 6.1.4, we consider the problem of computing the sizes of the component parts of the cohomology group for extensions of A by K. This is of interest because we can reduce the problem of counting the number of inequivalent group extensions of A by K to this problem. We give an efficient quantum algorithm for the case when A is given as a black box and K is given as a multiplication table (Theorem 6.8). We then show that the same problem is classically hard under standard cryptographic assumptions (Theorem 6.9).

Finally, in section 6.2, we consider the problem of computing the sizes of the component parts of the cohomology group for Δ-complexes with coefficients in an abelian group A. Our Δ-complexes are a generalization of simplicial complexes, which are commonly used as triangulations of certain topological spaces. The problem we consider is of interest because we can classically reduce the problem of computing the size of the cohomology groups of simplicial complexes to this problem and the latter is a topological invariant that allows us to distinguish non-homeomorphic spaces from one another whenever the sizes of the corresponding groups are different. We give an efficient quantum algorithm for the case when A is given as a black box and Δ is given explicitly (Theorem 6.15). We then show that the same problem is classically hard under standard cryptographic assumptions (Theorem 6.16).

We conclude in section 6.3.

6.1 Group Cohomology

In this section, our motivation comes from the problem of testing equivalence of group extensions. That is, given two groups G_1 and G_2, both of which are extensions of A by K, we want to know if there is an isomorphism $\gamma : G_1 \to G_2$ fixing A and K. Recall that being an extension of A by K means that $A \lhd G_i$ is an *abelian* normal subgroup and that taking the quotient by A leaves us with K, i.e., $G_i/A \cong K$. We say that γ fixes A and K if γ is the identity on A and on cosets of A (the latter being isomorphic to elements of K). In other words, two extensions are equivalent of A by K if there is an isomorphism between them that respects the structure of such extensions, their relationship to the groups A and K.

6.1.1 General Approach

As described in appendix B, the equivalence classes of extensions are characterized by elements of the second cohomology group. We will start, here, by briefly reviewing these concepts. For simplicity of presentation, we will focus on the case of *central extensions*. However, all of the algorithms considered in this section extend to the more general case (see [66] for details).

The most important object for us is $Z^2(K, A)$, the group of **cocycles**.[1] This consists of all functions $f : K \times K \to A$ that are normalized and satisfy the cocycle condition. The former means that we have $f(x, e) = e$ and $f(e, x) = e$ for all $x \in K$. The latter means, for central extensions, that we have $f(x, y) + f(xy, z) = f(x, yz) + f(y, z)$ for all $x, y, z \in K$.

One fact that will be important to us later on is that $Z^2(K, A)$ is an *abelian* group with the group operation being pointwise addition of functions. We can see immediately that this operation is abelian since the function values lie in A, which is an abelian group. Hence, it remains only to show that this really is a group. The function that is uniformly identity is identity under pointwise addition, and it is not hard to check that this function is indeed a cocycle. Likewise, for any $f \in Z^2(K, A)$, the function $-f$ is the inverse under pointwise addition, and it is not hard to check that this is a also cocycle. Finally, for two functions $f, g \in Z^2(K, A)$, it follows immediately that $f+g$ is normalized, and we can see that $(f+g)(x, y)+(f+g)(xy, z) = f(x, y) + f(xy, z) + g(x, y) + g(xy, z) = f(x, yz) + f(y, z) + g(x, yz) + g(y, z) = (f + g)(x, yz) + (f + g)(y, z)$, which shows that $f + g$ is a cocyle.

The next object we need is $B^2(K, A)$, the group of **coboundaries**. The elements in this group are constructed as follows. Starting with any function $h : K \to A$ that is normalized so that $h(e) = e$, we define $\partial h : K \times K \to A$ by $(\partial h)(x, y) = h(x) + h(y) - h(xy)$ for all $x, y \in K$. We define $B^2(K, A)$ to be the set of all functions

[1] We will leave off the "φ" subscript that appears in the definitions in appendix B since this function is uniformly identity for central extensions.

cocycles	$Z^2(K,A) = \{f \mid f : K \times K \to A$ normalized, cocycle condition$\}$
coboundaries	$B^2(K,A) = \{\partial h \mid h : K \to A$ normalized$\} \subset Z^2(K,A)$
cohomology	$H^2(K,A) = Z^2(K,A)/B^2(K,A)$

Figure 6-1: The main objects in group cohomology.

∂h formed in this manner. It is not hard to check (see appendix B) that ∂h is actually a cocycle, so we have $B^2(K,A) \subset Z^2(K,A)$.

Finally, the **second cohomology group** is $H^2(K,A) := Z^2(K,A)/B^2(K,A)$, the quotient of the two groups above. These definitions are summarized in Figure 6-1.

We can map an extension G to an element of $H^2(K,A)$ as follows. For each element $x \in K$, we choose a representative $\ell(x)$ of the corresponding coset in G. (That is, we have $\pi(\ell(x)) = x$ for each $x \in K$, where $\pi : G \to K$ is the usual projection, taking the quotient by A.) These choices can be made arbitrarily except that we require $\ell(e) = e$. Then we define $f_G : K \times K \to A$ by $f_G(x,y) = \ell(x)\ell(y)\ell(xy)^{-1}$ for each $x, y \in K$. It is not hard to check (see appendix B) that $f_G \in Z^2(K,A)$.

As we can see, this map depends on our choice of representatives in $\ell(x)$. However, a different choice of representatives will give rise to a function f'_G that differs from f_G by some coboundary. In other words, we always have $f_G + B^2(K,A) = f'_G + B^2(K,A)$. Hence, the map taking G to $f_G + B^2(K,A) \in H^2(K,A)$ is well-defined. Furthermore, Theorem B.1 shows that, for extensions G_1 and G_2 of A by K, the elements $f_{G_1} + B^2(K,A)$ and $f_{G_2} + B^2(K,A)$ are the same iff G_1 and G_2 are equivalent.

We can equivalently describe this by saying that the extensions are equivalent iff $f_{G_1} - f_{G_2} \in B^2(K,A)$. Hence, we can solve the equivalence testing problem by reducing it, in this manner, to a membership testing problem: the extensions are equivalent iff the element $f_{G_1} - f_{G_2} \in Z^2(K,A)$ is inside the subgroup $B^2(K,A)$.

As shown by the author in [66], the above approach can be implemented classically in time $\tilde{O}(|K|^6|A|^3)$ if all the groups involved are given as input in the form of multiplication tables. The multiplication table for a group G has size $O(|G|^2)$, so the above algorithm is efficient in terms of the size of its input. However, multiplication tables can only be written down for fairly small groups.

In the next section, we will consider the equivalence testing problem when some of the inputs are black-box groups. The classical algorithm above is no longer efficient in this setting because the input can now have size $O(\log |G|)$, which makes $O(|G|^2)$ exponentially large in the size of the input. Nonetheless, we will see that there are efficient quantum algorithms. This should not surprise us since the groups $Z^2(K, A)$ and $B^2(K, A)$ are abelian, and, as we know, quantum algorithms can solve many problems on abelian groups exponentially faster than is possible with classical algorithms.

6.1.2 Quantum Algorithms for Equivalence Testing

We start with a quantum algorithm for the simpler case when K is still given by a multiplication table, while A, G_1, and G_2 are all given as black boxes. This means that the algorithm is allowed to run in time polynomial in $|K|$ and still be efficient. Hence, this algorithm is useful mainly in cases when $|K|$ is small (e.g., $O(\log |A|)$).

Despite that limitation, this is still a potentially difficult problem. The example of solving the HSP for abelian groups versus the dihedral group (an extension of $\mathbb{Z}_p$ by $\mathbb{Z}_2$) shows that allowing extension by even a group of size $O(1)$ can make computational problems substantially more difficult: the former has efficient quantum algorithms, while only subexponential time algorithms are known for the latter [42]. Indeed, we will see below that equivalence testing is already classically hard, under reasonable assumptions, when $|K| = O(1)$, yet it can be solved efficiently by a quantum algorithm.

Theorem 6.1. *There exists an efficient quantum algorithm for testing the equivalence of G_1 and G_2, two central extensions of A by K, when G_1, G_2, and A are given as black-box groups, the homomorphisms $\pi_i : G_i \to K$ are given as oracles, and the group K is given by a multiplication table (so efficient means* $\text{poly}(|K|, \log |A|)$ *time).*

Proof. As described above, we will solve the problem by reduction to membership testing in a black-box abelian group. We will work with subgroups of the space of all functions $K \times K \to A$.

We first must show that we can efficiently implement these as a black-box group. We represent elements as vectors of $|K|^2$ elements of A. We can efficiently add such vectors and compute inverses pointwise since we are allowed running time polynomial in $|K|$: both operations amount to just $|K|^2$ calls to the oracles for the black-box group A. The identity in this group is simply $|K|^2$ copies of the identity of A, so we can test for the identity as well using $|K|^2$ calls to the oracles for A.

To test membership in $B^2(K, A)$, we must provide a generating set for this group. To do this, we can start with any generating set for the space of all normalized functions $K \to A$ and then apply the map $h \mapsto \partial h$. To see this, note that, if $s_1, \ldots, s_t$ is a set of generators for the functions $K \to A$, then any $h : K \to A$ can be written as $a_1 s_1 + \cdots + a_t s_t$ for some integers $a_i \in \mathbb{Z}$. A short calculation shows that the map $f \mapsto \partial f$ is actually a homomorphism, so we have $\partial h = a_1 \partial s_1 + \cdots + a_t \partial s_t$, which shows that $\partial s_1, \ldots, \partial s_t$ is a generating set for $B^2(K, A)$.

To make a generating set for the space of normalized functions $K \to A$, we use $|K| - 1$ copies of the set of generators for A, where the i-th copy has the generator $a_j \in A$ in the $(i+1)$-st element of the vector[2] and the identity elsewhere. This requires $|K|$ times more generators than for $|A|$, which is only a polynomial increase.

Next, we must show how to generate the element $f_{G_1} - f_{G_2}$. As we saw above, we can compute this pointwise difference efficiently, but first we need to generate the elements f_{G_1} and f_{G_2}. We can do this efficiently applying the definition $f_{G_i}(x, y) = \ell_i(x)\ell_i(y)\ell_i(xy)^{-1}$ pointwise once we have chosen the $\ell_i(x)$'s for each $x \in$K.

To do the latter, we can simply select an element $g \in G_i$ at random and apply the oracle $\pi_i(g)$ to get a representative g for $\pi_i(g) \in K$. We repeat this until we have a representative for each $k \in K \setminus \{e\}$.[3] Now, for a black-box group like G_i, it is not known how to choose elements uniformly at random; however, using random subproducts [3] allows us to select elements such that each has probability $(1 \pm \epsilon)/|G_i|$ in time polynomial in $\log |G_i|$ and $\log(1/\epsilon)$. Taking ϵ to be, say, 0.1, we will get representatives for each $k \in K$ after $O(|K| \log |K|)$ trials with high probability by a

[2] This assumes the first element corresponds to $e \in K$. That element is always $e \in A$ in a normalized function.

[3] Recall that we always choose $f_{G_i}(e) = e$.

standard balls-and-bins argument [21].

Finally, it remains to perform the membership test, which reduces to computing the size of an abelian group: we have $f_{G_1} - f_{G_2} \in B^2(K, A)$ iff the size of the subgroup $B^2(K, A)$ is the same as the size of the group with $f_{G_1} - f_{G_2}$ added as an additional generator. We saw above that we can produce all of the required generators, so it follows from Theorem 3.4 that we can efficiently compute the sizes of these two subgroups and compare them. □

We can see that the above approach depends in a fundamental way on the fact that $|K|$ is small since each group element, in our representation, has size proportional to $|K|$. To get an algorithm when K is a black-box group, we will need to assume additional structure that allows us to represent these functions more compactly.

For the algorithm below, we will allow K to be given as a black-box, but we make the assumption that K is an abelian group. As we saw in chapter 3, we can then efficiently decompose $K \cong \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_m}$, where each r_i is a prime power. Since we can efficiently translate between the original generators and the those of the product decomposition, we will simplify our presentation by assuming that K is simply given in this form.

To see why this helps, consider the problem of choosing representatives $\ell : K \to G$. Since K is now exponentially large, we cannot write down a choice for every $x \in K$. Instead, since $K = \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_m}$, suppose that we merely write down $\ell(e_i)$ for $i \in \{1, \dots, m\}$, where $e_i = (0, \dots, 0, 1, 0, \dots, 0)$ has a 1 in the i-th position only. Since $m = O(\log |K|)$, these choices can be recorded in polynomial space, and, as we will see, they give us enough information to make a choice for every $x \in K$.

Let $x \in K$, and write it as $x = (x_1, \dots, x_m) = x_1 e_1 + \cdots + x_m e_m$ for some x_i's. We will say that ℓ is in **factored form** if we have

$$\ell(x) = \ell(e_1)^{x_1} \cdots \ell(e_m)^{x_m} \tag{6.1}$$

for every $x \in K$. When this holds, we have $\ell(x)A = x_1\ell(e_1)A + \cdots + x_m\ell(e_m)A = (x_1, \dots x_m)A$, which means that $\pi\ell(x) = x$ for all $x \in K$, showing that these are valid choices for representatives.

By choosing our representatives $\ell : K \to G$ in factored form, we can efficiently record ℓ and compute its values. As we saw, we only need to record $\ell(e_1), \ldots, \ell(e_m)$, which requires only poly log $|K|$ space. As we can see from equation (6.1), we can then efficiently compute $\ell(x)$ for any $x \in K$.

We can use this approach to efficiently represent f_{G_1} and f_{G_2}, which is part of what we need to determine whether $f_{G_1} - f_{G_2} \in B^2(K, A)$. Unfortunately, the latter group is still too large to work with, for the same reasons we saw above. However, the following result tells us that, when f_{G_1} and f_{G_2} are both in the factored form described above, then, if they are equivalent, their difference is itself factored. Specifically, their difference then lies in the group $B^2_F(K, A)$, which we define to be the set of all ∂h, where $h : K \to A$ ranges over all normalized functions that are factored as above.

Lemma 6.2. *Let G_1 and G_2 be central extensions of A by K, with factor sets $f_{G_1}, f_{G_2} : K \times K \to A$ sets defined from representatives $\ell_1 : K \to G_1$ and $\ell_2 : K \to G_2$ in factored form. Then, we have $f_{G_1} - f_{G_2} \in B^2(K, A)$ iff $f_{G_1} - f_{G_2} \in B^2_F(K, A)$.*

Proof. The reverse direction is immediate since every $\partial h \in B^2_F(K, A)$ is also in $B^2(K, A)$ by definition.

For the forward direction, suppose that $f_{G_1} - f_{G_2} \in B^2(K, A)$. This means, in particular, that G_2 and G_1 are equivalent, so there exists an isomorphism $\gamma : G_2 \to G_1$ that fixes A and K. In that case, $\tilde{\ell}_2 := \gamma \circ \ell_2 : K \to G_1$ is a set of representatives in G_1 that give rise to same factor set f_{G_2}. To see this, note that, for all $x, y \in K$, we have $\gamma(f_{G_2}(x, y)) = f_{G_2}(x, y)$ since γ fixes A, but since γ is an isomorphism, we also have $\gamma(f_{G_2}(x, y)) = \gamma(\ell_2(x)\ell_2(y)\ell_2(xy)^{-1}) = \gamma(\ell_2(x))\gamma(\ell_2(y))\gamma(\ell_2(xy))^{-1}$. Thus, it is sufficient to show that $f_{G_1} - f_{G_2} \in B^2_F(K, A)$ when both factor sets are defined from representatives chosen in the group G_1.

We can now compute $f_{G_1} - f_{G_2}$ as follows. Switching back to multiplicative notation for G_1, for all $x, y \in K$, we can see that

$$\begin{aligned} f_{G_1}(x, y) f_{G_2}(x, y)^{-1} &= \ell_1(x)\ell_1(y)\ell_1(x + y)^{-1}(\tilde{\ell}_2(x)\tilde{\ell}_2(y)\tilde{\ell}_2(x + y)^{-1})^{-1} \\ &= \ell_1(x)\ell_1(y)\ell_1(x + y)^{-1}\tilde{\ell}_2(x + y)\tilde{\ell}_2(y)^{-1}\tilde{\ell}_2(x)^{-1}. \end{aligned}$$

Now, since $\ell_1(x+y)^{-1}A = (-x-y)A$ and $\tilde{\ell}_2(x+y)A = (x+y)A$, we can see that $\ell_1(x+y)^{-1}\tilde{\ell}_2(x+y)A = A$, which shows that $\ell_1(x+y)^{-1}\tilde{\ell}_2(x+y) \in A$. Since this is a central extension, meaning that A is contained in the center of G_1, we can move this term to the end, giving us

$$f_{G_1}(x,y)f_{G_2}(x,y)^{-1} = \ell_1(x)\ell_1(y)\tilde{\ell}_2(y)^{-1}\tilde{\ell}_2(x)^{-1}\ell_1(x+y)^{-1}\tilde{\ell}_2(x+y).$$

By the same argument, the term $\ell_1(y)\tilde{\ell}_2(y)^{-1}$ is in A, so we can move it further back:

$$f_{G_1}(x,y)f_{G_2}(x,y)^{-1} = \ell_1(x)\tilde{\ell}_2(x)^{-1}\ell_1(y)\tilde{\ell}_2(y)^{-1}\ell_1(x+y)^{-1}\tilde{\ell}_2(x+y).$$

This is almost the same as $(\partial\,\ell_1\tilde{\ell}_2^{-1})(x,y)$:

$$\begin{aligned}(\partial\,\ell_1\tilde{\ell}_2^{-1})(x,y) &= \ell_1(x)\tilde{\ell}_2(x)^{-1}\ell_1(y)\tilde{\ell}_2(y)^{-1}(\ell_1(x+y)\tilde{\ell}_2(x+y)^{-1})^{-1}\\ &= \ell_1(x)\tilde{\ell}_2(x)^{-1}\ell_1(y)\tilde{\ell}_2(y)^{-1}\tilde{\ell}_2(x+y)\ell_1(x+y)^{-1},\end{aligned}$$

differing from $f_{G_1}(x,y)f_{G_2}(x,y)^{-1}$ only in the order of the last two factors. However, we can show that the last two factors commute.

Let $a := \ell_1(x+y)\tilde{\ell}_2(x+y)^{-1}$, which we know is in A. This means that $\ell_1(x+y) = a\tilde{\ell}(x+y)$. Multiplying these factors in the other order gives $\tilde{\ell}_2(x+y)^{-1}\ell_1(x+y) = \tilde{\ell}_2(x+y)^{-1}a\tilde{\ell}_2(x+y) = a\tilde{\ell}_2(x+y)^{-1}\tilde{\ell}_2(x+y) = a$, where we have used the fact that a is in the center of G_1. Hence, the product in either order is a.

