On the Quantum Base Size of a Group

A Thesis
Presented to
The Division of Mathematics and Natural Sciences
Reed College

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Arts

Tristan Wylde-LaRue

May 2019

Approved for the Division
(Mathematics)

_____

Jamie Pommersheim

Acknowledgements

First and foremost, I would like to thank my family for their incredible patience and kindness, I could not have made it through this without them. I also need to express my overwhelming thanks to Roger for the motivation and support he gave me.

I am extremely grateful for my advisor Jamie who guided me through all of this. It was amazing to have been able to take classes with him and share in his excitement for everything math related. Undoubtedly, he is one of the best professors I have ever had. For thesis, he has been nothing but kind and patient. It's been a pleasure getting to work with him individually over the past year. It's very sad to think that I won't ever have another Jamie class.

The entire math community at Reed has been nothing but great to me in my time here. I have to thank all of my friends in the math department for the long nights working on real analysis problem sets. Without them, I certainly would not have enjoyed my math education as much as I did.

Finally, I want to thank all of my friends who have put up with me this last year working on this. I wish I could name everyone individually, but it would be an extremely long list. However, I need to give special thanks to Raina, Duncan and Sarina for really getting me through this year. Parts of this year have been very rough and they were always there to help without any second thoughts. I'm infinitely indebted to them.

Table of Contents

Abstract

In this thesis we study the query complexity of quantum oracle problems in which a learner must identify a hidden group element by querying its action on a set. A recent paper of Pommersheim and Copeland connects this problem to a classical problem in the representation theory of finite groups. We extend their work by computing the query complexity for symmetric oracle problems associated to the dihedral groups.

Dedication

To my sister Cordelia who inspired my love of math.

Introduction

In 1994, Peter Shor showed that if a working quantum computer could be built, it would be capable of factoring integers exponentially faster than is thought to be possible on a classical computer. This was perhaps the most prominent of a series of successes that publicly established the field of quantum computing as something worth seriously researching. It is a very unique subject today, as it is one of the few that is truly interdisciplinary. Influential papers in the field are routinely published by researchers in mathematics, physics, computer science and even occasionally chemistry departments.

The primary goal of this thesis is to introduce a reader familiar with linear algebra and abstract algebra to an interesting open problem at the intersection of quantum computing and representation theory. The first two chapters seek to give a self-contained introduction to quantum computing and representation theory respectively. The section on quantum computing is intended slightly towards someone with a mathematical background rather than a physics or computer science one. Finally, chapter three introduces the problem and the paper that motivates it and in chapter four explicit examples are computed.

Chapter 1

Quantum Computing

1.1 Learning Problems

1.1.1 A Motivating Example

Suppose you're involved in a simple card game: A dealer places two cards (you can assume that the cards are either red or black, with equal probability of occurring) face down on a table. You win the game and a substantial prize if you can guess whether the two face-down cards share the same color. You're allowed to ask the dealer to reveal cards to you, but for each card revealed your potential prize gets smaller.

How many of the cards do you need to see to determine with certainty whether the cards share the same color? Maybe you don't need to know for certain. How does the probability of you being able to guess the correct answer relate to the number of cards seen?

If you have no information at all, you can't do substantially better (or worse) than a fifty-percent chance. In fact, seeing only a single card, does not give you any more knowledge of what the answer to the question is. If you wanted to know with certainty what the answer was, you would need to see both cards.

This is of course a very simple game, but it is an example of a type of problem that we refer to as *learning problems* or often *oracle problems*. These problems consist of a learner who is trying to determine the answer to some question, generally to find the output of a certain function. The learner starts out with incomplete information, but is given access to an *oracle function* she can query to gain more information. An oracle function–also sometimes called a black-box function–is one that you can evaluate the oracle at any input of your choosing, but you have no information about the function other than its responses to your inputs. The learner's goal is to determine the answer to the question in as few questions as possible.

We rephrase the scenario given above in this language. Choose 0 to represent a black card and 1 to represent a red card. Suppose the two facedown cards are labeled a and b.

Example 1.1. Given oracle access to a function $f : \{a, b\} \to \{0, 1\}$. What is the minimum number of queries required to determine $f(a) \oplus f(b)$ where $\oplus$ denotes

addition mod 2?

For the reasons above, you can not do it fewer than two queries. We call the minimum number of queries you need to make in order to solve the problem the *query complexity* of the problem. In this case, the query complexity is 2.

1.2 Quantum Computing

Now we introduce the concepts required for quantum computation. In the quantum world, the *qubit* is the fundamental unit of information. In the same way that we can express any classical algorithm as a circuit of logical operations on bits, we can express any quantum operation as a sequence of *unitary* operations on *qubits*.

1.2.1 Qubits

The following treatment of quantum computing is largely taken from Nielsen and Chuang [5] and Nielsen's more recent article [4]. The goal of this chapter is to introduce a minimal model of quantum computing that is sufficient for the tasks we will need in chapters 2 and 3. It can be hard to follow quantum computing if one is not already familiar with it as there is a lot of notation and formal definitions. Moreover, since quantum mechanics has a reputation for being unintuitive, it is often hard to feel like the intuitions one has are correct. However, I think that quantum circuits do a very good job of capturing the intuition from classical computing and so I have tried to include many simple examples of circuits. For a very detailed and gentle introduction to the subject, I strongly recommend reading the article [4]. For a much more expansive view of the entire subject of quantum computing and topics not covered here, I would refer to Nielsen and Chuang's text [5].

Whereas the state of a bit has *two* possible values—either 0 or 1—the state of a qubit is a unit vector in complex vector space spanned by *two* basis states. We give the following precise definition

Definition 1.1. The state of a *qubit* is a unit vector in $\mathbb{C}^2$. That is, a pair of complex numbers

$$(\alpha, \beta) \in \mathbb{C}^2$$

satisfying the normalization condition

$$|\alpha|^2 + |\beta|^2 = 1$$

We refer to the space $\mathbb{C}^2$ as the *state space* of a qubit and consider it to be an inner product space equipped with the standard inner product on $\mathbb{C}^n$.[1] Recall the

[1] In quantum mechanics literature, state spaces of quantum systems are always referred to as *Hilbert spaces* regardless of their dimension. This can be confusing as *Hilbert spaces* in mathematics are generally associated with infinite dimensional spaces. Since all of the spaces in this thesis will be finite dimensional (and isomorphic to $\mathbb{C}^{2^n}$), it is easier to avoid the confusion and use the term state space instead.

inner product of two vectors $x, y \in \mathbb{C}^n$ is given by

$$\langle x, y \rangle = x_1\overline{y_1} + x_2\overline{y_2} + ... + x_n\overline{y_n} = \sum_{i=1}^{n} x_i\overline{y_i}$$

where $\overline{(\ )}$ denotes complex conjugation.

