Two Cases of the Symmetric Oracle Discrimination Problem: Representation Theory in Quantum Computing

A Thesis
Presented to
The Established Interdisciplinary Committee for Mathematical and Natural Sciences
Reed College

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

Dale D. Schandelmeier-Lynch

May 2025

Approved for the Committee
(Mathematics and Computer Science)

Jamie Pommershiem

Greg Anderson

Acknowledgements

I greatly appreciate Zajj Daugherty's help with the decomposition of $\mathbb{C}\Omega_k$ via Young's rule and also for several enlightening discussions. Additionally, some of Zajj's ideas are offered in the discussion section as potential clues for future proofs in this area.

I would also like to thank Jamie and Greg for being helpful advisors and for keeping me on track to finish my thesis on time.

One of the papers I cite often by Cohen De Valle was introduced to me by Daniel Copeland via Jamie; that input is what gave my thesis a solid direction to work in.

Table of Contents

Abstract

The preface pretty much says it all.

Introduction

Theoretical quantum computing often employs a model of asymptotic analysis known as *oracle problems* as an abstraction to make comparisons between classical and quantum computers easier. An oracle is a theoretical function whose inner workings cannot be observed or manipulated; algorithms are only allowed to choose an input to the oracle and receive its output. The goal when given an oracle problem is to create an algorithm which learns a given characteristic of the function hidden by the oracle. We usually assume that the function hidden by the oracle is sampled randomly from a distribution known to the algorithm. A simple example is as follows. For a function $f : \{a_0, a_1\} \to \mathbb{Z}/2\mathbb{Z}$ uniformly sampled from the space of possible functions, determine the mod 2 sum $a_0 \oplus a_1$. Here we define the oracle O to offer information to the algorithm through the black-box function $O(x) \coloneqq f(x)$. A classical algorithm would have to make 2 queries to O for exact learning, as it must query every element of the codomain to learn the values of a_0 and a_1. As it turns out, a well-known quantum algorithm called Deutsch's algorithm can learn the value of $a_0 \oplus a_1$ with certainty after just 1 query!

maybe this shouldn't be called a model

Also of interest in oracle problems is *bounded error* learning, where we consider how many queries are needed until an algorithm has some $> \frac{1}{2}$ chance of guessing the characteristic correctly. Contrast the previous example with a problem where the algorithm has to learn f exactly; before making any queries the classical algorithm has a $\frac{1}{2}$ chance of guessing $a_0 \oplus a_1$ correctly, and this doesn't change after making a query. On the other hand, learning the exact function has a $\frac{1}{4}$ chance of success with 0 queries, and a $\frac{1}{2}$ chance of success with 1 query. If we considered an extreme version of this problem, $f : \{a_0, a_1\} \to \mathbb{Z}/1000\mathbb{Z}$, then we observe that a classical algorithm has a near-zero chance of guessing the function before it makes its final query. For most oracle problems, this pattern holds true in the classical case; the algorithm knows nearly nothing about the function until it knows the whole function. Effectively, bounded error learning is equivalent to exact learning classically.

Shockingly, quantum algorithms have a different tendency. The function will be maybe this asymptotic claim is false

nearly unknown for the first few queries, but after an asymptotically smaller number of queries than those needed for exact learning, nearly the whole function will be known! We are thus concerned with comparing the exact classical query complexity with both the exact and bounded error quantum query complexity of a given oracle problem. Of note is that if we find a proof using our methods, we have found a necessary and sufficient number of queries to solve the problem; we prove the existence of an algorithm which solves the problem with the given asymptotics, and it is optimal, meaning no algorithm makes asymptotically fewer queries.

Our oracle problem comes from Copeland and Pommersheim 2021, as a specification of the more general coset identification problem: Suppose a group G acts by permutations on a finite set Ω (we call Ω a G-set). An algorithm is given access to an oracle which takes an element $\omega \in \Omega$ and returns $a \cdot \omega$ for some hidden group element (uniformly sampled) $a \in G$. The goal is to guess the hidden element $a \in G$. We will be concerned with the groups S_n and $A_n \leq S_n$ acting on the set of k element subsets of n, and on the set of regular partitions of n into b parts of size a.