Putting this together with the calculation above, we have shown that $f_{G_1}f_{G_2} \equiv \partial\,\ell_1\tilde{\ell}_2^{-1}$. It remains only to show that $\ell_1\tilde{\ell}_2^{-1}$ is factored. To see this, write $x \in K = \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_m}$ as $x = (x_1, \ldots, x_m)$. Then, since ℓ_1 is factored, we have $\ell(x) = \ell(e_1)^{x_1}\cdots\ell(e_m)^{x_m}$, and likewise for $\tilde{\ell}_2$. This means that

$$\begin{aligned}\ell_1(x)\tilde{\ell}_2(x)^{-1} &= \ell_1(e_1)^{x_1}\cdots\ell_1(e_m)^{x_m}(\tilde{\ell}_2(e_1)^{x_1}\cdots\tilde{\ell}_2(e_m)^{x_m})^{-1}\\ &= \ell_1(e_1)^{x_1}\cdots\ell_1(e_m)^{x_m}\tilde{\ell}_2(e_m)^{-x_m}\cdots\tilde{\ell}_2(e_1)^{-x_1}\end{aligned}$$

As before, we can use the fact that $\ell_1(e_m)^{x_m}\tilde{\ell}_2(e_m)^{-x_m}$ is in A to move it to the end.

Repeating the process, we get

$$\ell_1(x)\tilde{\ell}_2(x)^{-1} = \ell_1(e_1)^{x_1}\tilde{\ell}_2(e_1)^{-x_1} \cdots \ell_1(e_m)^{x_m}\tilde{\ell}_2(e_m)^{-x_m},$$

which is precisely the formula for a function in factored form. □

The above lemma tells us that we need only look for $f_{G_1} - f_{G_2}$ in the set $B^2_F(K, A)$. In our earlier algorithm, we tested whether $f_{G_1} - f_{G_2} \in B^2(K, A)$ by comparing the size of $B^2(K, A)$ to that of $\langle f_{G_1} - f_{G_2}\rangle B^2(K, A)$. In principle, we could test $f_{G_1} - f_{G_2} \in B^2_F(K, A)$ by comparing the size of $B^2_F(K, A)$ to that of $\langle f_{G_1} - f_{G_2}\rangle B^2_F(K, A)$.

To that end, from our earlier discussion, we know that $B^2_F(K, A)$ (unlike $B^2(K, A)$) has a small generating set. Hence, we can compute $|B^2_F(K, A)|$. However, it is not obvious that we can compute $|\langle f_{G_1} - f_{G_2}\rangle B^2_F(K, A)|$.

In order to do so, we would need to work within some larger subgroup than $B^2_F(K, A)$ since we do not know, a priori, that $f_{G_1} - f_{G_2}$ is in $B^2_F(K, A)$. It is not clear that there is any subgroup large enough to contain $f_{G_1} - f_{G_2}$ but also small enough to have a small generating set.[4]

Instead of comparing sizes, as before, we will develop a more direct approach for testing whether $f_{G_1} - f_{G_2}$ lies in $B^2_F(K, A)$. The key idea we will need is the following.

Lemma 6.3. *Let G be a central extension of A by K, and let $f : K \times K \to A$ be a factor set defined from representatives $\ell : K \to G$ in factored form. Then, there exist a_i's and $b_{i,j}$'s, all from A, such that, for all $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_m)$ in $K = \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_m}$, we have*

$$f(x, y) = \prod_{i=1}^{m} a_i^{\delta_i} \prod_{j=i+1}^{m} b_{i,j}^{y_i x_j}, \tag{6.2}$$

where $a_i = \ell(e_i)^{r_i}$, $b_{i,j} = [\ell(e_j), \ell(e_i)]$, and δ_i is 1 if $x_i + y_i \geq r_i$ and 0 otherwise.

[4] One natural idea is the space of all factor sets defined by representatives chosen in factored form. However, the product of such factor sets need not be of the same form (as we will see shortly).

Proof. Starting from the definition, we have

$$f(x,y) = \ell(x)\ell(y)\ell(x+y)^{-1},$$

where $x+y = (x_1+y_1, \ldots, x_m+y_m) \in \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_m}$. The i-th component of this is $(x_i+y_i) \bmod r_i$. As an integer, that is $x_i + y_i - \delta_i r_i$, where δ_i is 1 if $x_i + y_i \geq r_i$ and 0 otherwise. Thus, we can see that $\ell(x+y) = \ell(e_1)^{x_1+y_1-\delta_1 r_1} \ldots \ell(e_m)^{x_m+y_m-\delta_m r_m}$ since ℓ is in factored form.

Next, we note that $\ell(e_i)^{\delta_i r_i} \in A$ since $\pi(\ell(e_i)^{\delta_i r_i}) = \delta_i r_i e_i = 0 \in K$, as the i-th component is taken modulo r_i. This means that we can move such factors to the front in our expression above, giving us

$$\begin{aligned}
f(x,y) &= \ell(e_1)^{x_1} \cdots \ell(e_m)^{x_m} \ell(e_1)^{y_1} \cdots \ell(e_m)^{y_m} \times \\
&\qquad (\ell(e_1)^{x_1+y_1-r_1\delta_1} \ldots \ell(e_m)^{x_m+y_m-r_m\delta_m})^{-1} \\
&= \ell(e_1)^{x_1} \cdots \ell(e_m)^{x_m} \ell(e_1)^{y_1} \cdots \ell(e_m)^{y_m} \\
&\qquad \ell(e_m)^{-x_m-y_m+r_m\delta_m} \ldots \ell(e_1)^{-x_1-y_1+r_1\delta_1} \\
&= \ell(e_1)^{r_1\delta_1} \cdots \ell(e_m)^{r_m\delta_m} \times \\
&\qquad \ell(e_1)^{x_1} \cdots \ell(e_m)^{x_m} \ell(e_1)^{y_1} \cdots \ell(e_m)^{y_m} \ell(e_m)^{-x_m-y_m} \cdots \ell(e_1)^{-x_1-y_1} \\
&= a_i^{\delta_1} \cdots a_m^{\delta_m} \times \\
&\qquad \ell(e_1)^{x_1} \cdots \ell(e_m)^{x_m} \ell(e_1)^{y_1} \cdots \ell(e_m)^{y_m} \ell(e_m)^{-x_m-y_m} \cdots \ell(e_1)^{-x_1-y_1}.
\end{aligned}$$

We would like to cancel $\ell(e_m)^{-x_m-y_m}$ with the $\ell(e_m)^{x_m}$ and $\ell(e_m)^{y_m}$ factors. Unfortunately, we cannot do so as $\ell(e_m)^{x_m}$ is not adjacent to the other two factors. However, we can move it back at the expense of introducing a factor of $[\ell(e_m), \ell(e_i)]$ for each factor $\ell(e_i)$ appearing in between $\ell(e_m)^{x_m}$ and $\ell(e_m)^{y_m}$ in the formula. Each such factor is in A since K is abelian,[5] so we can move them to the front as they appear. Thus, we can cancel $\ell(e_m)^{-x_m-y_m}$ with the $\ell(e_m)^{x_m}$ and $\ell(e_m)^{y_m}$ factors at the cost of introducing an overall factor of $[\ell(e_m), \ell(e_1)]^{y_1 x_m} \cdots [\ell(e_m), \ell(e_{m-1})]^{y_{m-1} x_m}$

[5] That K is abelian says that all elements of G commute modulo A, which means that all commutators lie in A.

at the front.

We can repeat this for $\ell(e_{m-1})^{x}_{m-1}, \ldots, \ell(e_1)^{x_1}$, in each case, cancelling the factor of $\ell(e_i)^{-x_i-y_i}$ with the factors of $\ell(e_i)^{x_i}$ and $\ell(e_i)^{y_i}$, but introducing factors of $b_{i,j}^{y_i x_j}$, where $b_{i,j} := [\ell(e_j), \ell(e_i)]$, for each $j \in \{i+1, \ldots, m\}$. This results in the formula found in the statement of the lemma. □

Corollary 6.4. *Let G be a central extension of A by K. If $h : K \to A$ is normalized and in factored form, then all $b_{i,j}$'s in equation (6.2) are zero for the coboundary ∂h.*

Proof. The previous lemma applies to ∂h with $\ell = h$. In this case, the constant $b_{i,j}$ is given by $[h(j), h(i)]$. However, the values of h are in A, which is in the center of G, so these commute with all of G, meaning that $[h(j), h(i)] = e$. □

The above lemma leads to following characterization for $f_{G_1} - f_{G_2} \in B^2_F(K, A)$.

Lemma 6.5. *Let G_1 and G_2 be central extensions of A by K, with factor sets $f_{G_1}, f_{G_2} : K \times K \to A$ sets defined from normalized representatives $\ell_1 : K \to G_1$ and $\ell_2 : K \to G_2$ in factored form. Let us call the a_i's and $b_{i,j}$'s appearing in equation (6.2) the α_i's and $\beta_{i,j}$'s for f_{G_1} and ϵ_i's and $\xi_{i,j}$'s for f_{G_2}. Then, we have $f_{G_1} - f_{G_2} \in B^2_F(K, A)$ iff, for all $1 \leq i < j \leq m$, we have $\beta_{i,j} = \xi_{i,j}$ and $\alpha_i - \epsilon_i$ has an r_i-th root in A.*

Proof. First, suppose that the latter condition holds. In particular, for each $i \in \{1, \ldots, m\}$, let c_i be the r_i-th root of $\alpha_i - \epsilon_i$, so that $\alpha_i = c_i^d \epsilon_i$. Then, we define ℓ_2' by $\ell_2'(e_i) = c_i \ell_2(e_i)$ for each i and extend it to every other $x \in K$ per the factored form.

Now, let f_{G_2}' be the factor set produced from the representatives ℓ_2'. Since we have changed ℓ_2 only by multiplying by factors (the c_i's) in the center of A, the $b_{i,j}$'s from equation (6.2) are unchanged, which means they still agree with those of f_{G_2} and, hence, f_{G_1}. However, we now have $\ell_2'(e_i)^{r_i} = c_i^{r_i} \ell_2(e_i)^{r_i} = c_i^{r_i} \epsilon_i = \alpha_i = \ell_1(e_i)^{r_i}$. Thus, we can see that the a_i's from equation (6.2) are also the same for f_{G_1} and f_{G_2}'. So by Lemma 6.3, we have $f_{G_1} \equiv f_{G_2}'$.

By construction, we know that $f_{G_1} - f_{G_2} = f_{G_2}' - f_{G_2} \in B^2(K, A)$ because the latter two only differ by their choice of representatives. Since both functions are factored, by Lemma 6.2, we must have $f_{G_1} - f_{G_2} \in B^2_F(K, A)$, which is the first condition.

For the other direction, we will prove the contrapositive. First, suppose that $\beta_{i,j} \neq \xi_{i,j}$ for some $1 \leq i < j \leq m$. By Lemma 6.3, we can see that $f_{G_1}(e_j, e_i) - f_{G_2}(e_j, e_i) = \beta_{i,j} - \xi_{i,j} \neq e \in A$. However, the previous corollary tells us that the value of the factor set at (e_j, e_i), which is the $b_{i,j}$ from equation (6.2), must be zero for any coboundary. Thus, we cannot have $f_{G_1} - f_{G_2} \in B^2_F(K, A)$.

Next, suppose that $\alpha_i - \epsilon_i$ does not have an r_i-th root for some $1 \leq i \leq m$. We can see that $f_{G_1}(e_i, (r_i - 1)e_i) - f_{G_2}(e_i, (r_i - 1)e_i) = \ell_1(e_i)^{r_i} - \ell_2(e_i)^{r_i} = \alpha_i - \epsilon_i$. However, if $h : K \to A$ is any coboundary, then Lemma 6.3 tells us that the value of ∂h at $(e_i, (r_i - 1)e_i)$ is the a_i from the equation, which is equal to $h(e_i)^{r_i}$. The latter manifestly has an r_i-th root, namely, $h(e_i) \in A$. Thus, we, once again, cannot have $f_{G_1} - f_{G_2} \in B^2_F(K, A)$, so we have shown that, if the second condition in the statement does not hold, then neither does the first. □

We can now give our second quantum algorithm by showing that we can efficiently check the above characterization.

Theorem 6.6. *There exists an efficient quantum algorithm for testing the equivalence of G_1 and G_2, two central extensions of A by K, both* abelian, *when G_1, G_2, K, and A are given as black-box groups and the homomorphisms $\pi_i : G_i \to K$ are given as oracles (so efficient means* poly$(\log |K|, \log |A|)$ *time).*

Proof. If we can choose representatives for each extension, then we compute the α_i's, $\beta_{i,j}$'s, ϵ_i's, and $\xi_{i,j}$'s from the previous lemma using the formulas given in the proof above. This requires only $O(\log |K|)$ black-box operations for each of these $O(m^2) = O(\log^2 |K|)$ indexes $1 \leq i < j \leq m$, which is efficient. Furthermore, we can directly check $\beta_{i,j} = \xi_{i,j}$ within the same time bound.

To check whether $\alpha_i - \epsilon_i \in A$ has an r_i-th root, we first write A as a direct product $A \cong \mathbb{Z}_{t_1} \times \cdots \times \mathbb{Z}_{t_n}$, which we can do efficiently using the abelian decomposition algorithm. Then, we write $\alpha_i - \epsilon_i$ as $(c_1, \ldots, c_n)$ under this isomorphism. Looking for an r_i-th root, amounts to solving the system of equations $r_i x_j = c_j \pmod{t_j}$ in the variable x_j for $1 \leq j \leq n$. Equivalently, we can solve the equations $r_i x_j + t_j y_j = c_j$ in the variables x_j and y_j. This has a solution iff the greatest common denominator

$\gcd(r_i, t_j)$ divides c_j. As we know, we can compute gcd efficiently, so we have shown that we can efficiently test whether $f_{G_1} - f_{G_2} \in B^2(K, A)$ assuming we can choose representatives for each extension.

Since we will work with representatives chosen in factored form, we only need to choose representatives of each of the generators $e_1, \ldots, e_m$ of $K = \mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_m}$. As discussed earlier, we can choose nearly uniformly random elements from each G_i. Applying the oracle for the homomorphism π_i translates these into nearly uniformly random elements of K, along with a representative of that coset from G_i.[6] Taking $O(\log |K|)$ such samples gives us a generating set for K with high probability.

The fact that these elements generate K means that, for each $i \in \{1, \ldots, m\}$, there is some product of the generators that produces e_i. To find these products, we can simply perform the abelian decomposition algorithm once again using these new generators. The output of that algorithm includes matrices that tell us how to translate between the two types of generators efficiently.[7]

In summary, we have seen that we can efficiently choose representatives for the generators of the space of the cosets of A in each extension, G_1 and G_2, which we extend to representatives of all cosets by using the factored form. That allows us to define factor sets for each extension and efficiently test whether they differ by a coboundary via Lemma 6.3. □

6.1.3 Classical Hardness of Equivalence Testing

In this section, we will prove that the two problems efficiently solved by quantum algorithms in the previous section are classically hard under standard assumptions. In particular, we will show that a classical algorithm for equivalence testing would allow us to break the Goldwasser-Micali cryptosystem, which depends on the assumption that testing quadratic residuosity is classically hard [29].

The inputs to an instance of the latter problem are a large natural number N and

[6] This follows since each coset has the same size.

[7] Technically, the algorithm will produce a set of generators $e'_1, \ldots, e'_m$ that may differ from $e_1, \ldots, e_m$ in some respects. However, we can simply use the former generators for the parts of the algorithm described above. (There is no reason to prefer $e_1, \ldots, e_m$.)

a $y \in \mathbb{Z}_N^\times$. The latter group is the set of elements in $\mathbb{Z}_N$ with multiplicative inverses. (We are also assured that the Jacobi symbol of y is $+1$, but this will not be needed by our reduction.) The goal in this problem is to determine whether y is a quadratic residue, that is, whether y has a square root in $\mathbb{Z}_N^\times$. To be efficient, the algorithm must run in $\operatorname{poly}\log N$ time.

Theorem 6.7. *If there exists an efficient classical algorithm for testing equivalence of G_1 and G_2, two central extensions of A by K, when G_1, G_2, and A are given as black-box groups, the homomorphisms $\pi_i : G_i \to K$ are given as oracles, and K is an* abelian *group given by a multiplication table (so efficient means* $\operatorname{poly}(|K|, \log|A|)$ *time), then there exists an efficient classical algorithm for testing quadratic residuosity.*

In fact, as we will see in the proof below, equivalence testing is already classically hard if we take $K = \mathbb{Z}_2$ and $A = \mathbb{Z}_N^\times$.

Note that this theorem proves that both of the problems solve by our quantum algorithms are classically hard. Since K is given by a multiplication table, this is directly an instance of the problem solved by our first quantum algorithm. However, since K is also abelian and a multiplication table is easily turned into a black-box group,[8] this is reducible to an instance of the second problem as well.

Proof. We will reduce to testing equivalence on two extensions of $\mathbb{Z}_N^\times$ by $\mathbb{Z}_2$. It is well-known that the we can implement the group operations in $\mathbb{Z}_N^\times$ in $\operatorname{poly}\log N$ time[9] [21], so $\mathbb{Z}_N^\times$ is efficiently implementable as a black-box group. Since $\mathbb{Z}_2$ contains only two elements, we can produce a multiplication table for it in constant time.

Recall that an extension of $\mathbb{Z}_N^\times$ by $\mathbb{Z}_2$ consists of pairs (x, a), with $x \in \mathbb{Z}_N^\times$ and $a \in \mathbb{Z}_2$, and has a product of the form $(x,a)(y,b) = (xyf(x,y), a+b)$.[10] Since f can assumed to be normalized, we have $f(0,0) = f(0,1) = f(1,0) = 1$, so f is determined by the single value $f(1,1)$. We need f to be a cocyle, but a short calculation (since

[8] The multiplication table allows us to perform multiplication in constant time. After linear time preprocessing of the multiplication table, identity checking and inverses become constant time operations as well. Finally, we can simply use a list of all the elements as the list of generators.

[9] In particular, we can compute multiplicative inverses using the extended Euclid's algorithm.

[10] We are using multiplicative notation, since that is usual for $\mathbb{Z}_N^\times$, even though it is abelian.

there are only eight possible values for the $x, y, z \in \mathbb{Z}_2$ that appear in the definition) verifies that any value for $f(1,1)$ satisfies the cocycle condition.[11]

We can represent elements of such extensions by a pair of labels, using only $\log N + 1$ bits. We can check identity in $O(\log N)$ time, and we can multiply in $\operatorname{poly}\log N$ time since the latter requires (at most) two multiplications in $\mathbb{Z}_N^\times$. The inverse of (x, a) is $(x^{-1}f(a,-a)^{-1}, -a)$, and this requires only a constant number of multiplications and inverses in $\mathbb{Z}_N^\times$, which, as noted above, can be performed in $\operatorname{poly}\log N$ time.

Finally, the projections $\pi : G_i \mapsto \mathbb{Z}_2$ are just the maps $(x, a) \mapsto a$, which can be implemented in constant time.

Thus, we have seen that we efficiently implement all of the black-box groups and homomorphisms mentioned in the statement of the theorem for any extensions of $\mathbb{Z}_N^\times$ by $\mathbb{Z}_2$. It remains to show that this allows us to test quadratic residuosity.