We choose the standard basis $\{(0,1),(1,0)\}$ to represent $\mathbb{C}$. In order to make the analogy with bits more clear, we denote

$$|0\rangle := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle := \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The symbol $|\ \rangle$ is called a *ket* and is used to denote a vector representing a quantum state. It comes from the *bra-ket* notation which is widely used in the greater theory of quantum mechanics.

1.2.2 Measurement

Although a qubit has a quantum state in $\mathbb{C}^2$, the we cannot directly observe this quantum state. Despite having what seems like a much richer state, when we try to measure qubits using any sort of detector, they snap into the states $|0\rangle$ or $|1\rangle$. This of course defies any reasonable expectations one would have about the world, and has raised much philosophical debate about how it should be interpreted. However, regardless of what interpretation we take, it remains an experimental fact. The following postulate formalizes this concept as a process called *measurement in the computational basis*.

Postulate 1. *Given a qubit ψ, we may perform* measurement *of ψ in the* computational basis. *If ψ was in state $\alpha|0\rangle + \beta|1\rangle$ prior to the measurement, after the measurement it will be in state $|0\rangle$ with probability $|\alpha|^2$ and $|1\rangle$ with probability $|\beta|^2$.*

This is only a specific case of a much more general postulate of measurement in quantum mechanics. However, this conception of measurement will be sufficient enough for many of the results in quantum computing and all of the results that we will need to consider. A more in depth discussion of the postulates of quantum mechanics and how they relate to quantum computing can be found in [5].

If we naively think of the state of a qubit as being in $\mathbb{R}^2$ rather than $\mathbb{C}^2$ by just ignoring the imaginary part, we can generate some limited geometric intuition about what is going on.

A measurement on the qubit pictured in Figure 1.1 will return 0 with probability 3/4 and 1 with probability 1/4. This perspective gives some justification for why the state of a qubit must be a unit vector. When written in the standard basis, the coefficients of $|0\rangle$ and $|1\rangle$ correspond to the probability of a measurement producing that outcome. Thus requiring that this state be a unit vector is equivalent to requiring that the total probabilities sum to one.

This also motivates the final postulate of quantum mechanics that we will need to formalize this model of quantum computation.

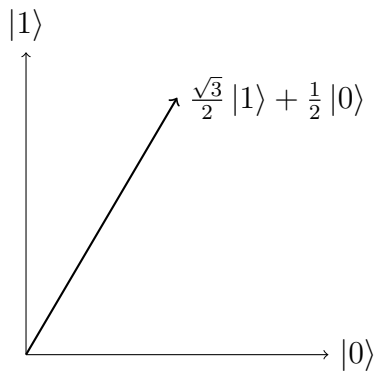


Figure 1.1: A qubit with state $\frac{\sqrt{3}}{2}|1\rangle + \frac{1}{2}|0\rangle$ pictured in the plane by ignoring the imaginary part.

Postulate 2. *The valid operations on quantum states are those given by* unitary transformations*.*

The unitary transformations on $\mathbb{C}^n$ are exactly the *isometries* of $\mathbb{C}^n$ that are linear. A linear map $U : \mathbb{C}^n \to \mathbb{C}^n$ is said to be an isometry if $\|Uv\| = \|v\|$ for every vector $v \in \mathbb{C}^n$, in other words, if it preserves the length of each vector. This is, perhaps not too surprising, given that preserving the length of each vector is necessary condition to ensure that the probabilities sum to one.

Unitary maps can also be defined algebraically. The following definition is probably the most commonly given one for a unitary map.

Definition 1.2. A matrix $U : \mathbb{C}^n \to \mathbb{C}^n$ is said to be *unitary* if it satisfies the identity $U^{-1} = U^\dagger$ where $U^\dagger$ denotes the conjugate-transpose of U.

1.2.3 Gates

By Postulate 2, the valid operations on a qubit are the 2×2 unitary matrices.

This part of quantum computing can be confusing at first, since the notation constantly switches between kets and matrices. For this reason, we have tried to write the examples clearly using each notation.

Example 1.2. The quantum `NOT` gate is the single qubit gate defined by

$$|0\rangle \mapsto |1\rangle$$
$$|1\rangle \mapsto |0\rangle$$

It should be noted that it is common to refer to the quantum `NOT` gate by the letter X.

In standard vector notation this is

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \mapsto \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$
$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Which gives us that the quantum `NOT` gate is represented by the matrix

$$X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Example 1.3. A very important single qubit gate is the *Hadamard* gate. The Hadamard gate, denoted H, can be defined in terms of its action on the standard basis as

$$|0\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$
$$|1\rangle \mapsto \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

These states appear frequently and so it is common to give them their own notation. We define

$$|+\rangle := \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

and

$$|-\rangle := \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$

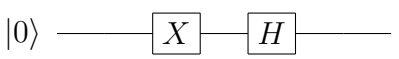
The Hadamard gate has matrix representation

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

1.2.4 Quantum Circuits

Much like in classical computation, it is convenient to represent manipulations of qubits as circuits. A *quantum circuit* is made up of qubits, wires, and gates. It is perhaps easiest to define by giving examples.

Consider the following very simple circuit

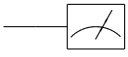


This circuit depicts a single qubit starting in state $|0\rangle$. As it travels through the wire left to right, X takes it to $|1\rangle$ and then H takes $|1\rangle$ to $|-\rangle$. This circuit represents the equation

$$|-\rangle = HX\,|0\rangle$$

Of course by associativity, we could have used a single gate HX.

We allow for one special gate that is not unitary to denote measurement. The measurement gate is depicted



1.2.5 Multiple qubits

Up until this point we have only considered single-qubit systems. However, computation is not very interesting if we're only restricted to a single qubit.

Consider a string of two bits. This string has four possible values: 00, 01, 10, 11. Much like in the single bit case, while a string of two bits has four possible states, a string of two qubits has four basis states. Keeping with the analogy to bits from before, we number these basis vectors in binary and call them $|00\rangle, |01\rangle, |10\rangle, |11\rangle$ which represent $(1,0,0,0), (0,1,0,0), (0,0,1,0)$, and $(0,0,0,1)$ respectively.