We choose the G_1 to be the extension with $f_{G_1}(1,1) = 1$ and G_2 to be the extension with $f_{G_2}(1,1) = y$. As we saw above, any value for $f(1,1)$ gives a valid extension. By definition, these two are equivalent iff there is a normalized cochain $\ell : \mathbb{Z}_2 \mapsto \mathbb{Z}_N^\times$ such that $\partial\ell \equiv f_{G_2}f_{G_1}^{-1}$.[12] The latter is just f_{G_2} since f_{G_1} is uniformly identity.

Since these are all factor sets, they are all identity except at $(1,1)$. There, the above equation says $y = f_{G_2}(1,1) = (\partial\ell)(1,1) = \ell(1)\ell(1)\ell(1+1)^{-1} = \ell(1)^2\ell(0)^{-1} = \ell(1)^2$, where we have $\ell(0) = 1$ since ℓ is normalized. Thus, we can see that the two are equivalent iff y has a square root, $\ell(1)$, i.e., iff y is a quadratic residue. □

6.1.4 Counting Equivalence Classes

Before moving on to another type of cohomology, we look at one final problem, which is to count the number of equivalence classes of central extensions of A by K. As we know from above, this is simply $|H^2(K, A)|$. Since this is an abelian group, we could use the abelian decomposition algorithm to compute this. Unfortunately, we do

[11] Alternatively, we can check that both $f(x,y) + f(x+y,z)$ and $f(x,y+z) + f(y,z)$ have value zero unless at least two of $\{x, y, z\}$ have value 1, in which case, it is $f(1,1)$. In particular, the value can never be $2f(1,1)$: e.g., if $f(x,y) = 1$, then $x = y = 1$, which means $x + y = 0$, so we have $f(x+y,z) = 0$.

[12] Once again, we are sticking with the usual multiplicative notation for $\mathbb{Z}_N^\times$, even though it is an abelian group. In our earlier notation, this would have been the more familiar $f_{G_2} - f_{G_1}$.

not have a convenient generating set for this group, so that approach is not available to us. Nonetheless, the following theorem shows that it is still possible to solve this problem efficiently.

Theorem 6.8. *There exists an efficient quantum algorithm for computing the sizes of the groups $|Z^2(K, A)|$ and $|B^2(K, A)|$ when A is given as a black-box group and K is given by a multiplication table (so efficient means* $\text{poly}(|K|, \log |A|)$ *time).*

Recall that we have $H^2(K, A) = Z^2(K, A)/B^2(K, A)$ by definition, which means that we can compute $|H^2(K, A)|$ as $|Z^2(K, A)|/|B^2(K, A)|$. Hence, there is an efficient classical reduction from computing the number of equivalence classes to the problem solved by this algorithm.

Proof. We saw in Theorem 6.1 that we can represent $B^2(K, A)$ as a black-box group. We can then use abelian decomposition to write this as $\mathbb{Z}_{r_1} \times \cdots \times \mathbb{Z}_{r_m}$ for some integers $r_1, \ldots, r_m$, which are also produced by the algorithm. The size of this group is $r_1 \cdots r_m$, which we can compute efficiently to get $|B^2(K, A)|$.

Since we do not have a nice generating set for $Z^2(K, A)$, we instead use the relationship $|Z^2(K, A)| = |C^2(K, A)|/|B^3(K, A)|$. See [66] for definitions of these groups. Here, we simply note that $|C^2(K, A)|$ can be computed by a simple formula in terms of $|A|$, which we can compute as above, and that $|B^3(K, A)|$ can be computed using the same approach as for $B^2(K, A)$, as the two groups are constructed in an analogous manner (as suggested by their names). Hence, we can efficiently compute the sizes of these two groups using techniques discussed already and then divide them to get $|Z^2(K, A)|$. □

As with the equivalence testing problem considered above, we can also show that this problem is classically hard under standard assumptions. In particular, we will show that an efficient classical algorithm would allow us to factor semiprimes[13], which would break the RSA cryptosystem [53].

[13] A semi-prime is an integer that is the product of exactly two primes.

Theorem 6.9. *If there exists an efficient classical algorithm for computing $|Z^2(K,A)|$ or $|B^2(K,A)|$ when A is a given as a black-box group and K as a multiplication table, then there is an efficient classical algorithm for factoring semiprimes.*

In fact, as we will see in the proof below, this problem is already classically hard if we take $K = \mathbb{Z}_2$ and $A = \mathbb{Z}_N^\times$.

Proof. As in our last hardness result, we will consider extensions of $\mathbb{Z}_N^\times$ by $\mathbb{Z}_2$, where $N = pq$ is a semiprime. We start by showing that learning either $|Z^2(K,A)|$ or $|B^2(K,A)|$ would allow us to determine $L := p + q$.

As we saw in Theorem 6.7, any normalized $f : \mathbb{Z}_2 \times \mathbb{Z}_2 \to \mathbb{Z}_N^\times$ is a cocycle. All values of such functions are 1 except for $f(1,1)$, which is arbitrary. Hence, the number of cocycles is $|Z^2(\mathbb{Z}_2, \mathbb{Z}_N^\times)| = |\mathbb{Z}_N^\times|$. The latter number, given by Euler's totient function [24], is $(p-1)(q-1) = N - (p+q) + 1 = N - L + 1$. Hence, if we know $|Z^2(\mathbb{Z}_2, \mathbb{Z}_N^\times)|$, then we can compute L since N is known.

Theorem 6.7 also showed us that the coboundaries are those cocycles such that $f(1,1)$ is a quadratic residue. The number of such residues is $(p-1)(q-1)/4 = (N+1-L)/4$ [61]. Hence, if we know $|B^2(\mathbb{Z}_2, \mathbb{Z}_N^\times)|$, then we can compute L as in the previous case.

Finally, to factor N, we note that p and q are the two solutions of the equation $N = x(L-x) = xL - x^2$ or, equivalently, $x^2 - Lx + N = 0$ in the variable x. By the quadratic formula, these solutions are $(L \pm \sqrt{L^2 - 4N})/2$, which consists of operations that be computed efficiently in terms of $O(\log N)$.[14] □

That completes our analysis of the cohomology of groups. Next, we look at the computational problems related to another form of cohomology.

6.2 Simplicial Cohomology

In this section, we consider the problem of computing the cohomology of a simplicial complex. Before we can define the problem, we need some background on simplicial

[14] We can compute the square root using binary seach, for example, taking advantage of the fact that we know the solution is an integer.

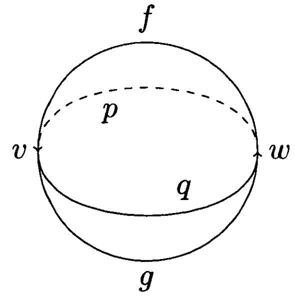

Figure 6-2: The 2-sphere, S^2.

complexes and their homology and cohomology. (For more details, see standard references such as [47, 34].)

6.2.1 Background

Complexes

There are multiple types of simplicial objects for which one can define cohomology. These types vary in their degree of generality. We will work with a definition closer to that of a simplicial set (more general), although we will also discuss the notation of an abstract simplicial complex (less general) as an example below.

We define a Δ-complex, Δ, to be a collection of sets, Δ^n, one for each non-negative $n \in \mathbb{Z}$, and a collection of maps, ∂_n, defined on the corresponding Δ^n. The maps must satisfy some additional conditions that we will explain below. (See [34] for an alternative definition of the same concept.)

The set Δ^n contains the n-dimensional **faces** of the complex. We say that Δ is n-dimensional if n is the largest integer for which Δ^n is non-empty.

The map ∂_n is called a **boundary map**. It takes each face $X \in \Delta^n$ to an integer linear combination of elements from Δ^{n-1}. In symbols, we have $\partial_n : \Delta^n \to \mathbb{Z}\Delta^{n-1}$, and we can extend this function linearly to $\partial_n : \mathbb{Z}\Delta^n \to \mathbb{Z}\Delta^{n-1}$.

Most importantly, the collection of maps must satisfy the condition $\partial_n \circ \partial_{n+1} \equiv 0$ for each non-negative $n \in \mathbb{Z}$. This captures the notion that the boundary of any face should be without boundary.

Example 6.10. *The 2-dimensional sphere is shown in Figure 6.2.1. We have broken S^2 into two hemispheres labeled f and g. The boundary of these hemispheres is the equator, which is broken into segments p and q. Finally, the boundary of those segments are the points v and w.*

We can make this into a Δ-complex as follows. First, we define $\Delta^2 = \{f, g\}$, $\Delta^1 = \{p, q\}$, and $\Delta^0 = \{v, w\}$, so that each face has the dimension that we would expect geometrically.

Next, we define the boundary maps as follows:

$$\partial_2 f = p + q \quad \textit{and} \quad \partial_2 g = -p - q$$
$$\partial_1 p = v - w \quad \textit{and} \quad \partial_1 q = w - v$$
$$\partial_0 v = \partial_0 w = 0$$

The first two equations say that p and q make up the boundary of f and g. We make the boundary of g the negative of that of f to reflect that we are taking these segments in the opposite direction. We do the same for the boundary of p and q: it is $v - w$ for p because this segment starts at w and ends at v, and it is $w - v$ for q because that does the opposite.

While the particular choices of signs may seem somewhat unintuitive, the consequence is that we then have

$$\partial_1 \partial_2 f = \partial_1 (p + q) = v - w + w - v = 0$$

and, likewise, $\partial_1 \partial_2 g = 0$. We also have $\partial_0 \partial_1 \equiv 0$ simply because $\partial_0 \equiv 0$.

Thus, we can see that this gives a valid Δ-complex that captures the geometry of the 2-sphere.

Our next two examples describe abstract simplicial complexes and demonstrate that these are just special cases of Δ-complexes.

Example 6.11. *The standard k-dimensional simplex is the convex hull of the $k+1$ points $v_0, v_1, \ldots, v_k \in \mathbb{R}^k$, where $v_0 = (0, \ldots, 0)$ and $v_i = e_i$ for $i \in \{1, \ldots, k\}$. In gen-*

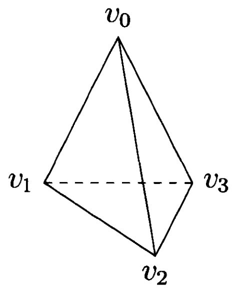

Figure 6-3: A 3-dimensional simplex.

eral, a simplex is an affine transformation of the standard simplex. See Figure 6.2.1 for an example.

We can make this into a Δ complex as follows. For each non-negative $n \in \mathbb{Z}$, we define Δ^n to be any subset of $n+1$ points from the set of v_i's above. Each subset $\{v_{i_0}, \ldots, v_{i_n}\}$ represents the n-dimensional face that is the convex null of those points.

Next, we define the boundary map as follows:

$$\partial_n\{v_{i_0}, \ldots, v_{i_n}\} = \sum_{j=0}^{n}(-1)^j\{v_{i_0}, \ldots, \hat{v_{i_j}}, \ldots, v_{i_n}\},$$

where $\hat{}$ means that this point has been removed from the set. One can check that the faces appearing in this sum are indeed those that are on the boundary. For example, if we consider the face $\{v_1, v_2, v_3\}$ in the simplex of Figure 6.2.1, then we find that the sum is over the three adjacent edges, $\{v_1, v_2\}$, $\{v_1, v_3\}$, and $\{v_2, v_3\}$, which indeed make up the boundary of the bottom of the simplex.

It is not hard to check that this definition also satisfies $\partial_n\partial_{n+1} \equiv 0$. Every face that appears in the result must be of the form $\{v_{i_0}, \ldots, \hat{v_{i_j}}, \ldots, \hat{v_{i_\ell}}, \ldots, v_{i_n}\}$ for some $j \neq \ell$ because both of the boundary operators remove one point. Such a face appears two times from the sum: once when v_{i_j} is removed by ∂_{n+1} and v_{i_ℓ} by ∂_n and once when the opposite occurs. Next, note that, when v_{i_j} is removed first, v_{i_ℓ} moves forward one position, which means that it ends up with the opposite sign when removed by ∂_n and, hence, that these two terms cancel. Since this argument applies to every face that appears in the sum, we can see that the result is identically zero.

Thus, we see that any simplex can be represented as a Δ-complex in this manner.

Example 6.12. *A simplicial complex is the union of a set of nicely-overlapping simplexes. By the latter, we mean that, whenever two simplexes intersect non-trivially, their intersection is a face of both simplexes.*

We can make a simplicial complex into a Δ-complex by taking each Δ^n to be the union of the face sets of the individual simplexes and by taking each ∂_n to act by the same formula as above. By the same argument as above, we have $\partial_n \partial_{n+1} \equiv 0$, so this gives us a valid Δ-complex.

A primary way that simplicial complexes arise is as tools for studying topological spaces. As we just saw, topological spaces that are formed from nicely-intersecting simplexes give rise to simplicial complexes in a straightforward manner. However, for more general topological spaces, it is often possible to find a simplicial complex that is homeomorphic to the space. (Such a simplicial complex is called a **triangulation** of the space.) In that case, we can fully understand the topology of the space by studying the simplicial complex, which is usually much simpler.

The homology and cohomology of a Δ-complex, which we will define next, turn out to be topological invariants. That is, if two topological spaces are homeomorphic[15], then any triangulations of the two spaces have isomorphic homology and cohomology groups. In other words, if two spaces have triangulations with different homology or cohomology, then they cannot be homeomorphic. Thus, the (co)homology of Δ-complexes provides useful information for distinguishing non-homeomorphic spaces.

Homology

We start by defining the n-th **chain group**, $C_n(\Delta) := \mathbb{Z}\Delta^n$, which is just integer linear combinations of elements of Δ^n. An element of $C_n(\Delta)$ is called an n**-chain**. As all of our chain complexes will be finite, we will always have $C_n(\Delta)$ trivial for all n sufficiently large.

[15] Homeomorphism is for topological spaces what isomorphism is for groups: the appropriate notion of what it means for two differently presented objects to really be the same.

As we saw above, each Δ-complex has maps $\partial_n : \Delta^n \to \mathbb{Z}\Delta^{n-1} = C_{n-1}(\Delta)$. By extending ∂_n linearly, we can define it on all integer linear combinations of faces. This gives us a group homomorphism $\partial_n : C_n(\Delta) \to C_{n-1}(\Delta)$. Since the two functions agree where both defined, it should cause no confusion to use the same name for both.

Next, we define the n-th **cycle group** $Z_n(\Delta) := \operatorname{Ker} \partial_n \leq C_n(\Delta)$ and the n-th **boundary group** $B_n(\Delta) := \operatorname{Im} \partial_{n+1} \leq C_n(\Delta)$.

By definition, a boundary is of the form $\partial_{n+1}c$ for some $c \in C_{n+1}(\Delta)$. The defining conditions of a Δ-complex tell us $\partial_n\partial_{n+1}c = 0$ since $\partial_n\partial_{n+1} \equiv 0$, which shows that every boundary is also a cycle (i.e., we have $B_n(\Delta) \leq Z_n(\Delta)$). Hence, we can define the quotient $H_n(\Delta) := Z_n(\Delta)/B_n(\Delta)$, which is the n-th **homology group**. Elements of this group are called homology classes.

Let us now look at a couple of examples, both from [47].

Example 6.13. *We first return to the 2-sphere, whose Δ-complex we saw above.*

An arbitrary 2-chain $c_2 \in C_2(\Delta)$ can be written as $c_2 = af + bg$ for some $a, b \in \mathbb{Z}$. (See Figure 6.2.1 again for the definitions of f and g.) The boundary of this is $\partial_2(af + bg) = a\,\partial_2 f + b\,\partial_2 g = a(p+q) + b(-p-q) = (a-b)(p+q)$. Hence, we can see that $c_2 \in Z_2(\Delta)$ iff $a = b$ iff $c_2 = a(f+g)$. In other words, we have $Z_2(\Delta) = \mathbb{Z}(f+g)$. Meanwhile, $B_2(\Delta)$ is trivial since there are no 3-dimensional faces. This means that we have $H_2(\Delta) = \mathbb{Z}(f+g)$ as well.

Next, consider an arbitrary 1-chain, $c_1 \in C_1(\Delta)$. We can write this as $c_1 = ap + bq$ for some $a, b \in \mathbb{Z}$. The boundary is $\partial_1 c_1 = a\,\partial p + b\,\partial q = a(v-w) + b(w-v) = (a-b)(v-w)$, so as above, we have $c_1 \in Z_1(\Delta)$ iff $a = b$ iff $c_1 = a(p+q)$, which means that $Z_1(\Delta) = \mathbb{Z}(p+q)$. On the other hand, our calculation of $\partial_2 c_2$ above showed that $B_1(\Delta) = \mathbb{Z}(p+q)$. Hence, in this case, we have $H_1(\Delta) = \mathbb{Z}(p+q)/\mathbb{Z}(p+q)$, which is trivial.

Finally, since v and w have no boundary, we have $Z_0(\Delta) = C_0(\Delta) = \mathbb{Z}v + \mathbb{Z}w$. Our calculation of $\partial_1 c_1$ above shows that $B_0(\Delta) = \mathbb{Z}(v-w)$. If we instead write $Z_0(\Delta) = \mathbb{Z}w + \mathbb{Z}(v-w)$, using w an $v - w$ as our basis, then we can see that taking the quotient by $B_0(\Delta)$ wipes out the second part, and we have $H_0(\Delta) = \mathbb{Z}v$.

Example 6.14. *Next, we consider the real projective plane, P^2, whose points correspond to lines through the origin in $\mathbb{R}^3$. Since each such line intersects S^2 in two diametrically opposite points, we can actually form a Δ-complex for P^2 by taking the Δ-complex for S^2 and setting $v = w$, $p = q$, and $f = g$ since, in each case, the two faces are directly opposite one other.*

We now have $C_2(\Delta) = \mathbb{Z}f$ as $g = f$, and we can see that, for any $a \in \mathbb{Z}$, we have $\partial_2(af) = a(p+q) = a(p+p) = 2ap$. This is only zero if $a = 0$, so, in this case, we have $Z_2(\Delta) = C_2(\Delta) = H_2(\Delta)$. (Again, $B_2(\Delta)$ is trivial since there are no 3-chains.)

Next, we have $C_1(\Delta) = \mathbb{Z}p$, and we can see that, for any $a \in \mathbb{Z}$, we have $\partial_1(ap) = a(v-w) = a(v-v) = 0$. Thus, in this case, we have $Z_1(\Delta) = \mathbb{Z}p$. On the other hand, our calculation above shows that $B_1(\Delta) = 2\mathbb{Z}p$: since $\partial_2(af) = 2ap$, we see that the coefficients of 1-boundaries are always multiples of 2. This means that $H_1(\Delta) = \mathbb{Z}p/2\mathbb{Z}p \cong \mathbb{Z}_2p$.

Finally, since $\partial_0 \equiv 0$, we have $Z_0(\Delta) = \mathbb{Z}v$, and, by our previous calculation, we saw that $B_0(\Delta)$ is trivial, so we have $H_0(\Delta) = \mathbb{Z}v$.

Finally, we note that each $C_n(\Delta)$ (and, hence, $Z_n(\Delta)$ and $B_n(\Delta)$) is a finitely generated abelian group. Since quotients of finitely generated groups are also finitely generated, this also means that $H_n(\Delta)$ is a finitely generated abelian group. Thus, it follows from the classification of such groups that $H_n(\Delta)$ is of the form

$$H_n(\Delta) \cong \mathbb{Z}^b \oplus \mathbb{Z}_{p_1^{t_1}} \oplus \cdots \oplus \mathbb{Z}_{p_m^{t_m}},$$

where the p_i's are primes and b and the t_i's are positive integers.

The exponent b in this decomposition is called the n-th **Betti number** of Δ. The remainder is the **torsion coefficient**. This decomposition splits $H_n(\Delta)$ into the part that is purely infinite-cyclic and the part that is purely finite-cyclic. The Betti number uniquely describes the former part, while the latter is uniquely described in the familiar manner for a finite abelian group.

We saw both positive Betti numbers and a non-trivial torsion coefficient in our examples above. In the remainder of this section, we will discuss computing each of these parts of the homology groups.

6.2.2 Quantum Algorithms for Computing Betti Numbers

The algorithm of Lloyd, Garnerone, and Zanardi [45] computes the Betti numbers of a simplicial complex by estimating the ranks of each cycle group, $Z_n(\Delta)$, i.e., the null space of the boundary map ∂_n. It accomplishes the latter by measuring the eigenvalues of the boundary map ∂_n: when applied to a particular mixed state, the probability of measuring each eigenvalue is proportional to the rank of the corresponding eigenspace. By performing many samples, it can estimate the probability of measuring an eigenvalue of zero and, thus, estimate the rank of the null space.

Their result was stated only for simplicial complexes, but their general approach can be applied to an arbitrary Δ-complex provided that the boundary map can be computed efficiently, which it can for all cases of which we are aware.[16]

In order to implement ∂_n as a quantum operation, though, it is necessary to work over $\mathbb{C}$ instead of $\mathbb{Z}$. By the Universal Coefficient Theorem [47, 34], this has the result of eliminating any torsion coefficient in the resulting homology groups. However, working over $\mathbb{C}$ means that knowing the rank of the null spaces is sufficient to also determine the rank via the Rank-Nullity Theorem and the fact that the rank of each chain group is known (or easily estimated). Hence, finding the rank of the null spaces, the $Z_n(\Delta)$'s, also tells us the ranks of the $B_n(\Delta)$'s and $H_n(\Delta)$'s.

While known classical algorithms run in time that grows polynomially in the number of faces, the algorithm of Lloyd et al. runs in time that grows polynomially in the logarithm of the number of faces. Thus, the algorithm provides an exponential speedup, in principle, over the classical approaches.

One caveat, however, is that, if the algorithm requires the boundary map be written down as a matrix at any point, then it must run in time at least linear in the size of the matrix (because it requires linear time just to do that), which eliminates any exponential speedup. Thus, in order to have an exponential speedup, the algorithm needs to work on a matrix that is generated algorithmically.

[16] In particular, the boundary map is row-sparse in all cases of which we are aware.

Such a situation arises in the context of so-called persistent homology. There, each face corresponds to a set of vectors in $\mathbb{R}^m$, for some fixed m, that are all within distance ϵ of one another, where ϵ is a parameter. Lloyd et al. show that it is possible to apply the boundary map efficiently by computing the matrix entries in quantum parallel, eliminating the need to write them down in classical memory.

Thus, the algorithm of Lloyd et al. for computing persistent homology provides a demonstration that quantum algorithms can compute estimates of the sizes of the cycle groups and, hence, the Betti numbers exponentially faster than any known classical algorithm.[17]

Below, we will discuss the hardness of another problem related to computing homology. This differs from the problem considered by Lloyd et al. in two respects. First, it also requires working with the torsion coefficient rather than just the Betti number. Second, we will see that it is classically hard to compute in the low-degree cases (even $n = O(1)$), whereas, for the problem just discussed, the classical algorithms are efficient when n is small.

6.2.3 Simplicial Cohomology

For our hardness result below, we will switch from computing the homology of the Δ-complexes discussed above to computing their cohomology.[18] As we will see, this affects only a couple of small changes, the more important of which is that it allows us to choose coefficients from an arbitrary abelian group A.

In simplicial cohomology, rather than working with the chain group $C_n(\Delta) = \mathbb{Z}\Delta^n$, we work with the **cochain group** defined by $C^n(\Delta, A) := \mathrm{Hom}(C_n(\Delta), A)$. That is, each element of $C^n(\Delta)$ is a homomorphism $C_n(\Delta) \to A$. Now, if $\Delta^n = \{X_1, \dots, X_t\}$, then we can write any $c \in C_n(\Delta^n)$ as $c_{X_1}e_{X_1} + \cdots + c_{X_t}e_{X_t}$, where, for each i, we have $c_{X_i} \in A$ and e_{X_i} is a vector with a 1 in the coordinate for X_i and 0's

[17] It is worth noting that existing classical algorithms achieve exponentially smaller error in their estimates of these values than the quantum algorithm. However, even when allowed exponentially more error, no existing algorithm achieves the running time of the quantum algorithm.

[18] In the language of category theory, cohomology is simply applying the functor $\mathrm{Hom}(-, A)$ to sequence of chain groups before computing homology. However, we will spell this out in detail below.

elsewhere. Then, since any $f \in \mathrm{Hom}(C_n(\Delta), A)$ is a homomorphism, we must have $f(c) = c_{X_1} f(e_{X_1}) + \cdots + c_{X_t} f(e_{X_t})$. Hence, each $f \in \mathrm{Hom}(C_n(\Delta), A)$ is determined by the values $(f(e_{X_1}), \ldots, f(e_{X_t}))$, so elements of $C^n(\Delta, A)$ can also be thought of as vectors with one coordinate for each $X_i \in \Delta^n$, just like elements of $C_n(\Delta)$. The only important difference, here, is that the coordinates are now chosen from A.[19]

A more substantial change appears in the maps between cochain groups. Instead of the boundary map $\partial_n : C_n(\Delta) \to C_{n-1}(\Delta)$, we define a **coboundary** map ∂^n that takes $f \in \mathrm{Hom}(C_n(\Delta), A)$ to the map defined by

$$(\partial^n f)(c) := f(\partial_n c).$$

We can see that this definition requires $c \in C_{n+1}(\Delta)$ in order for the right hand side to make sense. This means that ∂^n takes $C^n(\Delta, A)$ to $C^{n+1}(\Delta, A)$. Thus, while the boundary maps take n-chains to $(n-1)$-chains, the coboundary maps take n-cochains to $(n+1)$-cochains.

With these definitions in place, we can define the other important objects analogously to homology. Specifically, we define the n-th cocycle group to be $Z^n(\Delta, A) := \mathrm{Ker}\, \partial^n$, the n-th coboundary group to be $B^n(\Delta, A) := \mathrm{Im}\, \partial^{n-1}$, and the n-th homology group to be $H^n(\Delta, A) := Z^n(\Delta, A)/B^n(\Delta, A)$. (The last makes sense because we have $\partial^{n+1}\partial^n \equiv 0$,[20] which means that $B^n(\Delta, A) \subset Z^n(\Delta, A)$.)

While there appear to be substantial differences between homology and cohomology (especially between boundary and coboundary maps), these turn out to be superficial. In particular, the Universal Coefficient Theorem for Cohomology [47, 34] tells us that, when the chain and cochain groups are all finitely generated, the homology groups determine the integral cohomology groups and vice versa.[21]

[19] It is also possible to use coefficients in homology by taking a tensor product with A. Indeed, that is implicitly what is done by Lloyd et al. when computing the homology over $\mathbb{C}$ instead of $\mathbb{Z}$.

[20] For any $f \in C^n(\Delta, A)$ and $c \in C_{n+2}(\Delta)$, we have $(\partial^{n+1}\partial^n f)(c) = f(\partial_n \partial_{n+1} c) = f(0) = 0$.

[21] There are some minor algebraic advantages to cohomology. (In particular, it has a well defined product operation, making $H^n(\Delta, A)$ into a ring.) However, that will not matter for us here.

6.2.4 Classical Hardness of Simplicial Cohomology

We now consider computing cohomology with coefficients in an abelian group. Unlike the case considered earlier (computing just the Betti numbers), the resulting cohomology groups now depend on the torsion coefficient as well.

Theorem 6.15. *Let Δ be a Δ-complex with efficiently computable coboundary maps. If there exists a classical algorithm for computing the size of the 2-cocycle group or 2-coboundary group with coefficients in A, an abelian black-box group, that runs in time polynomial in $|\Delta^2|$ and $\log |A|$, then there is an efficient classical algorithm for factoring semiprimes.*

Similar to our earlier cases, it follows from the proof that this problem is already classically hard if we take $A = \mathbb{Z}_N^\times$ and have $|\Delta^2| = O(1)$.

Proof. Let K be a group. We construct a Δ-complex, Δ, with Δ^n containing one element for each n-tuple $(k_1, \ldots, k_n)$ with each $k_i \in K \setminus \{e\}$. Here, we will only need $n \leq 3$, so we may take Δ^n empty for $n > 3$.

Next, we describe the boundary maps. These are given by the formulas:

$$\begin{aligned}
\partial_3[x,y,z] &= [y,z] - [xy,z] + [x,yz] - [x,y] \\
\partial_2[x,y] &= [y] - [xy] + [x] \\
\partial_1[x] &= 0,
\end{aligned}$$

where any of the elements $[\cdot,\cdot]$ is zero if either component is e. It is straightforward to check that we have $\partial_2\partial_3 \equiv 0$, and it is immediate that $\partial_1\partial_2 \equiv 0$, so this meets our definition of a Δ-complex. It is also clear that we can compute these maps efficiently provided that we can efficiently compute products in K.

From the above formulas, we can see that $f : \mathrm{Hom}(C^2(\Delta), A)$ is a 2-cocycle iff

$$\begin{aligned}
0 &= (\partial^3 f)([x,y,z]) = f(\partial_3[x,y,z]) \\
&= f([y,z]) - f([xy,z]) + f([x,yz]) - f([x,y])
\end{aligned}$$

or, equivalently, $f([x,y]) + f([xy,z]) = f([x,yz]) + f([y,z])$, which is the cocycle condition of group cohomology. Specifically, if we identify f with the map $\tilde{f} : K \times K \to A$ defined by $\tilde{f}(x,y) = f([x,y])$, then we can see that the 2-cocycles of this chain complex are in 1-to-1 correspondence with cocycles of group cohomology.[22]

Likewise, we can see the 2-coboundaries are functions of the form $(\partial^2 h)$, with $h \in \mathrm{Hom}(C^1(\Delta), A)$, and defined by

$$(\partial^2 h)([x,y]) = h(\partial_2[x,y]) = h([x]) + h([y]) - h([xy]),$$

once again matching our definition of a coboundary from group cohomology.

Thus, we can see that the sizes of the 2-cocycles and 2-coboundaries of this complex are the same as those of the groups $Z^2(K,A)$ and $B^2(K,A)$, defined earlier. (This also shows where the superscript "2" comes from.) It then follows from Theorem 6.9 that the ability to classically compute the sizes of the groups of 2-cocycles and 2-coboundaries would give an efficient classical algorithm for factoring semiprimes. □

As the proof shows, the group cohomology we saw earlier is actually just the cohomology of a particular Δ-complex related to the group. In fact, the same complex can be presented in a way that makes it essentially a simplicial complex (see [47]).[23]

While the theory of group extensions dates to around the end of the 19th century, the connection to simplicial cohomology was only made a few decades later. However, the latter connection was impactful. MacLane called these the "decisive examples," which demonstrated that the topological idea of homology could be broadly useful [47], leading to the development of homological algebra as a separate subfield.

While we have already seen, in Theorem 6.7, that the specific problem instance constructed in the proof can be solved efficiently by a quantum computer, the following result shows that the same is true of the general problem just considered.

Theorem 6.16. *Let Δ be a Δ-complex with efficiently computable boundary maps. There exists a quantum algorithm for computing the sizes of the cocycle groups or*

[22] These functions are automatically normalized because $[x,y] = 0$ if either $x = 0$ or $y = 0$.

[23] The only difference is that some faces that would be present in the simplicial complex are not present due to the normalization condition.

coboundary groups with coefficients in A, an abelian black-box group, that runs in time polynomial in $|\Delta| := \sum_n |\Delta^n|$ and $\log |A|$.

Proof. The technique used in Theorem 6.7 applies without change to an arbitrary Δ-complex: we apply ∂_{n-1} to a generating set for $C^{n-1}(\Delta)$ in order to produce a generating set for $B^n(\Delta)$ and then compute the size of $B^n(\Delta)$ using the abelian decomposition algorithm.

Our generating set for $C^{n-1}(\Delta)$ will consist of functions $f_{X,a} \in \mathrm{Hom}(C_{n-1}(\Delta), A)$, where $X \in \Delta^{n-1}$ is a face and $a \in A$ is a coefficient, and we take $f_{X,a}$ to be a on X and 0 elsewhere. Letting X range over all faces of Δ^{n-1} and a range over a generating set for A gives a generating set for $\mathrm{Hom}(C_{n-1}(\Delta), A)$ of size $O(|\Delta^{n-1}| \log |A|)$.

This becomes a generating set for $B^n(\Delta)$ by applying the boundary map to each function. In particular, we can efficiently evaluate each function $\partial^{n-1} f_{X,a}$ on the faces in Δ^n to get a representation of each function as a simple vector of values in A, which we can use to present $B^n(\Delta)$ as a black-box group. Then, we use abelian decomposition to find the size $|B^n(\Delta)|$.

We can also compute $|Z^n(\Delta)|$ just as we did in Theorem 6.7, by the formula $|C^n(\Delta)|/|B^{n+1}(\Delta)|$, since $|C^n(\Delta)|$ is simply $|A|^{|\Delta^n|}$ and we can compute $|A|$ by abelian decomposition. □

These last two results, taken together, show that the exponential speedup of quantum algorithms over classical ones for the final problem we considered in our section on group cohomology is actually just a special case of an exponential speedup of quantum algorithms over classical ones for computing the the number of cocycles and coboundaries in the cohomology of Δ-complexes.

6.3 Conclusion

In this chapter, we saw a few example of how the constructions of cohomology lead to problems in which quantum algorithms have exponential speedups compared to classical ones. We saw this, first, in the problem of testing equivalence of group extensions, a variant of the group isomorphism problem appropriate for group extensions, and in a related problem about counting the number of the number of equivalence classes of extensions. We later saw that this second example was actually a special case of a problem related to computing the cohomology of a Δ-complex, which are topological invariants and give us a way of distinguishing non-homeomorphic spaces. The latter problem can also be seen as a complement to the result of Lloyd et al. for computing the Betti numbers of Δ-complexes as it focuses the torsion coefficient rather than the Betti number.

It is perhaps not surprising that the constructions of cohomology give us opportunities to leverage the ability of quantum computers to understand abelian groups into exponential speedups for problems on more complex structures. As we have seen, the central objects of cohomology, chains, cocyles, coboundaries, and cohomology classes, all live in abelian groups that are often very large. If we work with coefficients from $\mathbb{Z}_N^\times$ for example, then they are exponentially large in $\log N$. If we work with coefficients from $\mathbb{Z}$ or $\mathbb{C}$, then they are infinite abelian groups. In both of these cases, we have seen that the ability of quantum computers to work with such groups leads to apparent exponential speedups.

Finally, it is worth mentioning that cohomology has other applications to quantum computation outside of those discussed above. Let us briefly give some intuition for why this occurs.

As Hatcher noted [34], we can think of the elements of $Z_n(\Delta)$ as those n-chains that appear *locally* to be boundaries. To see this, recall that, for any boundary $\partial_{n+1}c \in B_n(\Delta)$, we have $\partial_n(\partial_{n+1}c) \equiv 0$, so $\partial_n d \equiv 0$ is a condition we can check on any $d \in C_n(\Delta)$ that will always hold for boundaries. Furthermore, this condition is local because, as we noted above, $\partial_n d$ is usually very sparse. For example, in an n-dimensional simplicial complex, which can have as many as $2^n - 1$ simplexes, no face has more than $n+1$ faces on its boundary.

The cycle group, $Z_n(\Delta)$, contains those elements that pass this local test. If this test always sufficed, we would have $Z_n(\Delta) = B_n(\Delta)$ and the homology group would be trivial. Whenever we find that $H_n(\Delta) = Z_n(\Delta)/B_n(\Delta)$ is non-trivial, it means that it is possible for n-chains to look locally like boundaries without being boundaries.

Hence, $H_n(\Delta)$ tells us something useful about the difference between the local and global views of these objects.

Differences between the local and global views of objects arise in quantum computing, for example, with maximally entangled states, which appear the same locally as tensor products of maximally mixed states. Abramsky et al. showed [1] that cohomological techniques can be used to prove that certain states are nonlocal based solely on their measurement outcomes. Hence, even in cases where the computational problems are classically solvable, cohomological techniques have proven useful in the study of quantum systems.

Chapter 7

Summary

In this thesis, we have examined ways in which we can leverage the ability of quantum computers to understand finite abelian groups in order to solve problems on more complex structures, namely, non-abelian groups, hypergroups, and Δ-complexes.

We considered three approaches in detail:

1. Taking advantage of a well-organized collection of abelian substructures.

   We applied this approach to certain nilpotent groups[1] in order to solve the hidden subgroup problem and to ultragroups in order solve the hidden subhypergroup problem. For ultragroups, this technique also allowed us to prepare uniform superpositions over subhypergroups, showing that a result of Watrous for solvable groups [62] can be extended to ultragroups.

2. Translating a non-abelian structure into an abelian one.

   We applied this approach to non-abelian groups, translating them into abelian hypergroups, in order to solve the hidden kernel problem and the hidden subgroup problem for class functions.

3. Examine certain abelian groups of maps from our structures into an abelian group (i.e., cohomology).

[1] Specifically, those nilpotent groups whose order is divisible by only large primes.

Objects	**Technique**	**Problem Solved**
nilpotent groups[2]	substructure	Hidden Subgroup Problem (HSP)
ultragroups	substructure	preparing subhypergroup superpositions
ultragroups	substructure	Hidden Subhypergroup Problem (HSHP)
non-abelian groups	translation	Hidden Kernel Problem (HKP)
non-abelian groups	translation	HSP for class functions
non-abelian groups	cohomology	testing equivalence of group extensions
non-abelian groups	cohomology	sizes of cocycle and coboundary groups
Δ-complexes	cohomology	sizes of cocycle and coboundary groups

Table 7.1: A summary of problems solved in this thesis.

> We applied this approach to non-abelian groups in order to test the equivalence of group extensions and to compute the number of equivalence classes. We also applied it to Δ-complexes in order to compute the size of the torsion coefficients of their cohomology groups—topological invariants useful for distinguishing non-homeomorphic spaces.