Formally, the state space of two qubits is given by $\mathbb{C}^2 \otimes \mathbb{C}^2$, where $\otimes$ denotes the *tensor product* of two vector spaces. If ψ and φ are two qubits with states $|\psi\rangle$ and $|\varphi\rangle$ respectively, then we denote their combined state $|\psi\rangle |\varphi\rangle := |\psi\rangle \otimes |\varphi\rangle$. It is also common to write $|\psi\varphi\rangle$.

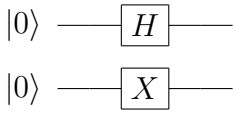
Definition 1.3. Let V and W be vector spaces with bases $\{v_1, ...v_n\}$ and $\{w_1, ...w_m\}$ respectively. Let B be the set of all formal pairs $v_i \otimes w_j$ (read v_i tensor v_j) for each $1 \leq i \leq n$ and $1 \leq j \leq m$.

The *tensor product space* $V \otimes W$ is defined to be the vector space spanned by the basis B.

The tensor product is a complicated construction and defining it in full generality here would take us beyond the scope of this thesis. Luckily, all of the spaces we are working with are finite and we have already chosen bases, so we don't need to worry about this. For the purposes of quantum computing this definition is sufficient.

In general, the combined state space of two quantum systems is given by their tensor product. The state space of an n-qubit system is $(\mathbb{C}^2)^{\otimes n} = \mathbb{C}^2 \otimes \mathbb{C}^2 ... \otimes \mathbb{C}^2$ (n times). A useful way to think about this is that the tensor product combines the two states without adding any relationship between the 2 qubits.

This construction naturally allows us to extend our circuit diagrams to multiple qubits. Consider the following two qubit circuit

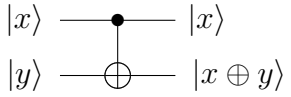


This circuit starts in state $|00\rangle$. The first qubit gets mapped by the Hadamard to $|+\rangle$ and the second qubit is mapped by the X gate to $|1\rangle$. Thus it returns $|+\rangle |1\rangle$. Algebraically, this expresses the equation

$$(H \otimes X)(|00\rangle) = |+\rangle |1\rangle$$

This example could have really just been two single qubit circuits. The two wires run entirely in parallel here. However, most quantum circuits are not like this. Most of the two qubit gates cannot be split up as a product of single qubit gates.

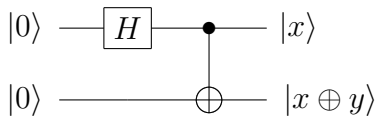
Consider the `Controlled-NOT` gate defined as



It has the following action on the standard basis: If the top wire is $|0\rangle$, then the bottom wire is unchanged. If the top wire is a $|1\rangle$, then it acts as a not gate on the bottom wire. Thus it has matrix representation

$$\texttt{CNOT} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

As an example consider the circuit



This circuit starts in state $|00\rangle$. The first timestep applies a Hadamard to the top wire and the identity to the bottom wire, thus it changes the state to $|+\rangle|0\rangle$. To compute the action of the `Controlled-NOT` notice that

$$|+\rangle|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle + |1\rangle\big)|0\rangle = \frac{1}{\sqrt{2}}\big(|0\rangle|0\rangle + |1\rangle|0\rangle\big) = \frac{1}{\sqrt{2}}\big(|00\rangle + |10\rangle\big)$$

Thus,

$$\texttt{CNOT}|+\rangle|0\rangle = \texttt{CNOT}\frac{1}{\sqrt{2}}\big(|00\rangle + |10\rangle\big) = \frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big)$$

Therefore this circuit sends $|0\rangle|0\rangle$ to $\frac{1}{\sqrt{2}}\big(|00\rangle + |11\rangle\big)$.

This state is an example of another important property of quantum systems that cannot be seen with only one qubit. It is what is referred to as an *entangled state* meaning that it has no representation as the product of single qubit states.

1.2.6 Quantum Oracles

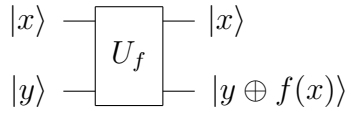
The majority of the problems in quantum computing will be phrased as oracle problems. The motivation for this is that oracle problems are easy to express and of importance classically, yet extend very naturally to analogous problems in the quantum setting.

The way that we give circuits for oracle problems is by adding an extra black-box gate to our collection. The issue here is that the black-box functions in most oracle problems are not unitary and thus not realizable on a quantum computer. The way to fix this, is to encode the oracle function f as a unitary matrix in a fashion similar to the `Controlled-NOT` gate.

Given a function $f : \{0,1\}^n \to \{0,1\}^m$, let $U_f : (\mathbb{C}^2)^{\otimes n} \bigotimes (\mathbb{C}^2)^{\otimes m} \to (\mathbb{C}^2)^{\otimes n} \bigotimes (\mathbb{C}^2)^{\otimes m}$ be the unitary map defined by

$$|x\rangle \otimes |y\rangle \mapsto |x\rangle \otimes |f(x) \oplus y\rangle$$

In a circuit we write this as



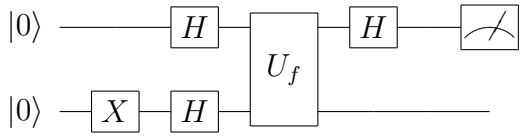
In this context, the wires labeled $|x\rangle$ and $|y\rangle$ represent the states of potentially multiple qubits and are referred to as *registers*. We refer to the register that is fixed by the oracle as the *query register* and the other as the *response register*.

1.2.7 Deutsch's Algorithm

First, let's revisit the problem given in the beginning this chapter in Example 1. Recall we had the following problem:

Problem 1. Given oracle access to a function $f : \{0,1\} \to \{0,1\}$. What is the minimum number of queries required to determine $f(0) \oplus f(1)$ where $\oplus$ denotes addition mod 2?

Consider the following circuit



The circuit starts in state $|00\rangle$, then the X gate on the bottom wire takes it to $|01\rangle$. The two Hadamard gates take $|01\rangle$ to $|+\rangle|-\rangle$. We rewrite this state in the standard basis

$$|+\rangle|-\rangle = \frac{1}{2}\big(|0\rangle + |1\rangle\big)\big(|0\rangle - |1\rangle\big) = \frac{1}{2}\big(|00\rangle - |01\rangle + |10\rangle - |11\rangle\big)$$

$$\begin{aligned} U_f\big(|+\rangle|-\rangle\big) &= \frac{1}{2}U_f\big(|00\rangle - |01\rangle + |10\rangle - |11\rangle\big) \\ &= \frac{1}{2}\big(U_f|00\rangle - U_f|01\rangle + U_f|10\rangle - U_f|11\rangle\big) \\ &= \frac{1}{2}\big(|0\rangle|0 \oplus f(1)\rangle - |0\rangle|1 \oplus f(0)\rangle + |1\rangle|0 \oplus f(1)\rangle - |1\rangle|1 \oplus f(1)\rangle\big) \end{aligned}$$

Now applying the Hadamard to the top gate takes this state to

$$\frac{1}{2}\big(|+\rangle|0 \oplus f(1)\rangle - |+\rangle|1 \oplus f(0)\rangle + |-\rangle|0 \oplus f(1)\rangle - |-\rangle|1 \oplus f(1)\rangle\big)$$

which expands to

$$\frac{1}{4}\big[\,|0\rangle|0 \oplus f(1)\rangle + |1\rangle|0 \oplus f(1)\rangle + \ldots - |0\rangle|1 \oplus f(1)\rangle + |1\rangle|1 \oplus f(1)\rangle\,\big]$$

If $f(0) \neq f(1)$, then $|0\rangle|0 \oplus f(1)\rangle = |0\rangle|1 \oplus f(1)\rangle$ and so those will cancel in the above sum. Writing this out for all the states, one obtains that the first qubit is $|1\rangle$ if and only if $f(0) = f(1)$.

Chapter 2

Representation Theory of Finite Groups

2.1 Introduction

The purpose of this chapter will be to introduce the terminology and results from representation theory that we will need to investigate symmetric oracle problems in chapters 3 and 4. A reader familiar with the basic results of the representation theory of finite groups can skip ahead to chapter 3.

This treatment of representation theory largely follows James and Liebeck [**?**]. We cannot hope to give more insightful proofs than the ones found there, so we omit many proofs here and instead leave the reader to refer to [3].

Although there are many analogues of the results given here for fields other than $\mathbb{C}$ and for infinite groups, we will not express them in their full generality. The representations that will arise in our study of quantum oracle problems will be of finite groups and over $\mathbb{C}$. Moreover, many proofs in representation theory are substantially simpler and more elegant when the underlying field is restricted to $\mathbb{C}$ since it is algebraically closed. This means that $\mathbb{C}$ is often the prototypical field to study representations in.

2.2 Representations

Loosely speaking, a representation of a group G is an encoding of G as a set of linear maps on some vector space V. When a basis of V is specified, this yields an explicit representation of each $g \in G$ as a matrix. Looking at the possible ways of representing G as a matrix group uncovers much information about the structure of G that may have been hidden otherwise. In particular, any encoding of G into a set of matrices will be unique only up to a change of basis. This exposes a connection between the symmetry arising in vector spaces by change of basis and symmetry inherent in the structure of groups, arising from conjugation.

Definition 2.1. Let G be a group and V be a vector space over a field F. A *representation* of G is a homomorphism $\rho : G \to \mathrm{Aut}(V)$, where $\mathrm{Aut}(V)$ denotes

the group of invertible linear maps from V to itself. Although, the representation is defined by the homomorphism ρ, it is common instead to refer to the representation by the vector space V it is acting on.

Equivalently, a representation can be characterized as an action of a group G on a vector space V. In this case, the action is required to be compatible with the vector space structure on V. That is, in addition to the requirements for an action of a group on a set, we must have

$$g \cdot (\alpha v_1 + v_2) = g \cdot \alpha(v_1) + g \cdot (v_2)$$

for all $g \in G,\ v_1, v_2 \in V$, and $\alpha \in F$. We denote the action of $g \in G$ on a vector v as $g \cdot v$ or gv.

There is yet a third way to express the information of a representation. Although this is the most abstract, it provides the most concrete language for working with representations.

One can view the group algebra as an encoding of a group into the structure of polynomial arithmetic.

Definition 2.2. Let G be a finite group of order n. The *group algebra* of G, denoted $\mathbb{C}[G]$ or $\mathbb{C}G$, is defined as the vector space over $\mathbb{C}$ with one basis element for every $g \in G$. Thus every element of $\mathbb{C}[G]$ is of the form

$$\alpha_1 g_1 + \alpha_2 g_2 + ... + \alpha_n g_n$$

where $\alpha_1, ..., \alpha_n \in \mathbb{C}$ and $\{g_1, ..., g_n\} = G$.

The operations of addition, and scalar-multiplication are inherited from the properties of vector spaces. In addition we define multiplication first and monomials and then extend it distributively. The multiplication of monomials is defined to follow the group structure, that is For any $\alpha_1, \alpha_2 \in \mathbb{C}$ and $g_1, g_2 \in G$ we define

$$(\alpha_1 g_1)(\alpha_2 g_2) = (\alpha_1 \alpha_2)(g_1 g_2)$$

where $\alpha_1 \alpha_2$ is the product of α_1, α_2 in $\mathbb{C}$ and $g_1 g_2$ is the product of g_1, g_2 in G. In particular, this means $\mathbb{C}[G]$ is a commutative ring if and only if G is abelian.

The connection between the group algebra and representations is made clear using the theory of modules. For a comprehensive treatment of modules, refer to Dummit and Foote [2] chapter 10.

Definition 2.3. Let R be a ring. An R-module is a pair $(M, \cdot)$ where M is an abelian group and an action of R on M, that is a map $R \times M \to M$ that is associative and distributes over addition in R.

One might think of an R-module as the ring analogy to a G-set. Although the action is in some sense the most defining part of an R-module, the object M being acted on is referred to as the R-module

A representation $\rho : G \to \mathrm{Aut}(V)$ defines a $\mathbb{C}G$-module structure on V by the action

$$(\alpha_1 g_1 + ... + \alpha_n g_n)v := \alpha_1 \rho(g_1)v + ... + \alpha_n \rho(g_n)v$$

Conversely, a $\mathbb{C}G$-module with action $\varphi : \mathbb{C}G \times V \to V$ determines a representation ρ defined by $\rho(g) = \varphi(1g)$. This establishes a correspondence between representations of G and $\mathbb{C}G$-modules.

2.3 The Structure of Representations

In linear algebra one studies linear maps looking at the subspaces that are invariant under them. The big result in linear algebra is that a linear map is determined entirely by its invariant subspaces. The methods for studying representations should feel analogous to the methods used for studying linear maps.

Instead of considering the subspaces invariant under just one linear map, we want the subspaces that are invariant under the action of the entire group.