Each of these problems, save one, can be solved on classical computers. For each, however, we also gave strong evidence of an exponential speedup for quantum algorithms versus classical ones, improving either from exponential to subexponential time or from subexponential to polynomial time.

These results are summarized in Table 7. Along with those of Ivanyos et. al [37] and Watrous [62], they provide a compelling demonstration of the power of quantum computers to solve important problems on more complex finite structures by utilizing their ability to solve problems on abelian groups.

[2] For those nilpotent groups whose order is divisible by only large primes.

Appendix A

Representation Theory

In this chapter, we review all of the representation theory background that we will need in this thesis. Some proofs below are sketches. See the references for complete proofs. Unless stated otherwise, all of this material can be found in [57].

A.1 Irreducible Representations

Initially, it appears as though the problem of identifying all the representations of a finite group G may be intractable since there could be (and, in fact, are) infinitely many non-isomorphic representations. In this section, we will take a giant step in taming this problem by showing that it is sufficient to identify only the smallest representations, as all other representations are built up from these in a simple manner.

A representation ρ is called **irreducible** if there is no nontrivial vector subspace of V_ρ that is stable under multiplication by $\rho(g)$ for every $g \in G$. In symbols, there is no nonzero subspace $W \lneq V_\rho$ such that $\rho(g)W \leq W$ for all $g \in G$.

Theorem A.1 (Maschke). *Every representation is isomorphic to a direct sum of irreducible representations.*

Proof. The function $\langle v \mid w \rangle_{\text{inv}} := \sum_{g \in G} \langle \rho(g)v \mid \rho(g)w \rangle$ defines an inner product on V_ρ. For any $h \in G$, we can see that $\langle \rho(h)v \mid w \rangle_{\text{inv}} = \sum_{g \in G} \langle \rho(g)\rho(h)v \mid \rho(g)w \rangle = \sum_{g \in G} \langle \rho(gh)v \mid \rho(g)w \rangle = \sum_{g'=gh^{-1} \in G} \langle \rho(g')v \mid \rho(g'h^{-1})w \rangle = \langle v \mid \rho(h^{-1})w \rangle_{\text{inv}}$.

Suppose that $W \lneq V_\rho$ is a subspace stable under $\rho(g)$ for all $g \in G$. If $v \in W^\perp$, then, by definition, we have $\langle v \,|\, w \rangle = 0$ for all $w \in W$. For any $h \in G$, this means that, with the new inner product, we have $\langle \rho(g)v \,|\, w \rangle_{\text{inv}} = \langle v \,|\, \rho(g^{-1})w \rangle_{\text{inv}} = 0$ since $\rho(g^{-1})w \in W$. Thus, the space $W^\perp$ is also stable under ρ.

From this, we can write $V = W \oplus W^\perp$, where W and $W^\perp$ are both representations using ρ restricted to that subspace. Since any 1-dimensional space is trivially irreducible (as it has no proper subspaces), the theorem follows by induction on the dimension V_ρ. □

A representation $\rho : G \to \mathrm{GL}(V)$ is said to be **unitary** if each $\rho(g)$ is a unitary transformation, i.e., $\rho(g) \in \mathrm{U}(V) \subset \mathrm{GL}(V)$, for every $g \in G$.

Theorem A.2. *Every representation is isomorphic to a unitary representation.*

Proof. The inner product $\langle \cdot \,|\, \cdot \rangle_{\text{inv}}$ defined in the proof of Maschke's Theorem above is invariant under multiplication by any $\rho(h)$. That is, we have $\langle \rho(h)v \,|\, \rho(h)w \rangle_{\text{inv}} = \langle v \,|\, w \rangle_{\text{inv}}$ for every $h \in G$, which says precisely that each $\rho(h)$ is a unitary transformation (since we already know it is invertible).

If we choose a basis $f_1, \ldots, f_{d_\rho}$ that is orthonormal according to this inner product, then the change of basis $e_i \mapsto f_i$ transforms ρ into a unitary representation. □

The following simple fact about irreducible representations has sophisticated and important consequences.

Lemma A.3 (Schur). *Let ρ and σ be irreducible representations of G, and suppose that $T : V_\rho \to V_\sigma$ is an intertwining map. If ρ and σ are not isomorphic, then $T = 0$, and if $\rho = \sigma$, then T is a scalar multiple of the identity map.*

Proof. The kernel and image of T are stable under ρ and σ, respectively. Since both are irreducible, the kernel must be $\{0\}$ or V_ρ and the image must be $\{0\}$ or V_σ. If the kernel is V_ρ (or equivalently, the image of T is $\{0\}$), then the lemma is proved. Otherwise, the kernel of T is $\{0\}$ and T is an isomorphism $V_\rho \cong V_\sigma$ intertwining ρ and σ, so $\rho \cong \sigma$ and we must be in the case $\rho = \sigma$.

Let λ be an eigenvalue of T (which always exists working over $\mathbb{C}$), and consider the map $S = T - \lambda\,\mathrm{I}$. Since $T\rho(g) = \sigma(g)T$ and $\mathrm{I}\,\rho(g) = \rho(g) = \sigma(g) = \sigma(g)\,\mathrm{I}$, we have $S\rho(g) = \sigma(g)S$ for all $g \in G$. Applying the argument above to S and noting that $\operatorname{Ker} S \neq \{0\}$ by construction, we must have $S = 0$, or equivalently, $T = \lambda\,\mathrm{I}$. □

While the above requires an intertwining map, we can extend this result to arbitrary linear transformations if we symmetrize the map first.

Corollary A.4. *Let ρ and σ be irreducible representations of G, and suppose that $T : V_\rho \to V_\sigma$ is an (arbitrary) linear transformation. Define the tranformation*

$$S := |G|^{-1} \sum_{g\in G} \sigma(g)^{-1} T \rho(g).$$

If ρ and σ are not isomorphic, then $S = 0$, and if $\rho = \sigma$, then $S = \lambda\,\mathrm{I}$ with $\lambda = \operatorname{tr} T/d_\rho$.

Proof. A short calculation verifies that $S : V_\rho \to V_\sigma$ is an intertwining map. The result then follows from Schur's Lemma, with the exact value of λ found by taking the trace on both sides of $S = \lambda\,\mathrm{I}$. □

The fact that T is an arbitrary linear map makes this last corollary much more substantial than it may seem at first. In particular, expanding the right hand side of the definition of S gives a linear function in the $d_\rho \times d_\sigma$ coefficients of T. Whenever the above result says that this is zero, it means this linear function must be zero everywhere, which requires that every coefficient be zero. That leads to the following.

Corollary A.5. *Let ρ and σ be irreducible representations of G given in matrix form. Then for every $i, j \in \mathbb{Z}_{d_\rho}$ and $k, \ell \in \mathbb{Z}_{d_\sigma}$, the sum*

$$|G|^{-1} \sum_{g\in G} [\sigma(g^{-1})]_{k,\ell} [\rho(g)]_{i,j}$$

is zero if ρ and σ are not isomorphic and $\delta_{i,k}\delta_{j,\ell}/d_\rho$ if $\rho = \sigma$.

Proof. This follows from the above discussion. Looking at the (k, j) entry of S gives the above sum for the coefficient on $[T]_{\ell,i}$. □

Finally, we derive one more consequence that will be important in the next section.

Corollary A.6. *Let ρ be an irreducible representation of G, and suppose that $f : G \to \mathbb{C}$ is a class function. Then the linear transformation $T_f \in \mathrm{GL}(V_\rho)$ defined by*

$$T_f := \sum_{g \in G} f(g)\rho(g)$$

is $\lambda \mathrm{I}$ with $\lambda = (|G|/\dim V_\rho)\langle f \,|\, \overline{\chi}_\rho \rangle$, where $\langle \cdot \,|\, \cdot \rangle$ is the same inner product we defined for characters (and which makes sense for any class function).

Proof. We can see that $\rho(x^{-1})T_f\rho(x) = \sum_{g \in G} f(g)\rho(g^x) = \sum_{h=g^x \in G} f(h^{x^{-1}})\rho(h) = \sum_{h \in G} f(h)\rho(g) = T_f$, where $f(h^x) = f(h)$ for any $h, x \in G$ since f is a class function. This shows that $T_f\rho(h) = \rho(h)^{-1}T_f$, which says that T_f is an intertwining map. Thus, $T_f = \lambda \mathrm{I}$ by Schur's Lemma.

To calculate the value of λ, we take the trace of both sides. We have $\mathrm{tr}(\lambda \mathrm{I}) = \lambda \dim V_\rho$ and $\mathrm{tr}\, T_f = \sum_{g \in G} f(g) \,\mathrm{tr}\, \rho(g) = \sum_{g \in G} f(g)\chi_\rho(g) = |G|\langle f \,|\, \overline{\chi}_\rho \rangle$. □

A.2 Character Theory

In section 2.2, we defined the character χ_ρ of a representation ρ and noted that it removed redundancy by identifying isomorphic representations. In this section, we will show that characters equate *only* isomorphic representations and no others (i.e., they are in 1-to-1 correspondence with isomorphism classes of representations), they allow us to easily identify irreducible representations, and they give us a second basis for the space of class functions.

We begin by showing that the irreducible characters are orthonormal under the inner product we defined before.

Theorem A.7. *Let ρ and σ be irreducible representations of G. Then $\langle \chi_\rho \,|\, \chi_\sigma \rangle$ is 1 if $\rho \cong \sigma$ and 0 otherwise.*

Proof. Choosing an arbitrary basis for V_ρ and V_σ, we may assume that they are given in matrix form. By the definition of trace, we have $\chi_\rho(g) = \sum_{i=1}^{d_\rho} [\rho(g)]_{i,i}$ and likewise for χ_σ. Hence, we can see that $\langle \chi_\sigma \,|\, \chi_\rho \rangle = \sum_{i=1}^{d_\rho} \sum_{k=1}^{d_\sigma} |G|^{-1} \sum_{g \in G} [\sigma(g^{-1})]_{k,k} [\rho(g)]_{i,i}$. By Corollary A.5, this sum is zero when ρ and σ are not isomorphic, and when $\rho \cong \sigma$, the sum is $\sum_{i=1}^{d_\rho} \sum_{k=1}^{d_\sigma} \delta_{i,k}/d_\rho = 1$. □

We have already seen that an arbitrary representation ρ is isomorphic to a direct sum of irreducible representations. If $\sigma_1, \ldots, \sigma_k$ are the irreducible representations appearing in this direct sum decomposition of ρ, then we have $\rho \cong m_1\sigma_1 \oplus \cdots \oplus m_k\sigma_k$, for some $m_1, \ldots, m_k \in \mathbb{Z}$ with $m_i \geq 0$. (Here, $m\sigma$ denotes the direct sum of m copies of σ.) This means that $\chi_\rho = \sum_{i=1}^{k} m_i \chi_{\sigma_k}$ (since $\chi_{\sigma_i \oplus \sigma_j} = \chi_{\sigma_i} + \chi_{\sigma_j}$), and, hence, that $\langle \chi_\rho \,|\, \chi_{\sigma_i} \rangle = m_i$ gives the number of copies of σ_i in the direct sum decomposition of ρ.

These observations are also sufficient to prove that character are in 1-to-1 correspondence with isomorphism classes of representations (i.e., that they characterize the isomorphic representations).

Theorem A.8. *Representations have identical characters iff they are isomorphic.*

Proof. We have seen that isomorphic representations have identical characters. Now, suppose that representations ρ and γ have identical characters. Expanding our set of irreducible representations to include those appearing in the direct sum decomposition of γ as well, we can write $\rho \cong m_1\sigma_1 \oplus \cdots \oplus m_k\sigma_k$ and $\gamma \cong n_1\sigma_1 \oplus \cdots \oplus n_k\sigma_k$, where some of the m_i's and n_i's may now be zero. Applying Theorem A.7, we can see that $\langle \chi_\rho \,|\, \chi_{\sigma_i} \rangle = m_i$. Since $\chi_\rho = \chi_\gamma$, we conclude that $m_i = n_i$ for all i. Thus, ρ and γ are both isomorphic to $m_1\sigma_1 \oplus \cdots \oplus m_k\sigma_k$ and, hence, to each other. □

Let ρ and γ be two arbitrary representations, and write $\rho \cong m_1\sigma_1 \oplus \cdots \oplus m_k\sigma_k$ and $\gamma \cong n_1\sigma_1 \oplus \cdots \oplus n_k\sigma_k$ as in the proof above. By the linearity of the inner product and our formula for the inner product of irreducible characters (Theorem A.7), we can see that $\langle \chi_\rho \,|\, \chi_\gamma \rangle = \sum_{i=1}^{k} m_i n_i$.

This calculation shows that $\langle \chi_\rho \,|\, \chi_\gamma \rangle$ is actually the dimension of the space of intertwining maps between V_ρ and V_γ. (In symbols, $\langle \chi_\rho \,|\, \chi_\gamma \rangle = \dim \mathrm{Hom}_G(V_\rho, V_\gamma)$.) Schur's Lemma tells us that the space of maps between the same irreducible representation is actually 1-dimensional, so the space of maps from the direct sum of m

copies into the direct sum of n copies is isomorphic to the space of $m \times n$ matrices, which has size mn. When we have multiple inequivalent irreducible representations, we can make independent choices for each isomorphism class, so the dimensions add. We will use this correspondence again in section A.3.

This formula also allows us to immediately derive the following consequence.

Theorem A.9. *Let ρ be a representation of G. Then $\langle \chi_\rho \,|\, \chi_\rho \rangle$ is a positive integer, and it is 1 iff ρ is irreducible.*

Proof. We have $\langle \chi_\rho \,|\, \chi_\rho \rangle = \sum_{i=1}^{k} m_i^2$, which is always positive and equals 1 iff $k = 1$ and $m_1 = 1$, i.e., if ρ is isomorphic to the irreducible representation σ_1. □

This last result demonstrates that the irreducible characters form an orthonormal basis for the space of all character functions. As we noted in section 2.2, characters are also class functions, so the irreducible representations form a basis for some subspace of the space of class functions. The following theorem shows that this subspace is, in fact, the whole space.

Theorem A.10. *The irreducible characters form an orthonormal basis for the space of class functions on G.*

Proof. Let $f : G \to \mathbb{C}$ be any class function from the space orthogonal to that spanned by the irreducible characters. For any irreducible representation ρ, the map T_f from Corollary A.6 must be zero since $\langle f \,|\, \overline{\chi}_\rho \rangle = 0$ by assumption (as $\overline{\chi}_\rho$ is irreducible when χ_ρ is). Since an arbitrary character is equal to a sum of irreducible characters, we can see that $\langle f \,|\, \chi_\rho \rangle = 0$ and, hence, $T_f \equiv 0$ for any representation ρ.

Next, we apply this to the regular representation. In that case, we have $0 = T_f|e\rangle = \sum_{g \in G} f(g)\,\mathrm{reg}(g)|e\rangle = \sum_{g \in G} f(g)|g\rangle$, which means that $f(g) = 0\ \forall g \in G$. So the only class function orthogonal to the space spanned by irreducible characters is the zero function, which means the irreducible characters span the whole space. □

The technique used above, of proving new results by applying the basic theorems to a carefully chosen (reducible) representation, will reoccur throughout this chapter.

An important consequence of this theorem is the following.

Theorem A.11 (Column Orthogonality). *Let $\chi_{\sigma_1}, \ldots, \chi_{\sigma_k}$ be the complete set of irreducible characters of G. Then, for any $g, h \in G$, the sum $\sum_{i=1}^{k} \overline{\chi}_{\sigma_i}(g)\chi_{\sigma_i}(h)$ is $|G|/|C_g|$ if $h \in C_g$ and zero otherwise.*

Proof. Let f_g be the function that is 1 on C_g and 0 elsewhere. By Theorem A.10, we can write $f_g = \sum_{i=1}^{k} \langle f \,|\, \chi_{\sigma_i} \rangle \chi_{\sigma_i}$. By the definition of f_g, we have $\langle f \,|\, \chi_{\sigma_i} \rangle = \overline{\chi}_{\sigma_i}(g)|C_g|/|G|$. Hence, for any $h \in G$, we have $f_g(h) = (|C_g|/|G|) \sum_{i=1}^{k} \overline{\chi}_{\sigma_i}(g)\chi_{\sigma_i}(h)$. Substituting the definition of $f_g(h)$ on the left gives the desired result. □

Finally, we give one simple application of these results.

Proposition A.12. *Let $\chi_{\sigma_1}, \ldots, \chi_{\sigma_k}$ be the complete set of irreducible characters of G. Then $\sum_{i=1}^{k} d_{\sigma_k}^2 = |G|$.*

Proof. The number of copies of σ_i occurring in the regular representation is $\langle \chi_{\mathrm{reg}} \,|\, \chi_{\sigma_i} \rangle$, and since χ_{reg} is nonzero only at e, we can see that $\langle \chi_{\mathrm{reg}} \,|\, \chi_{\sigma_i} \rangle = |G|^{-1}\chi_{\mathrm{reg}}(e)\chi_{\sigma_i}(e) = |G|^{-1}|G|d_{\sigma_i} = d_{\sigma_i}$. By the direct sum decomposition, the dimension of reg is $\sum_{i=1}^{k} d_{\sigma_i}^2$, but we also know that this dimension is $|G|$ by the definition of reg. □

Now that we have proven the essential facts about representations and characters in general, we will move on to more sophisticated ways of constructing actual representations. Our goal in the final section is to show how to construct all the irreducible representations of supersolvable groups.

A.3 Induced Representations

Let ρ be a representation of a subgroup $H \leq G$. The **induced representation**, denoted $\mathrm{Ind}_H^G \rho$, is a representation of G that merges the representation ρ on the subgroup H with the regular representation of G to get a representation of the whole group.

The cleanest way to construct the induced representation is as follows. We start with the vector space $V_{\mathrm{reg}_G} \otimes V_\rho$. We want to have $g \in G$ act just on the left part by $\mathrm{reg}_G(g) \otimes \mathrm{I}$, and we want to have $h \in H$ act on just the right part by

$\mathrm{I} \otimes \rho(h)$, so we quotient out the vector space of differences between the two, i.e., $\mathrm{reg}_G(h)|v\rangle \otimes |w\rangle - |v\rangle \otimes \rho(h)|w\rangle$ for every $h \in H, |v\rangle \in V_{\mathrm{reg}_G}, |w\rangle \in V_\rho$. Then $\mathrm{Ind}_H^G \rho$ acts on the quotient of these two vector spaces, which we denote $V_{\mathrm{reg}_G} \otimes_H V_\rho$.[1]

We can describe this more concretely after noting that, if $h \in H$ acts only on V_ρ, then only the coset gH affects how g affects V_{reg_G}. Choosing a set of representatives $g_1, \ldots, g_m$ for the cosets in G/H, the underlying vector space of $\mathrm{Ind}_H^G \rho$ will be $\bigoplus_{i=1}^m |g_i\rangle \otimes V_\rho$. Now, when $g \in G$ acts on this space, it will take $|g_i\rangle$ to the unique $|g_j\rangle$ such that $gg_iH = g_jH$. However, the product gg_i (in G) need not be exactly g_j. Instead, we will have $gg_i = g_jh$ for some $h \in H$. In fact, we have $h = g_j^{-1}gg_i$. This part in H will act on V_ρ instead. The operation corresponding to g under $\mathrm{Ind}_G^H \rho$ will take $|g_i\rangle \otimes |w\rangle$ to $|g_j\rangle \otimes \rho(g_j^{-1}gg_i)|w\rangle$ for the one g_j such that $g_j^{-1}gg_i \in H$.[2]

Example A.13. $\mathrm{Ind}_{\{e\}}^G \mathrm{triv}_{\{e\}}$ *is just the regular representation of* G. *More generally, if* $N \lhd G$ *is a normal subgroup, then* $\mathrm{Ind}_N^G \mathrm{triv}_N$ *is the regular representation of* G/N *since* triv *simply throws away the* N *part of each element.*

It is follows quickly from these definitions that induction is transitive.