Definition 2.4. Let ρ be a representation of a group G on a vector space V. A *subrepresentation* is a vector subspace $W \subset V$ that is invariant under the action of G. That is, $g \cdot w \in W$ for all $g \in G$, $v \in W$.

A representation is said to be irreducible if it has no nontrivial subrepresentations (the only G-invariant subspaces of V are $\{0\}$ and V itself).

Theorem 2.1 (Maschke's Theorem)**.** *Let G be a finite group and V a representation of G over $\mathbb{C}$. If W is a subrepresentation of V, then there is a subrepresentation W' such that $V = W \oplus W'$. (Here and throughout this chapter, we use $\oplus$ to denote the direct sum of vector spaces).*

Proof. We are given that W is a subrepresentation of V. Let $\pi : V \to W$ be a projection of V onto W. Although $\ker \pi \oplus W = V$, we do not know that $\ker \pi$ must be a subrepresentation. In fact, it will not be a subrepresentation in general. We will need to modify it slightly.

We construct a new function $\pi_G : V \to W$ by averaging over G. Define it by

$$\pi_G(v) = \frac{1}{|G|} \sum_{g \in G} g \cdot \pi(g \cdot v)$$

Now let $h \in G$ and $u \in V$.

$$h \cdot \pi_G(u) = \frac{1}{|G|} \sum_{g \in G} hg \cdot \pi(g \cdot v)$$

We can reindex our sum to get

$$h \cdot \pi_G(u) = \frac{1}{|G|} \sum_{g \in G} g \cdot \pi(h^{-1} g \cdot v)$$

Checking this shows that it is indeed fixed under the action of h and so $ker\pi$ must be a subrepresentation. □

Theorem 2.2. *Every representation over $\mathbb{C}$ decomposes into irreducible subrepresentations. i.e. If (ρ, V) is a representation of group G over $\mathbb{C}$, then there exist irreducible subrepresentations $W_1, ...W_n$ such that $V = W_1 \oplus W_2 \oplus ... \oplus W_n$.*

Proof. If V is irreducible, then we are done. Otherwise, it must have some subrepresentation W. By Maschke's Theorem, there is a subrepresentation U such that $V = U \oplus W$. Now, if U and W are irreducible, we are done. Otherwise, they admit subrepresentations by Maschke's Theorem. We repeat this process inductively until each subrepresentation is irreducible. This is guaranteed to terminate in a finite number of iterations because $\dim(V)$ is finite. □

This result establishes the basic philosophy of representation theory: the set of irreducible representations of a group tells us almost everything about the group. Indeed, the representation theory of finite groups was one of the key developments that lead to the classification of finite simple groups. Much of the rest of this chapter will be devoted to developing tools that help us find the set of irreducible representations. However, there is one caveat though, we do not as of yet know this decomposition into irreducible subrepresentations is unique. With the addition of a bit more machinery, we will show that this is indeed the case.

2.4 Character Theory

Definition 2.5. Let M be an $n \times n$ matrix. The *trace* of M is defined to be the sum of the diagonal entries of M i.e.

$$\mathrm{Tr}\,(M) := \sum_{i=1}^{n} M_{ii}$$

Alternatively, some authors define the trace to be the sum of the eigenvalues of M. As it turns out, these two values are always equivalent. This result can be seen as a consequence of the Jordan canonical form.

Definition 2.6. Let $\rho : G \to V$ be a representation. The map $\chi : G \to \mathbb{C}$ given by $\chi(g) = \mathrm{Tr}\,\rho(g)$ is called the *character* of the representation V.

At first glance, it doesn't seem like characters should be very useful at all. A character only associates a single complex number to each $g \in G$ while a representation associates an entire matrix of complex numbers. Why are we throwing all of this information away?

Surprisingly, the characters encode almost all of the information we need about the group. This is maybe the dream of every student taking their first linear algebra course. Many students complain about having to row reduce or write out determinants for 4×4 matrices, but very few complain about at being asked to add up the four numbers on the diagonal. Many of the results on characters seem too good to be true.

Although it seems crazy at first to take the trace and throw everything else away, the fact that this works isn't actually so surprising upon further inspection.

Definition 2.7. A *class function* of a group G is a function $\varphi : G \to \mathbb{C}$ that is constant on the conjugacy classes of G. That is, for all $g, h \in G$, $\varphi(g) = \varphi(hgh^{-1})$.

The set of class functions on a group G form a vector space under pointwise addition. This space has an inner-product given by

$$\langle \varphi, \psi \rangle := \frac{1}{|G|} \sum_{g \in G} \varphi(g)\overline{\psi(g)}$$

Is is often convenient to write explicit class functions as tables of the form

g	g_1	g_2	$\cdots$	g_n
$\varphi(g)$	$\varphi(g_1)$	$\varphi(g_2)$	$\cdots$	$\varphi(g_n)$

where $g_1, ...g_n$ are representatives from each of the n distinct conjugacy classes of G.

Theorem 2.3. *The characters of a group G are class functions from $G \to \mathbb{C}$.*

Proof. Let χ be a character of G arising from the representation ρ. Let $g, h \in G$. Recall from linear algebra that the trace of a matrix is equivalent to the sum of its eigenvalues. Thus the trace is invariant under change of basis. Therefore

$$\chi(hgh^{-1}) = \operatorname{Tr} \rho(hgh^{-1}) = \operatorname{Tr} \rho(h)\rho(g)\rho(h^{-1}) = \operatorname{Tr} \rho(g) = \chi(g)$$

□

This result reveals some of the intuition behind why the trace becomes so useful. The trace can be thought of in some sense as the simplest map from $\mathrm{GL}(n, \mathbb{C}) \to \mathbb{C}$ that is invariant under change of basis.

2.4.1 Irreducible Characters

Definition 2.8. Let G be a group. We say that χ is a character of G if it is the character of some representation of G.

Moreover, if χ is the character of an irreducible representation of G, we say that χ is an *irreducible character of G*.

The irreducible characters are particularly special class functions.