Proposition A.14. *For any representation* ρ *of a subgroup* $K \leq H \leq G$, *we have* $\mathrm{Ind}_H^G \mathrm{Ind}_K^H \rho = \mathrm{Ind}_K^G \rho$.

Proof. From our first definition, the representation $\mathrm{Ind}_H^G \mathrm{Ind}_K^G$ acts by the regular representation on $V_{\mathrm{reg}_G} \otimes_H V_{\mathrm{reg}_H} \otimes_K V_\rho$. We will focus on the first two parts, $V_{\mathrm{reg}_G} \otimes_H V_{\mathrm{reg}_H}$. The action of any $h \in H$ is, by construction, equivalent on the two parts, we may take the operation of every $g \in G$ to be identity on the V_{reg_H} subspace. Hence, we can get an equivalent representation by dropping this subspace, which leaves us with $V_{\mathrm{reg}_G} \otimes_K V_\rho$ and the action of $g \in G$ on it being $\mathrm{reg}_G(g) \otimes I$, so this is $\mathrm{Ind}_K^G \rho$. □

We can also easily describe the character of $\mathrm{Ind}_H^G \rho$.

Theorem A.15 (Frobenius Formula). *The character of the representation* $\mathrm{Ind}_H^G \rho$ *is given by* $\chi_{\mathrm{Ind}_H^G \rho}(g) = \sum_{x \in G/H} \dot{\chi}_\rho(g^x)$, *where* $\dot{\chi}_\rho(g) = \chi_\rho(g)$ *if* $g \in H$ *and* 0 *otherwise.*

[1] The subscription "H" on the $\otimes$ refers to the fact that we have modded out the differences between the two representations of the subgroup H.

[2] While it may seem that this construction depends on the choice of representatives, our first definition manifestly does not and the second is merely a more concrete description of the former.

Note that it does not matter which representative we choose for each coset in G/H. If we choose xh instead of x, then the sum contains $\chi_\rho(g^{xh}) = \chi_\rho((g^x)^h)$ and, since χ_ρ is a class function on H, this is still $\chi_\rho(g^x)$. (Also, $(g^x)^h \in H$ iff $g^x \in H$.)

Proof. The only parts of the operator for g that appear in the trace are those that show up on the diagonal. This means that we need $gg_i = g_i h$, i.e., $g_i^{-1} g g_i = g^{g_i} \in H$. The sum above includes only those g_i and, for those, the trace of the operation $\rho(g^{g_i})$ on V_ρ is $\chi_\rho(g^{g_i})$ by definition. □

It is also possible to turn a representation of the group G into a representation of the subgroup $H \leq G$ simply by restricting the function to that subgroup. If σ is a representation of G, then we denote its restriction to H by by $\operatorname{Res}_H^G \sigma$.

While seemingly simple, the operation of restriction has a close relationship with induction. The following lemma shows that every intertwining map between a representation ρ of H and $\operatorname{Res}_H^G \sigma$ can be uniquely extended to an intertwining map between $\operatorname{Ind}_H^G \rho$ and σ. That is, these two spaces of intertwining maps are isomorphic.[3]

Lemma A.16. *Let σ be a representation of G and ρ be a representation of $H \leq G$. Then* $\operatorname{Hom}_G(\operatorname{Ind}_H^G \rho, \sigma) \cong \operatorname{Hom}_H(\rho, \operatorname{Res}_H^G \sigma)$.

Proof. For brevity, let $\psi := \operatorname{Ind}_H^G \rho$. Let $S \in \operatorname{Hom}_H(\rho, \operatorname{Res}_H^G \sigma)$ be an intertwining map, and let $T \in \operatorname{Hom}_G(\operatorname{Ind}_H^G \rho, \sigma)$ be another that is equal to S on the subspace $|e\rangle \otimes V_\sigma$. Then, for any $|g_i\rangle \otimes |w\rangle \in V_{\mathrm{reg}_H} \otimes_H V_\rho$, since we have $\psi(g_i^{-1})(|g_i\rangle \otimes |w\rangle) \in |e\rangle \otimes V_\rho$, we know that $T\psi(g_i^{-1})(|g_i\rangle \otimes |w\rangle) = S|w\rangle$. This means that $T(|g_i\rangle \otimes |w\rangle) = T\psi(e)(|g_i\rangle \otimes |w\rangle) = T\psi(g_i)\psi(g_i^{-1})(|g_i\rangle \otimes |w\rangle) = \sigma(g_i)T\psi(g_i^{-1})(|g_i\rangle \otimes |w\rangle) = \sigma(g_i)S|w\rangle$. Thus, if we know that T matches S on the subspace $|e\rangle \otimes V_\rho$, then we know how it acts everywhere, which means that there can be at most one such T for each S.

On the other hand, starting with an $S \in \operatorname{Hom}_H(\rho, \operatorname{Res}_H^G \sigma)$, we can define a T by

[3] In fact, this says a lot more. Though we will not need it here, in the language of category theory, this lemma shows that Ind and Res are so-called adjoint functors.

$T(|g_i\rangle \otimes |w\rangle) := \sigma(g_i)S|w\rangle$. Then, just using the properties of S, we can see that

$$\begin{aligned} T\psi(g)(|g_i\rangle \otimes |w\rangle) &= T(|g_j\rangle \otimes \rho(g_j^{-1}gg_i)|w\rangle) = \sigma(g_j)S\rho(g_j^{-1}gg_i)|w\rangle \\ &= \sigma(g_jg_j^{-1}gg_i)S|w\rangle = \sigma(gg_i)S|w\rangle = \sigma(g)\sigma(g_i)S|w\rangle \\ &= \sigma(g)T(|g_i\rangle \otimes |w\rangle), \end{aligned}$$

which shows that T is an interwining map.

Thus, we have established a 1-to-1 correspondence between the two types of intertwining maps, which completes the proof. □

The following is an immediate consequence.

Theorem A.17 (Frobenius Reciprocity). *Let σ be a representation of G and ρ be a representation of $H \leq G$. Then $\langle \chi_{\mathrm{Ind}_H^G \rho} \,|\, \chi_\sigma \rangle_G = \langle \chi_\rho \,|\, \chi_{\mathrm{Res}_H^G \sigma} \rangle_H$.*

Proof. As noted before, the inner product of two characters is also the dimension of the space of intertwining maps between the corresponding representations, so this follows immediately from the lemma. □

The next result shows how the operations of restriction and induction interact.

Theorem A.18 (Mackey's Decomposition Formula). *Let ρ be a representation of $H \leq G$, and let $K \leq G$ be another subgroup. Then we have*

$$\mathrm{Res}_K^G \mathrm{Ind}_H^G \rho \cong \bigoplus_{x \in K \backslash G/H} \mathrm{Ind}_{H_x}^K \rho^x,$$

where $H_x := K \cap H^{x^{-1}}$ and ρ^x is the representation of H_x defined by $\rho^x(g) := \rho(g^x)$.

Note that the above definition makes sense since ρ^x is defined on $H^{x^{-1}}$: if $g \in H^{x^{-1}}$, then $g = h^{x^{-1}}$ for some $h \in H$, which means that $\rho^x(g) = \rho(g^x) = \rho(h^{x^{-1}x}) = \rho(h)$ and ρ is defined on H.

Proof. In $\mathrm{Ind}_H^G \rho$, we can take $|g_i\rangle \otimes V_\rho$ into $|g_j\rangle \otimes V_\rho$ by applying the operator corresponding to $g_jg_i^{-1}$. Once we restrict to the subgroup K, we can only take the $|g_i\rangle \otimes V_\rho$

into those $|g_j\rangle \otimes V_\rho$ such that $g_j \in Kg_i$. In other words, the spaces corresponding to cosets g_iH and g_jH only interact if we have $g_jH \in Kg_iH$ or, equivalently, $Kg_jH = Kg_iH$. Hence, cosets that do not lie in the same double coset of $K\backslash G/H$ have no interaction, which means that our representation is a direct sum over such double cosets. It remains only to determine the structure within each double coset.

If we pick a $g_\ell \in K\backslash G/H$, then we can see that the subset of $k \in K$ such that the operator for k takes $|g_\ell\rangle \otimes V_\rho$ to itself consists of exactly those such that $k = g_\ell h g_\ell^{-1}$ for some $h \in H$, so this is precisely the set H_{g_ℓ}. For $k_1, k_2 \in K$, we get the same G/H coset in $Kg_\ell H$ iff $k_1 g_\ell H = k_2 g_\ell H$ iff $(k_2 g_\ell)^{-1}(k_1 g_\ell) \in H$ iff $g_\ell^{-1} k_2^{-1} k_1 g_\ell \in H$ iff $k_2^{-1} k_1 \in H^{g_\ell^{-1}}$ iff $k_2 H^{g_\ell^{-1}} = k_1 H^{g_\ell^{-1}}$. This shows that the distinct G/H cosets in $Kg_\ell H$ are in 1-to-1 correspondence with cosets of K/H_{g_ℓ}, which shows that the subset of $\operatorname{Res}^G_K \operatorname{Ind}^G_H \rho$ corresponding to $Kg_\ell H$ is isomorphic to $\bigoplus_{x \in K/H_{g_\ell}} V_{\rho^{g_\ell}}$ as a vector space.

Finally, from the definition, when taking $|k_1 g_\ell\rangle \otimes V_\rho$ to $|k_2 g_\ell\rangle \otimes V_\rho$, the operator for k in $\operatorname{Res}^G_K \operatorname{Ind}^G_H \rho$ takes $|k_1 g_\ell\rangle \otimes |w\rangle$ to $|k_2 g_\ell\rangle \otimes \rho((k_2 g_\ell)^{-1} k (k_1 g_\ell))|w\rangle$. The operation on V_ρ is $\rho((k_2 g_\ell)^{-1} k (k_1 g_\ell)) = \rho(g_\ell^{-1} k_2^{-1} k k_1 g_\ell) = \rho^{g_\ell}(k_2^{-1} k k_1)$, which is, by definition, the operator for k in $\operatorname{Ind}^K_{H_{g_\ell}} \rho^{g_\ell}$ taking $|k_1\rangle \otimes V_{\rho^{g_\ell}}$ to $|k_2\rangle \otimes V_{\rho^{g_\ell}}$. So we can see that this is indeed the representation $\operatorname{Ind}^K_{H_{g_\ell}} \rho^{g_\ell}$ (up to a vector space isomorphism). □

This theorem also gives us a formula for the character of $\operatorname{Res}^G_K \operatorname{Ind}^G_H \rho$ since the character of a direct sum of representations is just the sum of the characters of those representations and we know the characters of each representation appearing in the sum above from Frobenius's formula.

We can also say exactly when an induced representation is irreducible.

Theorem A.19 (Mackey's Irreducibility Criterion). *Let ρ be a representation of $H \leq G$. Then $\operatorname{Ind}^G_H \rho$ is irreducible iff (1) ρ is irreducible and (2), for every $x \in G$, the representations ρ^x and $\operatorname{Res}^H_{H_x} \rho$ contain no common irreducible representations (i.e., zero inner product).*

Proof. Since a representation is irreducible iff its inner product with itself is 1, we merely need to compute this inner product, which we can do easily using Frobenius reciprocity and Mackey's decomposition formula:

$$\begin{aligned}\langle \chi_{\mathrm{Ind}_H^G \rho} \,|\, \chi_{\mathrm{Ind}_H^G \rho} \rangle_G &= \langle \chi_\rho \,|\, \chi_{\mathrm{Res}_H^G \mathrm{Ind}_H^G \rho} \rangle_H \\ &= \sum_{x \in H\backslash G/H} \langle \chi_\rho \,|\, \chi_{\mathrm{Ind}_{H_x}^H \rho^x} \rangle_H \\ &= \sum_{x \in H\backslash G/H} \langle \chi_{\mathrm{Res}_{H_x}^H \rho} \,|\, \chi_{\rho^x} \rangle_{H_x}.\end{aligned}$$

When $x = e$, we have $\rho^x = \rho$ and $\mathrm{Res}_{H_x}^H \rho = \rho$ since $H_x = H \cap H^e = H$. Since $\langle \rho \,|\, \rho \rangle$ is an integer no smaller than 1, we can only have $\langle \mathrm{Ind}_H^G \rho \,|\, \mathrm{Ind}_H^G \rho \rangle = 1$ if $\langle \rho \,|\, \rho \rangle = 1$ and every other term in the sum is zero. These are conditions (1) and (2) above. □

An important special case is when H is a normal subgroup. In that case, we have $H^{x^{-1}} = H$, which means that $H_x = H$ and $\mathrm{Ind}_{H_x}^H \rho^x$ is just ρ^x. Hence, the irreducibility criterion becomes much simpler.

Corollary A.20. *Let ρ be a representation of $N \lhd G$. Then $\mathrm{Ind}_N^G \rho$ is irreducible iff ρ is irreducible and $\rho^x \ncong \rho$ for every $x \neq e$ in G/N.*

We can see that $\langle \chi_\rho \,|\, \chi_\rho \rangle = \langle \chi_{\rho^x} \,|\, \chi_{\rho^x} \rangle$ since the latter just reorders the sum, so ρ is irreducible iff ρ^x is. Hence, if ρ is irreducible, then $\langle \chi_\rho \,|\, \chi_{\rho^x} \rangle$ is either zero or one depending on whether ρ and ρ^x are equivalent or inequivalent. Thus, another way of stating the corollary is that $\mathrm{Ind}_N^G \rho$ is irreducible iff $\{\rho^x \,|\, x \in G/N\}$ is a set of irreducible, inequivalent representations. This will come up again in the next section.

A.4 Clifford Theory

Clifford Theory describes how irreducible representations of G are built out of irreducible representations of a normal subgroup $N \lhd G$. The main result is the following.

Theorem A.21 (Clifford). *Let ρ be an irreducible representation of G, $N \lhd G$ a normal subgroup, and μ an irreducible representation of N occurring in $\mathrm{Res}_N^G \rho$. Define*

*$I_G(\mu)$ to be the set of $x \in G$ such that $\mu^x \cong \mu$ (called the **inertia group** of μ). Then*

$$\operatorname{Res}^G_N \rho \cong \Big(\bigoplus_{x \in G/I_G(\mu)} \mu^g \Big)^{\oplus e},$$

where $e \in \mathbb{Z}$ divides $[I_G(\mu) : N]$. In fact, we have $\rho \cong \operatorname{Ind}^G_{I_G(\mu)} \psi$, where ψ is an irreducible representation of $I_G(\mu)$ such that $\operatorname{Res}^{I_G(\mu)}_N \psi \cong \mu^{\oplus e}$.

Proof. V_ρ is isomorphic to a direct sum of irreducible representations of N, at least one of which is, by assumption, V_μ. Let $\tilde{V}_\mu$ be the subspace of V_ρ corresponding to V_μ under this isomorphism. For $g \in G$, $\rho(g)\tilde{V}_\mu$ is another representation of N, where $n \in N$ acts by $\rho(n)\rho(g)|v\rangle = \rho(ng)|v\rangle = \rho(gn^g)|v\rangle = \rho(g)\mu(n^g)|v\rangle \in \rho(g)\tilde{V}_\mu$, using that $n^g \in N$ since N is normal. I.e., this representation on $\rho(g)\tilde{V}_\mu$ is equivalent to μ^g.

We can see that $\bigcup_{g \in G} \rho(g)\tilde{V}_\mu$ is a subspace of V_ρ. Since it is stable under every $\rho(g)$, it is itself a representation. But since ρ is irreducible, this must be all of V_ρ.

Since $\rho(g)$ is invertible (i.e., it is a vector space isomorphism), each space $\rho(g)\tilde{V}_\mu$ has the same dimension as $\tilde{V}_\mu$. As noted above, each representation μ^g is an irreducible representation of N since μ is. Thus, $\rho(g)$ must permute these spaces: if any $|v\rangle$ is common to $\rho(g_1)\tilde{V}_\mu$ and $\rho(g_2)\tilde{V}_\mu$, then since the span of $\{\mu^{g_i}(n)|v\rangle \mid n \in N\}$ is the whole space for either $i \in \{1,2\}$, they are the same space.

Now, we can define a subgroup $K \leq G$ such that $k \in K$ if $\rho(k)\tilde{V}_\mu = \tilde{V}_\mu$. We certainly have $N \leq K$. We can also define a larger subgroup $I_G(\mu)$ such that $g \in I_G(\mu)$ if $\rho(g)\tilde{V}_\mu \cong V_\mu$ as representations. We then have $N \leq K \leq I_G(\mu) \leq G$.

Let ψ be the representation obtained by restricting ρ to $I_G(\mu)$ and to the subspace $\bigcup_{g \in I_G(\mu)/K} \rho(g)\tilde{V}_\mu$. We can see that $\operatorname{Res}^{I_G(\mu)}_N \psi \cong \mu^{\oplus e}$, where $e = [I_G(\mu) : K]$ divides $[I_G(\mu) : N] = [I_G(\mu) : K][K : N]$.

By the same arguments as above, we can see that the $\rho(g)V_\psi$'s are representations of $I_G(\mu)$ corresponding to ψ^g. But now these are all inequivalent by construction (as they contain μ^g's that are inequivalent). Thus, by the discussion at the end of last section, we must have $\rho = \operatorname{Ind}^G_{I_G(\mu)} \psi$. □

It is possible to say more about the representation ψ that appears in Clifford's

Theorem. It is actually constructed in a simple way from ρ and a representation of $I_G(\mu)/N$. However, the pieces involved in this construction are not ordinary representations but rather so-called projective representations. See [39] for details.

Thankfully, the situation becomes drastically simpler if we require that N has prime index in G, which will be the main case of interest in this thesis, as we will see in the next section. In the remainder of this section, we will determine how ρ is related to other irreducible representations of G whose restriction to N contains μ. The key calculation is the following. This and the corollary that follows are from [54].