Theorem 2.4. *The set of distinct irreducible characters of a group G form an orthonormal basis for the space of class functions on G.*

Refer to [3] Chapter 14

Theorem 2.5. *Let V be a representation of a group G and let ψ be its associated character. V is irreducible if and only if and only if $\langle \psi, \psi \rangle = 1$.*

Proof. See [3] Chapter 14 □

Corollary 2.6. *The number of irreducible characters of G is equal to the number of conjugacy classes of G.*

Proof. Let $C_1, C_2, ..., C_n$ be the conjugacy classes of G. For each $1 \leq i \leq n$ define $\varphi : G \to \mathbb{C}$ by

$$\varphi_i(G) = \begin{cases} 1 & \text{if } g \in C_i \\ 0 & \text{if } g \notin C_i \end{cases}$$

The set $\{\varphi_1, ..., \varphi_n\}$ forms an orthonormal basis for the space of class functions from G to $\mathbb{C}$. In fact, one might consider this to be the standard basis on this space. By the previous theorem, the set of irreducible characters also form a basis for the space of class functions, so they must be the same size. □

Corollary 2.7. *Let $\chi_1, ..., \chi_n$ be the irreducible characters of a group G. Let $\varphi : G \to \mathbb{C}$ be a class function on G. Then φ decomposes as*

$$\varphi = \alpha_1\chi_1 + ... + \alpha_n\chi_n$$

where $\alpha_i = \langle \varphi, \chi_i \rangle$.

Proof. This follows straightforwardly from the properties of inner-products. □

Lemma 2.8. *Let (φ, V) be a representation of group G with subrepresentations U and W such that $V = U \oplus W$. Let χ_V, χ_U, χ_W be the characters of V, U, and W respectively.*

Then $\chi_V = \chi_U + \chi_W$.

Theorem 2.9. *In the special case that φ is a character, then its decomposition into irreducibles given by*

$$\varphi = m_1\chi_1 + m_2\chi_2 + ... + m_n\chi_n$$

is guaranteed to have non-negative integral coefficients. That is, each $m_1, ..., m_n$ must be a nonnegative integer. These integers count the multiplicity of the associated irreducible subrepresentations appearing in the decomposition of the associated representation of ψ.

Proof. Let (ρ, V) be a representation of G and χ the character of ρ. By Theorem 2.3, there exist irreducibles $W_1, ..., W_n$ such that $V = W_1 \oplus \cdots \oplus W_n$. Let $\psi_1, ..., \psi_n$ denote the characters of $W_1, ..., W_n$ respectively. Note that these characters need not necessarily be distinct.

Let $\{\psi_k\}_i$ be a subset of all of the unique characters in $\psi_1, ..., \psi_n$. This must be the set of irreducible characters, therefore we know that χ decomposes as $\chi = \alpha_1\psi_{k_1} + ... + \alpha_n\psi_{k_n}$. However, by the previous lemma, we also have $\chi = \psi_1 + ... + \psi_n$. Therefore α_i must be exactly the multiplicity that the character ψ_{k_i} appears in $\psi_1, ..., \psi_m$. □

Proposition 2.10. *Let χ be the character of a representation (ρ, V) of group G. As is common in representation theory literature, let 1 denote the identity of G. Then*

$$\chi(1) = \dim(V)$$

The value of $\chi(1)$ is referred to as the degree of the χ.

Proof. By definition $\chi(1) = \mathrm{Tr}\,(\rho(1)) = \mathrm{Tr}\,(I_V) = \dim(V)$. □

Theorem 2.11. *Let $\chi_1, ..., \chi_n$ be the irreducible characters of G. Then*

$$|G| = \chi_1(1)^2 + ... + \chi_n(1)^2$$

Definition 2.9. The characters of degree 1 are referred to as the *linear characters*. They can equivalently be thought of as the homomorphisms from G into $\mathbb{C} \setminus \{0\}$.

Writing the information of the characters of a group into a table, gives us an extremely compact way write down the important information associated to a group.

2.4.2 Products of Characters

These results will be about the tensor product of representations. Just as in Chapter 1, giving a rigorous definition of the tensor product is probably outside the scope of this thesis. A good treatment of the tensor product can be found in chapter 10 of Dummit and Foote [2] and a treatment of what is relevant for representation theory can be found in chapter 19 of [3].

Definition 2.10. Let G be a group. Define a product on the space of class functions of G by pointwise multiplication. i.e. for all class functions χ, ψ of G,

$$\chi\psi(g) = \chi(g)\psi(g) \text{ for all } g \in G$$

This is sometimes referred to as the *tensor product* of characters.

If one uses purely character theoretic arguments, the tensor product is actually not needed. In fact, Steinberg's text [7] proves the major results in the representation theory of finite groups without invoking the tensor product. We take the same approach, however as we need to connect product characters to tensor products of quantum state spaces, we will have to take the following result as given. Proofs can be found in [3], chapter 19.

Proposition 2.12. *Let G be a group. Let V and W be representations of with characters χ and ψ respectively. Then the character of the representation $V \otimes W$ is the product character $\chi\psi$*

Theorem 2.13. *(Burnside-Brauer) Let χ be a character of a faithful representation of G, and suppose that $\chi(g)$ takes precisely r different values as g varies over all the elements of G. Then every irreducible character of G is a constituent of one of the powers $\chi^0, \chi^1, ..., \chi^{r-1}$*

Chapter 3

Symmetric Oracle Problems

In this chapter, we introduce a specific class of oracle problems in which the hidden information is sampled from a group G. We then summarize the results of a recent paper [1] of Pommersheim and Copeland that show the optimal query complexity of such problems is determined entirely by the character theory of G.

3.1 Symmetric Oracle Problems

This follows the approach taken in [1].

3.1.1 Classical learning

Let G be a finite group that acts on a finite set Ω. Denote the action of $g \in G$ on $x \in \Omega$ as $g \cdot x$. Suppose an element $g \in G$ is selected uniformly at random and hidden. A learner is given oracle access to the action of g as the function $f : \Omega \to \Omega$ given by $f(x) = g \cdot x$. That is, the learner can input elements of Ω as queries and see how g acts on them. The learner's goal is to determine g in as few queries as possible.

Example 3.1. Consider the example of the action of the symmetric group S_n on $\{1, ..., n\}$: Suppose a permutation σ is chosen uniformly at random from the set of permutations on $\{1, ..., n\}$ and hidden. We are given access to an oracle that takes an $x \in \{1, ..., n\}$ as input and returns $\sigma(x)$. How many queries do we need to determine σ?

This task requires $n - 1$ queries in the worst case. To see this, suppose σ were chosen to be the identity permutation. Even after seeing that σ fixes all but 2 elements of $\{1, ..., n\}$, we still cannot distinguish whether σ is the identity or the transposition of the last two elements.

This problem turns out historically to have been important to computational group theory and the efficient implementation of permutation groups in computer algebra systems. Storing groups in memory as permutations is very cumbersome and

so instead one expresses the group by its action on a *base*, which is defined as a sequence

$$B = [\beta_1, \beta_2, ...\beta_k]$$

of k distinct elements for which the only $g \in G$ that fixes every $\beta_i \in B$ is the identity. The size of the smallest base of a permutation group is exactly the worst-case number of queries that would be needed to identify a permutation in this group with oracle access to its action. A very good resource on this is the text [6] of Seress.