Lemma A.22. *Let ρ be a representation of G, $N \lhd G$ a normal subgroup, and μ an irreducible representation occurring in $\operatorname{Res}_N^G \rho$. Then the irreducible components of $\operatorname{Ind}_N^G \mu$ are all of the form $\rho \otimes \varphi$, where φ is an irreducible representation of G/N.*

Proof. By Mackey's decomposition formula, we know that $\operatorname{Res}_N^G \operatorname{Ind}_N^G \mu \cong \sum_{x \in G/N} \mu^x$. If $x \in I_G(\mu)/N$, then we have $\mu^x \cong x$, so we can write this also as $\operatorname{Res}_N^G \operatorname{Ind}_N^G \mu \cong [I_G(\mu) : N] \sum_{x \in G/I_G(\mu)} \mu^x \cong ([I_G(\mu) : N]/e)e \sum_{x \in G/I_G(\mu)} \mu^x \cong ([I_G(\mu) : N)/e) \operatorname{Res}_N^G \rho$, where we have rewritten $\operatorname{Res}_N^G \rho$ using Clifford's Theorem.

Now, let $\psi = \operatorname{Ind}_N^G \mathrm{triv}$, and consider the representation $\rho \otimes \psi$. Since $\psi(n)$ is a $[G : N]$-dimensional identity matrix for $n \in N$, we can see that $\chi_{\rho\otimes\psi}(n) = \chi_\rho(n)\chi_\psi(n) = \chi_\rho(n)[G : N] = [G : N](e/[I_G(\mu) : N])\chi_{\operatorname{Ind}_N^G \mu}(n) = e[G : I_G(\mu)]\chi_{\operatorname{Ind}_N^G \mu}(n)$. For $g \notin N$, we know that both $\chi_\psi(g) = 0$ and $\chi_{\operatorname{Ind}_N^G \mu}(g) = 0$ by the Frobenius formula (or by noting that $\operatorname{Ind}_N^G \gamma$ permutes the γ^x blocks). Thus, we have $\chi_{\rho\otimes\psi} \equiv e[G : I_G(\mu)]\chi_{\operatorname{Ind}_N^G \mu}$ for all $g \in G$, which means that $\rho \otimes \psi \cong e[G : I_G(\mu)] \operatorname{Ind}_N^G \mu$.

As noted in the proof of A.12, every irreducible representation of G/N occurs in the regular representation of G/N, which is ψ above. Thus, $\rho \otimes \psi$ contains every irreducible representation of the form $\rho \otimes \varphi$, where φ is an irreducible representation of G/N, so the same must be true of the $\operatorname{Ind}_N^G \mu$ by our isomorphism above. □

It follows immediately, by Frobenius reciprocity, that the set of irreducible representations σ of G such that $\operatorname{Res}_N^G \sigma$ contains μ (i.e., $\langle \operatorname{Res}_N^G \sigma \mid \mu \rangle_N > 0$) is precisely the set of irreducible representations of G contained in $\operatorname{Ind}_N^G \mu$. In more detail, we have the following.

Corollary A.23. *Let $N \lhd G$ be a normal subgroup, and let ρ and σ be two irreducible representations of G such that $\operatorname{Res}^G_N \rho \cong \operatorname{Res}^G_N \sigma$. Then $\rho \cong \sigma \otimes \varphi$, where φ is a 1-dimensional irreducible representation of G/N.*

Proof. By the previous paragraph, we must have $\rho \cong \sigma \otimes \varphi$ for some irreducible representation of G/N. However, we cannot have $\operatorname{Res}^G_N \rho \cong \operatorname{Res}^G_N \sigma$ unless ρ and σ have the same dimension, which means φ must be 1-dimensional. □

A.5 Representations of Supersolvable Groups

Let G be a supersolvable group. By definition, there is a normal series $\{e\} = N_0 \leq N_1 \leq \cdots \leq N_k = G$ with the property that $[N_{i+1} : N_i]$ is prime for each $0 \leq i \leq k$.

Let ρ be a representation of N_{i+1} and μ a representation of N_i contained in $\operatorname{Res}^{N_{i+1}}_{N_i} \rho$. Since there are no subgroups that lie strictly between N_i and N_{i+1}, we either have $I_G(\mu) = N_i$ or $I_G(\mu) = N_{i+1}$. In the former case, Clifford's Theorem tells us we have $\rho = \operatorname{Ind}^{N_{i+1}}_{N_i} \mu$. In the latter case, it tells us that $\operatorname{Res}^{N_{i+1}}_{N_i} \rho \cong \mu^{\oplus e}$.

Furthermore, whether we have $\mu^g \cong \mu$ or $\mu^g \ncong \mu$ for any $g \in N_{i+1} \setminus N_i$ (and hence all $g \in N_{i+1} \setminus N_i$ since N_{i+1}/N_i is generated by one element) is independent of which ρ we are talking about. Hence, either every representation of N_{i+1} whose restriction to N_i contains μ is induced from μ or none are.

Let σ be another irreducible representation of N_{i+1} whose restriction to N_i contains μ. As just noted, either both ρ and σ are induced or neither is. In the former case, we have $\rho, \sigma \cong \operatorname{Ind}^{N_{i+1}}_{N_i} \mu$, so these are the same representation. In the latter case, by Corollary A.23, we have $\sigma \cong \rho \otimes \varphi$ for some 1-dimensional irreducible representation φ of G/N and, since the characters of these representations are different, they are inequivalent. Hence, in the latter case, the number of inequivalent irreducible representations of N_{i+1} whose restriction to N_i contains μ is exactly $[N_{i+1} : N_i]$.[4]

The above is described in more detail in [7]. Although we will not need it in this thesis, the authors also make the observation that, in the case where $\mu^g \cong \mu$ (so

[4] Since the size of N_{i+1}/N_i is prime, it is abelian, and in that case, the number of inequivalent irreducible representations is equal to the size of the group.

$\rho = \mu \otimes \varphi$), ρ and μ both operate on the same vector space and agree for all $n \in N_i$, so we have $\mu^g(n) = \mu(n^g) = \rho(n^g) = \rho(g)^{-1}\rho(n)\rho(g) = \rho(g)^{-1}\mu(n)\rho(g)$, which shows that $\rho(g) \in \operatorname{Hom}_G(\mu^g, \mu)$ is an intertwining map. Conversely, if $X \in \operatorname{Hom}_G(\mu^g, \mu)$ is any intertwining map scaled so that $X^g = \mathrm{I}$, then defining $\rho(g^i n) := X^i \mu(n)$ gives a representation of $N_{i+1} = \langle g \rangle N_i$. Furthermore, they show that, if we choose appropriate bases for all of these representations, then there is always an intertwining map that is a monomial matrix, and the latter can be constructed by working through the N_i's in a bottom-up fashion.

A.6 Representations of Nilpotent Groups

The results of the previous section show that every representation of a supersolvable group with normal series $N_0 \leq \cdots \leq N_k$ is built up by starting with the trivial representation of $N_0 = \{e\}$ and then, for each $i = 1, \ldots, k$, extending it to a representation of N_{i+1} by either using induction or taking the tensor product with a (1-dimensional) representation of N_{i+1}/N_i. For nilpotent groups, however, we can refine this description even further [43].

Theorem A.24. *Every irreducible representation of a nilpotent group G is of the form $\operatorname{Ind}_H^G \varphi$ for some subgroup $H \leq G$ containing the center, $Z(G)$, and some 1-dimensional representation $\varphi : H \to \mathbb{C}$.*[5]

Proof adapted from [43]. We will prove this by induction on $|G|$. The result is immediate for all abelian groups, including the case $|G| = 1$, since all irreducible representations of such groups are 1-dimensional.[6]

Likewise, we can quickly dispatch the case when ρ has a non-trivial kernel. If $\operatorname{Ker} \rho \neq \{e\}$, then ρ can be viewed as an (irreducible) representation $\bar{\rho}$ of $G/\operatorname{Ker} \rho$, and, since the latter group is strictly smaller, the induction hypothesis tells us that

[5] In particular, this shows that every irreducible representation of a nilpotent group is monomial.

[6] This follows from the fact that we cannot have an isomorphism from an abelian group to a non-abelian one. In particular, the image of the group under the representation must be a commuting set of operators, which can be simultaneously diagonalized. This would contradict irreducibility were the vector space not 1-dimensional.

$\bar{\rho} \cong \operatorname{Ind}_{H/\operatorname{Ker}\rho}^{G/\operatorname{Ker}\rho} \bar{\varphi}$ for some $H \leq G$. If we define $\varphi : G \to \mathbb{C}$ by $\varphi(g) = \bar{\varphi}(g \operatorname{Ker}\rho)$, then this gives us a 1-dimensional (hence, irreducible) representation of K. It is routine (but tedious) to verify that we then have $\rho = \operatorname{Ind}_H^G \varphi$.

Thus, we are left to deal with the case where G is a non-abelian, nilpotent group and the representation ρ has a trivial kernel. In this case, let $x \in Z_2(G) \setminus Z(G)$. Then the subgroup $\langle x \rangle Z(G)$ is normal: since x is in the center of $G/Z(G)$, for every $g \in G$, we know that $(xZ(G))^g = x^g Z(G) = xZ(G)$, which shows that $\langle x \rangle Z(G)$ is normal. We can also see that $(x^i z)(x^j z') = x^{i+j} z z' = (x^j z')(x^i z)$ for any $z, z' \in Z(G)$, so $\langle x \rangle Z(G)$ is also abelian.

We will apply Clifford's Theorem with normal subgroup $N := \langle x \rangle Z(G)$ to any representation ρ of G. Let μ be an irreducible representation of N arising in $\operatorname{Res}_N^G \rho$. Since N is abelian, we know that μ must be 1-dimensional.

We first note that $I_G(\mu)$ cannot be all of G. Suppose not. Then $\operatorname{Res}_{I_G(\mu)}^G \rho \cong \mu^{\oplus e}$. This means that there is a change of basis U such that $U\rho(n)U^{-1} = \mu(n)\,\mathrm{I}$ for any $n \in N$, so we have $\rho(n) = U^{-1}\mu(n)\,\mathrm{I}\,U = \mu(n)\,\mathrm{I}$. Hence, for any $g \in G$, we have $\rho(n^g) = \rho(g)^\dagger \rho(n) \rho(g) = \rho(g)^\dagger \mu(n)\,\mathrm{I}\,\rho(g) = \mu(n)\,\mathrm{I} = \rho(n)$. Now, since $x \in N \setminus Z(G)$, there is some g such that $x^g \neq x$, and since ρ is an isomorphism onto its range, we must have $\rho(x^g) \neq \rho(x)$. However, $x^g \in N$ as well since N is normal, so this contradicts what we just showed.

Clifford's Theorem tells us that $\rho \cong \operatorname{Ind}_{I_G(\mu)}^G \psi$, where ψ is an irreducible representation of $I_G(\mu)$. By the previous paragraph, $|I_G(\mu)|$ is strictly smaller than $|G|$, so by induction, we have $\psi = \operatorname{Ind}_H^{I_G(\mu)} \varphi$ for some normal subgroup $H \leq I_G(\mu)$. This means that $\rho \cong \operatorname{Ind}_{I_G(\mu)}^G \operatorname{Ind}_H^{I_G(\mu)} \varphi = \operatorname{Ind}_H^G \varphi$, with the last equality by Proposition A.14. □

Appendix B

Group Cohomology

In this chapter, we review the background on group cohomology that we will need in this thesis. (For a more detailed treatment, see [55].) Our eventual goal is a proof that group extensions are in 1-to-1 correspondence with elements of the corresponding second cohomology group.

Recall that G is said to be an extension of A by K if $A \lhd G$ is an *abelian* normal subgroup and $G/A \cong K$. If G' is another extension of A by K, then we say that G and G' are **equivalent** if there exists an isomorphism $\gamma : G \to G'$ that leaves A and K fixed. In detail, the latter means that (1) γ is the identity on H and (2) γ leaves each $g \in G$ in the same coset of A. We can write condition (2) in symbols as $\pi'(\gamma(g)) = \pi(g)$, for all $g \in G$, where $\pi : G \to K$ and $\pi' : G' \to K$ are the usual projections that take the quotient by A. This definition captures what it means to be "the same" extension of A by K.

Notation Since A is an abelian group, we will use additive notation—writing $a + b$ instead of ab—for calculations that take place entirely within A.

B.1 From Extensions to Factor Sets

Let G be an extension of A by K. In this section, we will describe two pieces of information derived from G. In the next section, we will see that these two pieces of information, in fact, completely characterize the extension up to equivalence.

First, we can define a homomorphism $\varphi_G : K \to \operatorname{Aut} A$ into the automorphism group of A by taking the coset gA to the inner automorphism $a \mapsto a^g$. Note that this is well defined (i.e., independent of our choice of representative for the coset gA) since, if $g' = gb$, for some $b \in A$, then $a^{g'} = a^{gb} = (gb)^{-1}a(gb) = b^{-1}g^{-1}agb = b^{-1}a^g b = a^g$ since $b, a^g \in A$ and A is abelian. We will denote the value of this homomorphism applied to x by φ_x rather than $\varphi(x)$ since this is itself a function and this notation lets us write $\varphi_x(a)$ rather than $\varphi(x)(a)$.

Second, we can define a map $f_G : K \times K \to A$ from pairs of elements from K into A. To construct this map, we first choose a set of representatives of each the cosets. For each $x \in K$, let $\ell(x)$ be an element of the coset mapping to x, i.e., satisfying $\pi(\ell(x)) = x$. The values of $\ell : K \to G$ can be arbitrary except that we fix e to be the representative for the coset $A = eA$. We then define $f_G(x, y) := \ell(x)\ell(y)\ell(xy)^{-1}$.

The map f_G is called a **factor set**. It has two important properties:

1. Normalization: For any $x \in K$, we have $f(x, e) = f(e, x) = e$.

   This follows from our choice of $\ell(e) = e$ since we have $f(x, e) = \ell(x)\ell(e)\ell(x)^{-1} = \ell(x)\ell(x)^{-1} = e$ and similarly for $f(e, x)$.

2. Cocycle Condition: For any $x, y, z \in K$, we have

$$f(x, y) + f(xy, z) = \varphi_x(f(y, z)) + f(x, yz).$$

   To see this, we first note that, by definition, we have $\ell(x)\ell(y) = f(x, y)\ell(xy)$. Then, we have $(\ell(x)\ell(y))\ell(z) = f(x, y)\ell(xy)\ell(z) = f(x, y)f(xy, z)\ell(xyz)$ and, on the other hand, $\ell(x)(\ell(y)\ell(z)) = \ell(x)f(y, z)\ell(yz) = f(y, z)^{\ell(x)}\ell(x)\ell(yz) = \varphi_x(f(y, z))f(x, yz)\ell(xyz)$. Cancelling $\ell(xyz)$ gives the equation above.

In chapter 2, we saw two examples of group extensions: semidirect products and central extensions. Let us look at the above maps for these two types of extensions.

In a central extension, by definition, we have $A \leq Z(G)$. This means that $a^g = g^{-1}ag = a$ for every $a \in A$ and $g \in G$. As a result, the homomorphism φ is the identity since φ_x is the identity for every $x \in K$. We will see below that the converse is also true: if φ is the identity, then G is a central extension.

In a semidirect product, the quotient $G/A \cong K$ is isomorphic to a subgroup of G itself. In detail, if $G = \tilde{A} \rtimes_\varphi K$, with product $(a,x)(b,y) = (a\varphi_x(b), xy)$, then K is isomorphic to the set of elements $\{(e,x) \mid x \in K\}$. This forms a subgroup since it contains (e,e) and $(e,x)(e,y) = (\varphi_x(e), xy) = (e, xy)$.

Hence, in this semidirect product, we can choose $\ell(x) = (e,x)$ as the representative of the coset $(e,x)A$, where A is the subgroup $\{(a,e) \mid a \in \tilde{A}\}$. In that case, we have $f_G(x,y) = (e,x)(e,y)(e,xy)^{-1} = (e,xy)(e,xy)^{-1} = (e,e)$, which shows that f_G is uniformly identity. We will see below that the converse is also true: if f_G is uniformly identity, then G is a semidirect product.

More generally, this last example shows that f_G is a measure of how close ℓ is to being a homomorphism. If ℓ is fully a homomorphism, then f_G is uniformly identity. If there is even a subgroup of K on which ℓ is a homomorphism, then f_G is uniformly identity when restricted to that subgroup.

Putting these examples together, we can see that φ describes the way in which G looks like a semidirect product and f_G describes the way in which G looks like a central extension. In a pure semidirect product or central extension, we need only one of these pieces of information as the other can be uniformly identity. In the general case, we think of any extension as a combination of a semidirect product described by φ and a central extension described by f_G.

B.2 Low-Degree Group Cohomology

In this section, we will show that the two pieces of information described above, after one minor change, characterize the extension up to equivalence. This minor change is required to fix the fact that the function f_G described above is not well-defined. Specifically, as described, the function f_G depends on our choice of representatives for each of the cosets of A in G.

If we should choose a different set of representatives $\ell' : K \to G$, then, for any

$x \in K$, by definition, we still have $\ell(x)A = \ell'(x)A$, so $\ell(x)$ and $\ell'(x)$ simply differ by some element in A. Let us define $h(x) := \ell'(x)\ell(x)^{-1} \in A$ to be this element. Then, the two functions are related by $\ell'(x) = h(x)\ell(x)$. (Also, note that we always have $h(e) = e$ since we required $\ell(e) = \ell'(e) = e$.)

Now, let f'_G be the factor set defined using ℓ' instead of ℓ. Then, we can see that

$$\begin{aligned} f'_G(x,y) &= \ell'(x)\ell'(y)\ell'(xy) \\ &= h(x)\ell(x)h(y)\ell(y)\ell(xy)^{-1}h(xy)^{-1} \\ &= h(x)h(y)^{\ell(x)}\ell(x)\ell(y)\ell(xy)^{-1}h(xy)^{-1} \\ &= h(x)h(y)^{\ell(x)}f_G(x,y)h(xy)^{-1} \\ &= h(x)h(y)^{\ell(x)}h(xy)^{-1}f_G(x,y) \\ &= h(x)\varphi_x(h(y))h(xy)^{-1}f_G(x,y), \end{aligned}$$

where, on the second-to-last line, we have used the fact that all the values are in A and, hence, commute with one another. If we define $(\partial h)(x,y) := h(x) + \varphi_x(h(y)) - h(xy)$,[1] then the above calculation shows that $f'_G \equiv f_G + \partial h$.

Thus, we can see that different choices of representatives for the cosets of A in G lead to factor sets that differ just by ∂h for some $h : K \to A$. Furthermore, reversing the above calculation shows that, if we have $f'_G \equiv f_G + \partial h$, for some $h : K \to A$, then these factor sets just differ by choices of coset representatives.

To formalize this, let $Z^2_\varphi(K,A)$ denote the set of all **cocycles**, that is, all functions $f : K \times K \to A$ that satisfy the normalization and cocycle conditions. As we saw above, all factor sets are cocycles. Next, we let $B^2_\varphi(K,A)$ denote the set of all **coboundaries**, which are simply functions of the form ∂h for some $h : K \to A$, where we require $h(e) = e$ to preserve normalization. Finally, we define the **second cohomology group** to be $H^2_\varphi(K,A) := Z^2_\varphi(K,A)/B^2_\varphi(K,A)$.[2]

In other words, elements of the second cohomology group are cosets of the form $f + B^2_\varphi(K,A)$ in the space of cocycles. From our discussion above, we can see that

[1] All of these values are in A, so we can use additive notation here.