3.1.2 Quantum Learning

This problem naturally extends to the quantum setting. Let $\pi : G \to \mathrm{Aut}(\Omega)$ an action of G on a set Ω. Let V be a vector space with orthornormal basis Ω. Then the action $\pi : G \to \mathrm{Aut}(\Omega)$ defines a permutation of basis vectors and thus linearly extends to a unitary representation $\rho_\pi : G \to U(V)$. Here $U(V)$ denotes the group of unitary transformations of V.

This construction of a representation from a group action on a set is common in representation theory. It is common to refer to the character associated to this representation as the *permutation character* often denoted χ_π or just π. Since ρ_π assigns a permutation matrix to each g in G, the trace of $\rho_p i(g)$ for any $g \in G$ is exactly the number of points that g fixes under the action.

Definition 3.1. The permutation character χ_π of an action $\pi : G \to \mathrm{Aut}(\Omega)$, commonly just written π, is given by

$$\chi_\pi(g) := \mathrm{Fix}(g)$$

where $\mathrm{Fix}(g)$ is the number of elements in Ω fixed by the action of G.

3.1.3 Query complexity of quantum symmetric learning

Finding the classical query complexity of such a problem is relatively straightforward and there are many well-known algorithms for determining it. However, in the quantum case, it is less obvious how to determine the optimal number of queries.

A recent result of Pommersheim and Copeland [1] reduces this problem entirely to a problem of representation theory. We state theorem 4.2 from their paper

Theorem 3.1. *Suppose G is a finite group and $\pi : G \to U(V)$ is a unitary representation of G. Then an optimal t-query algorithm to solve symmetric oracle discrimination succeeds with probability*

$$P_{opt} = \frac{d_v}{|G|}$$

where

$$d_v = \sum_{\chi \in I(V^{\otimes t})} \chi(e)^2$$

where $I(V)$ denotes the set of irreducible characters of V.

Proof. For the proof of this fact, we refer to the paper itself. □

As mentioned before, the optimal number of queries to solve this problem classically for any permutation group is known as the *base size*. Since every permutation group also has an analogous learning problem in the quantum setting, we might call minimal number of queries needed for exact learning the *quantum base size* and the number of queries needed for learning with sufficiently high probability the *bounded quantum base size*. Using Theorem 3.1 we can define these terms in entirely in group theoretic language.

Definition 3.2. Let G be a finite group acting on a finite set Ω. Let π be the permutation character of this action. The *quantum base size of* G, denoted $\gamma(G)$, is the smallest t such that every irreducible character of appears in $\pi^{\otimes t}$.

The *bounded quantum base size of* G, denoted $\gamma_{\text{bdd}}(G)$, is the smallest t such that

$$\sum_{\chi \in I(V^{\otimes t})} \chi(e)^2 \geq \frac{2}{3}$$

Chapter 4

Constructions of the Quantum Base Size

This chapter gives explicit computations of the *quantum base size* defined in Chapter 3. We prove the number of queries required for exact and bounded error learning for the dihedral groups D_{4k}.

We work through the example of the dihedral groups in much detail to illustrate how one might go about computing these bounds for more families of groups.

4.1 Character Table of the Dihedral Group

The *dihedral group* of order $2n$ is given by

$$D_{2n} := \langle a, b \mid a^n = b^2 = 1,\ bab^{-1} = a^{-1} \rangle$$

It arises naturally as the planar symmetries of the n-gon. For this reason, elements of the form a^i are often referred to as rotations and elements of the form $a^i b$ are referred to as reflections.

The character table for the case when n is even is slightly different from the case when n is odd. To keep things as clear as possible and to avoid too much repetition, we will do everything only for the case where n is even, i.e. for the dihedral group of order $4k$ for some integer k. The results for the odd case readily follow from the exact same approach, only there are fewer conjugacy classes and thus fewer cases to check.

4.1.1 Irreducible Characters of D_{4k}

Linear Characters

The linear characters are precisely the homomorphisms $\lambda : D_{4k} \to \mathbb{C}^\times$. We can compute all such homomorphisms by looking at how they must act on the generators.

Let $\lambda : D_{4k} \to \mathbb{C}^\times$ be a homomorphism. Since $b^2 = 1$ we must have $\lambda(b)^2 = 1$, hence the only possible values for $\lambda(b)$ are 1 and -1. Since we have the relation

$abab = 1$, then $1 = \lambda(abab) = \lambda(a)\lambda(b)\lambda(a)\lambda(b) = \lambda(a)^2$. Thus $\lambda(a)$ must also be either 1 or -1. This leaves us with four possible choices, each of which is a linear character of D_{4k}

The following table gives the four linear characters of D_{4k}

g	1	a^k	a^i	b	ab
λ_1	1	1	1	1	1
λ_2	1	1	1	-1	-1
λ_3	1	$(-1)^k$	$(-1)^i$	1	-1
λ_4	1	$(-1)^k$	$(-1)^i$	-1	1

The Other Irreducible Characters

Consider the family of representations $\{\rho_j\}$ given by parameter $1 \leq j \leq k-1$ defined by

$$a \mapsto \begin{bmatrix} \zeta_{2z}^j & 0 \\ 0 & \zeta_{2k}^{-j} \end{bmatrix}, \qquad b \mapsto \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Let ψ_j denote the character of the representation ρ_j. Taking the trace gives us

$$\psi_j(a^n b^m) = \begin{cases} \zeta_n^j + \zeta_n^{-j}, & \text{if } b = 0 \\ 0, & \text{if } b = 1 \end{cases}$$

In terms of the conjugacy classes this is

g	1	a^k	a^i	b	ab
ψ_j	2	$2(-1)^j$	$\zeta_{2k}^{ij} + \zeta_{2k}^{-ij}$	0	0

In order to show ψ_j is irreducible we compute $\langle \psi_j, \psi_j \rangle$.

$$\begin{aligned} \langle \psi_j, \psi_j \rangle &= \frac{1}{|G|} \sum_{g \in G} \psi_j(g)\overline{\psi_j(g)} \\ &= \frac{1}{4k}\left[2^2 + \left(2(-1)^j\right)^2 + 2\sum_{i=1}^{k-1}\left(\zeta_{2k}^{ij} + \zeta_{2k}^{-ij}\right)\left(\overline{\zeta_{2k}^{ij} + \zeta_{2k}^{-ij}}\right)\right] \\ &= \frac{1}{4k}\left[4 + 4 + 2\sum_{i=1}^{k-1}\left(\zeta_{2k}^{ij} + \zeta_{2k}^{-ij}\right)^2\right] \\ &= \frac{1}{4k}(8 + 2(4)(k-1)) \\ &= 1 \end{aligned}$$

Thus by Theorem 2.7, ψ_j is an irreducible character.

We have found 4 irreducible linear characters and $k-1$ irreducible characters of degree 2. There are $k+3$ conjugacy classes and so this must be a complete set of characters. Hence the character table of D_{4k} is

g	1	a^k	a^i	b	ab
λ_1	1	1	1	1	1
λ_2	1	1	1	-1	-1
λ_3	1	$(-1)^k$	$(-1)^i$	1	-1
λ_4	1	$(-1)^k$	$(-1)^i$	-1	1
ψ_j	2	$2(-1)^j$	$\zeta_{2k}^{ij} + \zeta_{2k}^{-ij}$	0	0

4.2 Quantum Base Size of the Dihedral Group

4.2.1 Permutation Character

Let $G = D_{4k}$. Consider the natural action of D_{4k} on the vertices of the $2k$-gon. We want to compute $\mathrm{Fix}(g)$ for each conjugacy class.

Every rotation (excluding the identity) will fix no points. Similarly an odd number of rotations and then a flip will never yield any fixed points. The only elements that will fix anything will be the identity (which will fix all $2k$ vertices) and the primitive reflections, which will each fix exactly the two vertices that lie in the axis of reflection. Thus the permutation character $\pi(g) = \mathrm{fix}(g)$ is given by

g	1	a^k	a^i	b	ab
$\pi(g)$	$2k$	0	0	2	0

for all $1 \leq i \leq k-1$.

4.2.2 Queries for Exact Learning

We want to determine $\gamma(D_{4k})$. In order to do this, we look at the irreducible constituents of the tensor powers of the permutation character. We need to find the smallest integer t such that π^t, the tth tensor power of pi, contains every irreducible character as a constituent. That is, the smallest t such that $\langle \pi^t, \chi \rangle \neq 0$ for all irreducible characters χ.

We summarize the information we have computed so far about the characters of D_{4k}

g	1	a^k	a^i	b	ab
$\lvert \mathrm{cl}(g) \rvert$	1	1	2	k	k
λ_1	1	1	1	1	1
λ_2	1	1	1	-1	-1
λ_3	1	$(-1)^k$	$(-1)^i$	1	-1
λ_4	1	$(-1)^k$	$(-1)^i$	-1	1
ψ_j	2	$2(-1)^j$	$\zeta_{2k}^{ij} + \zeta_{2k}^{-ij}$	0	0
π	$2k$	0	0	2	0

where ψ_j is a family of representations, one for each $1 \leq j \leq k-1$ and a^i is a unique conjugacy classes for each $1 \leq i \leq k-1$.

To determine which irreducible characters are constituents of π, we compute $\langle \pi, \chi \rangle$ for each irreducible χ. Since π is nonzero only on the classes 1 and b we omit these terms.

$$\langle \pi, \lambda_1 \rangle = \frac{1}{|G|}\Big((1)(1)(2k) + (2)(1)(k)\Big) = \frac{1}{4k}(2k + 2k) = 1$$
$$\langle \pi, \lambda_2 \rangle = \frac{1}{|G|}\Big((1)(1)(2k) + (2)(-1)(k)\Big) = \frac{1}{4k}(2k - 2k) = 0$$
$$\langle \pi, \lambda_3 \rangle = \frac{1}{|G|}\Big((1)(1)(2k) + (2)(1)(k)\Big) = \frac{1}{4k}(2k - 2k) = 1$$
$$\langle \pi, \lambda_4 \rangle = \frac{1}{|G|}\Big((1)(1)(2k) + (2)(-1)(k)\Big) = \frac{1}{4k}(2k - 2k) = 0$$
$$\langle \pi, \psi_j \rangle = \frac{1}{2n}(2n) = 1$$

Thus, the decomposition of π into irreducibles is given by

$$\pi = \lambda_1 + \lambda_3 + \sum_{j=1}^{k-1} \psi_j$$

Hence, the irreducibles λ_2 and λ_4 are not constituents of π and so $\gamma(D_{4k})$ must be greater than one. By Theorem 2.8, every irreducible must appear in the decomposition of π^t for some $t \leq 2$, thus we know without any further computation that $\gamma(D_{4k}) = 2$.

4.2.3 Queries for bounded probability of learning

Although this means a quantum oracle does not help you solve the problem exactly with fewer queries, it is better when you allow for finding a solution with high probability.

As given in chapter 3, the number of queries needed to succeed with probability greater than $2/3$ is the smallest positive integer t such that

$$\frac{1}{|G|} \sum_{\chi \in N^{\otimes t}} \chi(e)^2 \;\geq\; \frac{2}{3}$$

We can find this value using what we computed in section 4.2.2. The constituent irreducibles of π are λ_1, λ_3, and ψ_j for each $1 \leq j \leq k-1$. These are 2 characters of degree 1 and $k-1$ characters of degree 2. Thus

$$\frac{1}{|D_{4k}|} \sum_{\chi \in N^{\otimes t}} \chi(e)^2 = \frac{1}{4k}\big(2(1^2) + (k-1)(2^2)\big) = \frac{1}{4k}(2 + 4k - 4) = 1 - \frac{1}{2k}$$

Therefore one can succeed using only one query with probability $1 - \frac{1}{2k}$, which is bounded above $2/3$ for every dihedral group of order $4k$.

References

[1] Daniel Copeland and James Pommersheim. Quantum query complexity of symmetric oracle problems. 12 2018.

[2] D.S. Dummit and R.M. Foote. *Abstract Algebra*. Wiley, 2004.

[3] G. James and M. Liebeck. *Representations and Characters of Groups*. Cambridge University Press, 2001.

[4] Andy Matuschak and Michael A. Nielsen. Quantum computing for the very curious. `https://quantum.country/qcvc`, 2019.

[5] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, New York, NY, USA, 10th edition, 2011.

[6] Ákos Seress. *Permutation Group Algorithms*. Cambridge Tracts in Mathematics. Cambridge University Press, 2003.

[7] B. Steinberg. *Representation Theory of Finite Groups: An Introductory Approach*. Universitext. Springer New York, 2011.