[2] The reason for the superscript "2" relates to the fact that these are all functions of two variables.

the coset $f_G + B^2_\varphi(K, A)$ is well-defined—independent of our choice of representatives. Furthermore, this coset tells us a great deal about the extension G. The following theorem shows that the coset characterizes the extension G up to equivalence.

Theorem B.1. *Let G and G' be two extensions of A by K with conjugation of A by K given by $\varphi : K \to \operatorname{Aut} A$. Then G and G' are equivalent extensions iff $f_G + B^2_\varphi(K, A) \equiv f'_G + B^2_\varphi(K, A)$.*

Proof. Suppose that G and G' are equivalent. Then, there is an isomorphism $\gamma : G \to G'$ that fixes A and K. If f_G is a factor set for G using representatives $\ell : K \to G$, then we can define a set of representatives for G' by $\ell' := \gamma \circ \ell$. This means that f'_G is given by $f'_G(x, y) = \ell'(x)\ell'(y)\ell'(xy)^{-1} = \gamma(\ell(x))\gamma(\ell(y))\gamma(\ell(xy))^{-1} = \gamma(\ell(x)\ell(y)\ell(xy)^{-1})$, for each $x, y \in K$, using the fact that γ is a homomorphism. This shows that $f'_G(x, y) = \gamma(f_G(x, y))$, but since $f_G(x, y) \in A$ and γ fixes A, we in fact have $f'_G \equiv f_G$, which certainly implies that $f_G + B^2_\varphi(K, A) \equiv f'_G + B^2_\varphi(K, A)$.

Now, suppose that $f_G + B^2_\varphi(K, A) \equiv f'_G + B^2_\varphi(K, A)$. This means that f_G and f'_G differ by ∂h for some $h : K \to A$. Let ℓ and ℓ' be the coset representatives giving the factor sets f_G and f'_G, respectively.

We can define a map $\gamma : G \to G'$ as follows. First, note that any element $g \in G$ lies in a unique coset of A, which is $\ell(x)A$ for some $x \in K$, so we can write $g = a\ell(x)$ for a unique $a \in A$.[3] We then define $\gamma(g) = \gamma(a\ell(x)) = ah(x)\ell'(x)$.

We can see that $\gamma(a) = \gamma(a\ell(e)) = ah(e)\ell'(e) = a$, so γ fixes A. Also, for any $x \in K$, we have $\pi'(\gamma(a\ell(x))) = \pi'(ah(x)\ell'(x)) = x$ since $\pi'(\ell'(x)) = x$, by definition of ℓ', and since multiplication by $ah(x) \in A$ does not change the coset of A. This calculation shows that γ fixes K as well.

It remains only to prove that γ is a homomorphism.[4] Since γ fixes A we have $\gamma(e) = e$. Now, let $g = a\ell(x)$ and $h = b\ell(y)$ be arbitrary elements of G. Then, we can see that $\gamma(gh) = \gamma(a\ell(x)b\ell(y)) = \gamma(a\varphi_x(b)\ell(x)\ell(y)) = \gamma(a\varphi_x(b)f_G(x, y)\ell(xy)) = a\varphi_x(b)f_G(x, y)h(xy)\ell'(xy)$ since $a\varphi_x(b)f_G(x, y) \in A$. On the other hand, we have $\gamma(g)\gamma(h) = \gamma(a\ell(x))\gamma(b\ell(y)) = ah(x)\ell'(x)bh(y)\ell'(y) = ah(x)\varphi_x(bh(y))\ell'(x)\ell'(y) =$

[3] Recall that A is normal, so left and right cosets are identical.

[4] Note that the fact that γ fixes A and K implies that it a bijection.

$ah(x)\varphi_x(bh(y))f'_G(x,y)\ell'(xy)$. Cancelling the factors of $\ell'(xy)$, everything that remains is in A, which is abelian, and we can see that $\gamma(gh) = \gamma(g)\gamma(h)$ provided that $f_G(x,y) + h(xy) = f'_G(x,y) + h(x) + \varphi_x(h(y))$ or, equivalently, $f_G(x,y) = f'_G(x,y) + (\partial h)(x,y)$, which holds by assumption. □

B.3 From Factor Sets to Extensions

Above, we showed that we derive a factor set f_G and homomorphism φ from an extension, both of which provide useful information about the extension G. In this section, we will show that it is also possible to start with a factor set f and homomorphism φ and (re-)derive the group. Hence, these two pieces of information actually tell us everything there is to know about the extension, up to equivalence.

Let $\varphi : K \to \operatorname{Aut} A$ be a homomorphism and $f : K \times K \to A$ be a cocycle. We will construct a group, denoted G^f_φ, as follows. The set of elements will be simply $A \times K$, and the product is given by

$$(a,x)(b,y) = (a + \varphi_x(b) + f(x,y), xy),$$

for any $a, b \in A$ and $x, y \in K$.

It remains only to check the group axioms. First, we can verify that (e,e) is the identity since $(e,e)(b,y) = (e + \varphi_e(b) + f(e,y), y) = (b,y)$, where $f(e,y) = e$ follows from the normalization property of f and φ_e is the identity since φ is a homomorphism. We likewise have $(a,x)(e,e) = (a + \varphi_x(e) + f(x,e), x) = (a,x)$.

Next, we can check that $(a,x)^{-1} = (-\varphi_{x^{-1}}(a + f(x,x^{-1})), x^{-1})$ is an inverse since $(a,x)(-\varphi_{x^{-1}}(a + f(x,x^{-1})), x^{-1}) = (a - \varphi_{xx^{-1}}(a + f(x,x^{-1})) + f(x,x^{-1}), xx^{-1}) = (a - a - f(x,x^{-1}) + f(x,x^{-1}), e) = (e,e)$. Since we have an identity, this right inverse must also be a left inverse.[5]

Lastly, we must check associativity. On the one hand, we have $((a,x)(b,y))(c,z) =$

[5] In general, let G be a group, $a \in G$ an element, and $b \in G$ a right inverse for a. This means that $bab = be = b$. If every element has a right inverse, then we have $ba = babb^{-1} = (bab)b^{-1} = bb^{-1} = e$, by the previous equation, which shows that b is also a left inverse for a.

$(a + \varphi_x(b) + f(x,y), xy)(c,z) = (a + \varphi_x(b) + f(x,y) + \varphi_{xy}(c) + f(xy,z), xyz)$, and, on the other hand, we have $(a,x)((b,y)(c,z)) = (a,x)(b + \varphi_y(c) + f(y,z), yz) = (a + \varphi_x(b) + \varphi_{xy}(c) + \varphi_x(f(y,z)) + f(x,yz), xyz)$. These are the same provided that $f(x,y) + f(xy,z) = \varphi_x(f(y,z)) + f(x,yz)$, which is precisely the cocycle condition that is assumed to hold for f.

If we have an extension G with homomorphism φ and factor set f_G, then we can also construct $G_\varphi^{f_G}$ as above. However, the previous theorem tells us that the resulting group is equivalent (and, hence, isomorphic) to G since it has the same cocyle f_G.

We now have all the pieces necessary to show the 1-to-1 correspondence between elements of the second cohomology group and (equivalence classes) of extensions.

Theorem B.2. *There is a bijection between $H_\varphi^2(K,A)$ and the equivalence classes of extensions of A by K with homomorphism φ.*

Proof. The bijection operates as taking the coset $f + B_\varphi^2(K,A)$ to the equivalence class of the group G_φ^f. The previous theorem tells us that this map is well-defined (any choice of cocycle from this coset gives rise to a group in the same equivalence class) and that this map is injective since two groups are in the same class iff their cocycles are from the same coset of $B_\varphi^2(K,A)$. Finally, we can see that this map is surjective since G is in the class that is the image of $f_G + B_\varphi^2(K,A)$. □

Finally, recall that the semidirect product $A \rtimes_\varphi K$ has a factor set that is uniformly identity. This means that identity coset $e + B_\varphi^2(K,A) \in H_\varphi^2(K,A)$ corresponds to the equivalence class of this semidirect product group.

Bibliography

[1] Samson Abramsky, Shane Mansfield, and Rui Soares Barbos. The cohomology of non-locality and contextuality. In *Proceedings of the 8th International Workshop on Quantum Physics and Logic*, pages 1–14. EPTCS, 2011, arXiv:1111.3620.

[2] Massoud Amini, Mehrdad Kalantar, and Mahmood M. Roozbehani. Hidden subhypergroup problem. 2006, arXiv:quant-ph/0609220.

[3] László Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 164–174. ACM, 1991.

[4] László Babai and Robert Beals. A polynomial-time theory of black-box groups I. In C. M. Campbell, E. F. Robertson, N. Ruskuc, and G. C. Smith, editors, *Groups St Andrews 1997 in Bath, I*, volume 260 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1999.

[5] Dave Bacon. How a clebsch-gordan transform helps to solve the heisenberg hidden subgroup problem. *Quantum Information & Computation*, 8(5):438–467, 2008, arXiv:quant-ph/0612107.

[6] Dave Bacon, Andrew M. Childs, and Wim van Dam. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In *Proceedings of the 46th IEEE Symposium on Foundations of Computer Science*, pages 469–478. IEEE Computer Society, 2005, arXiv:quant-ph/0504083.

[7] Ulrich Baum and Michael Clausen. Computing irreducible representations of supersolvable groups. *Mathematics of computation*, 63(207):351–359, 1994.

[8] Robert Beals and László Babai. Las Vegas algorithms for matrix groups. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 427–436. IEEE Computer Society, 1993.

[9] Juan Bermejo-Vega and Maarten Van den Nest. Classical simulation of abelian-group normalizer circuits with intermediate measurements. *Quantum Information & Computation*, 14(3–4):181–216, 2014, arXiv:1201.4867.

[10] Juan Bermejo-Vega, Cedric Yen-Yu Lin, and Maarten Van den Nest. The computational power of normalizer circuits over black-box groups. *Quantum Information & Computation*, Submitted, arXiv:1409.4800.

[11] Juan Bermejo-Vega, Cedric Yen-Yu Lin, and Maarten Van den Nest. Normalizer circuits and a gottesman-knill theorem for infinite-dimensional systems. *Quantum Information & Computation*, Submitted, arXiv:1409.3208.

[12] Juan Bermejo-Vega and Kevin C. Zatloukal. Abelian hypergroups and quantum computation. 2015, arXiv:1509.05806.

[13] Dan Boneh and Richard J. Lipton. Quantum cryptanalysis of hidden linear functions. In *Advances in Cryptology—CRYPTO '95*, pages 424–437. Springer, 1995.

[14] Raoul Bott and Loring W. Tu. *Differential Forms in Algebraic Topology.* Graduate Texts in Mathematics. Springer, 1982.

[15] P. Oscar Boykin, Tal Mor, Matthew Pulver, Vwani Roychowdhury, and Farrokh Vatan. A new universal and fault-tolerant quantum basis. *Information Processing Letters*, 75(3):101–107, 2000.

[16] Gilles Brassard and Peter Høyer. An exact quantum polynomial-time algorithm for simon's problem. In *Proceedings of the Fifth Israel Symposium on the Theory of Computing Systems (ISTCS '97)*, pages 12–23. IEEE Computer Society, 1997, arXiv:quant-ph/9704027.

[17] Johannes A. Buchmann. *Introduction to Cryptography.* Undergraduate Texts in Mathematics. Springer, 2nd edition, 2004.

[18] Kevin K. H. Cheung and Michele Mosca. Decomposing finite abelian groups. *Quantum Information & Computation*, 1(3):26–32, 2001, arXiv:cs/0101004.

[19] Efficient classical simulations of quantum Fourier transforms and normalizer circuits over abelian groups. Maarten van den nest. *Journal of Quantum Information & Computation*, 13(11–12):1007–1037, 2013, arXiv:1201.4867.

[20] Richard Cleve, Artur Ekert, Chiara Macchiavelo, and Michele Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London A*, 454(1969):339–354, 1998, arXiv:quant-ph/9708016.

[21] Thomas H. Cormen, Charles E. Leiserson, and Ronald R. Rivest. *Introduction to Algorithms.* MIT Press, 1996.

[22] Piergiulio Corsini and Violeta Leoreanu. *Applications of Hyperstructure Theory.* Advances in Mathematics. Springer, 2003.

[23] Etienne de Klerk and D.V. Pasechnik. On semidefinite programming relaxations of association schemes with application to combinatorial optimization problems. CentER Discussion Paper, 2009-54, 2009.

[24] David S. Dummit and Richard M. Foote. *Abstract Algebra.* John Wiley & Sons, Inc., third edition, 2004.

[25] Charles F. Dunkl. The measure algebra of a locally compact hypergroup. *Transactions of the American Mathematical Society*, 179:331–348, 1973.

[26] Abraham D. Flaxman and Bartosz Przydatek. *STACS 2005: 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, February 24-26, 2005. Proceedings*, chapter Solving medium-density subset sum problems in expected polynomial time, pages 305–314. Springer Berlin Heidelberg, 2005.

[27] Gerald B. Folland. *Fourier Analysis and Its Applications.* Pure and Applied Undergraduate Texts. American Mathematical Society, 2009.

[28] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theroy of NP-Completeness.* Series of Books in the Mathematical Sciences. W.H. Freeman, 1979.

[29] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[30] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction.* PhD thesis, California Institute of Technology, 1997.

[31] Dima Grigoriev. Testing the shift-equivalence of polynomials using quantum machines. *Journal of Mathematical Sciences*, 82(1):3184–3193, 1996.

[32] Sean Hallgren, Alexander Russell, and Amnon Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 627–635. ACM, 2000.

[33] Aram W. Harrow, Avinatan Hassadim, and Seth Lloyd. Quantum algorithms for solving linear systems of equations. *Physical Review Letters*, 103, 2009, arXiv:0811.3171.

[34] Allen Hatcher. *Algebraic Topology.* Cambridge University Press, 2001.

[35] Peter Høyer. Conjugated operators in quantum algorithms. *Physical Review A*, 59(5), 1999.

[36] Gábor Ivanyos, Frédéric Magniez, and Miklos Santha. Efficient quantum algorithms for some instances of the non-abelian hidden subgroup problem. In *Proceedings of the thirteenth annual ACM symposium on parallel algorithms and architectures (SPAA '01)*, pages 263–270. ACM, 2001, arXiv:quant-ph/0102014.

[37] Gábor Ivanyos, Luc Sanselme, and Miklos Santha. An efficient quantum algorithm for the hidden subgroup problem on nil-2 groups. *Algorithmica*, 62(1–2):480–498, 2012, arXiv:0707.1260.

[38] Robert I. Jewett. Spaces with an abstract convolution of measures. *Advances in Mathematics*, 18:1–101, 1975.

[39] Gregory Karpilovsky. *Clifford Theory for Group Representations.* North-Holland, 1989.

[40] Alexei Y. Kitaev. Quantum measurements and the abelian stabilizer problem. *Electronic Colloquium on Computational Complexity*, 3(3), 1996, arXiv:quant-ph/9511026.

[41] Alexei Y. Kitaev. Anyons in an exactly solved model and beyond. *Annals of Physics*, 321(1):2–111, 2006, arXiv:cond-mat/0506438.

[42] Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35(1):170–188, 2005, arXiv:quant-ph/0302112.

[43] Serge Lang. *Algebra.* Graduate Texts in Mathematics. Springer, third edition, 2005.

[44] Grigory L. Litvinov. Hypergroups and hypergroup algebras. *Journal of Soviet Mathematics*, 38(2):1734–1761, 1987, arXiv:1109.6596.

[45] Seth Lloyd, Silvano Garnerone, and Paolo Zanardi. Quantum algorithms for topological and geometrical analysis of data. *Nature Communications*, 7, 2016, arXiv:1408.3106.

[46] Chris Lomont. The hidden subgroup problem — review and open problems. 2004, arXiv:quant-ph/0411037.

[47] Saunders MacLane. *Homology.* Classics in Mathematics. Springer, 1995.

[48] Michele Mosca. *Quantum Computer Algorithms.* PhD thesis, University of Oxford, 1999.

[49] Michele Mosca and Artur Ekert. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In *Selected papers from the First NASA International Conference on Quantum Computing and Quantum Communication (QCQC '98)*, pages 174–188. Springer-Verlag, 1998, arXiv:quant-ph/9903071.

[50] A. Dehghan Nezhad, S. M. Moosavi Nejad, M. Nadjafikhah, and B. Davvaz. A physical example of algebraic hyperstructures: Leptons. *Indian Journal of Physics*, 86(11):1027–1032, 2012.

[51] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information.* Cambridge University Press, 2000.

[52] Oded Regev. Quantum computation and lattice problems. *SIAM Journal on Computation*, 33(3):738–760, 2004, arXiv:cs/0304005.

[53] Ronald L. Rivest, Adi Shamir, and Leonard Adleman. A method of obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[54] Richard L. Roth. Character and conjugacy class hypergroups of a finite group. *Annali di Matematica Pura ed Applicata*, 105(1):295–311, 1975.

[55] Joseph Rotman. *An Introduction to the Theory of Groups.* Graduate Texts in Mathematics. Springer, 1994.

[56] Ákos Seress. *Permutation Group Algorithms.* Cambridge Tracts in Mathematics. Cambridge University Press, 2003.

[57] Jean-Pierre Serre. *Linear Representations of Finite Groups.* Graduate Texts in Mathematics. Springer, 1977.

[58] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997, arXiv:quant-ph/9508027.

[59] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1473–1483, 1997.

[60] R. Spector. Mesures invariantes sur les hypergroupes. *Transactions of the American Mathematical Society*, 239:147–165, 1978.

[61] Walter D. Strangl. Counting squares in $\mathbb{Z}_n$. *Mathematics Magazine*, 69(4):285–289, 1996.

[62] John Watrous. Quantum algorithms for solvable groups. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 60–67. ACM, 2001, arXiv:quant-ph/0011023.

[63] N. J. Wildberger. Finite commutative hypergroups and applications from group theory to conformal field theory. In *Applications of Hypergroups and Related Measure Algebras*, volume 183 of *Contemporary Mathematics*, pages 428–449. AMS, 1995.

[64] N. J. Wildberger. Lagrange's theorem and integrality for finite commutative hypergroups with applications to strongly regular graphs. *Journal of Algebra*, 182:1–37, 1996.

[65] N. J. Wildberger. Duality and entropy for finite commutative hypergroups and fusion rule algebras. *Journal of the London Mathematical Society*, 56:275–291, 1997.

[66] Kevin C. Zatloukal. Classical and quantum algorithms for testing equivalence of group extensions. In Simone Severini and Fernando Brandao, editors, *8th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC 2013)*, volume 22 of *Leibniz International Proceedings in Informatics*, pages 126–145, 2013, arXiv:1305.1327.