To answer these problems we will use a crucial result from Copeland and Pommersheim 2021 which enables us to determine optimal quantum query complexity with a correspondence. Taking the tensor product of a representation derived from our G-set Ω with itself repeatedly will be equivalent to making oracle queries, turning our quantum computing problem into one of representation theory. By understanding the representation corresponding to Ω and the way it tensors with itself, we will be able to find the query complexity. Because our G-sets have S_n and A_n as groups, we will be deeply concerned with the representation theory of S_n; combinatorial structures known as *Young diagrams* and *Young tableaux* will offer a useful correspondence through which we can understand S_n's representations.

Chapter 1

Background

We first introduce the necessary quantum computing and representation theory background to understand the problem statement in context.

1.1 Quantum Computing

Quantum computing utilizes the nonclassical nature of quantum mechanics to achieve asymptotic improvements over classical algorithms. Not all problems admit a quantum "speedup"; we will be more specific about our computational framework after the necessary background. To illuminate the physical-mathematical framework in which quantum computing takes place, we will expound the postulates of quantum mechanics and the associated linear algebra from Nielsen and Chuang 2010, Chapter 2.

1.1.1 Quantum Mechanics

Due to theorem 4.2 from Copeland and Pommersheim 2021, it turns out that our results rely almost entirely on representation theory; nevertheless we introduce the four fundamental postulates from Nielsen and Chuang 2010, Chapter 2 to provide context for the quantum model of computation. Throughout this section we use the "bra-ket" (or Dirac) notation to represent vectors in vector spaces. This is inherited from physics and uses the symbols $|\cdot\rangle$ called ket, $\langle\cdot|$ called bra, and $\langle\cdot|\cdot\rangle$ called braket (as we have "bracketed" the symbols);

$$|\psi\rangle$$

is a vector we have labeled ψ. In quantum computing we are concerned with complex vector spaces by postulate; we need to define inner products both for this postulate and to elaborate on bra-ket notation.

Definition 1.1.1 (Inner product, Nielsen and Chuang 2010, p. 2.1.4)**.** A function $(\cdot,\cdot) : V \times V \to \mathbb{C}$ is a (complex) inner product if it satisfies the following requirements:

1. $(\cdot,\cdot)$ is linear in the second argument:

$$\left(|v\rangle, \sum_i \lambda_i |w_i\rangle \right) = \sum_i \lambda_i \left(|v\rangle, |w_i\rangle \right).$$

2. $(|v\rangle, |w\rangle) = (|v\rangle, |w\rangle)^*$.

3. $(|v\rangle, |v\rangle) \geq 0$ with equality if and only if $|v\rangle = 0$.

The intuition of braket notation is that $\langle\phi|$ is the dual of the vector $|\phi\rangle$, which in the corresponding matrix presentation of vectors corresponds to turning a column vector into a row vector. Therefore taking the standard matrix product

$$\langle\phi| \, |\psi\rangle$$

is equivalent to the standard inner product $(|\phi\rangle, |\psi\rangle) = \sum_{k=1}^n \phi_k^* \psi_k$, which we notate as

$$\langle\phi|\psi\rangle ;$$

here the symbol $\cdot^*$ denotes the unary complex conjugation operation. The elegance of braket notation, apart from saving space, is that we can easily tell the output of complicated linear expressions; it's easy to see that $\langle\phi|\psi\rangle \langle x|y\rangle \in \mathbb{F}$, while it takes a little thinking to see that

$$\begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

equals a scalar.

We are now ready to introduce our first postulate. As mathematicians we needn't worry about how physicists experimentally constructed these postulates, although their (approximate) truth will be necessary if we are to physically realize quantum computers and employ the algorithms we design. If we take these postulates at face value, phenomena that are strange from a classical physics standpoint feel like natural consequences of the mathematics; in that sense it can be helpful to abstract away your physical intuition and instead focus on the consequences of these postulates when we start doing computation.

Postulate 1.1.1 (Nielsen and Chuang 2010, p. 2.2.1)**.** Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the *state space* of the system. The system is completely described by its *state vector*, which is a unit vector in the system's state space.

Here unit vector refers to a vector with norm 1; the standard norm we employ is $|| \, |\psi\rangle \, || := \sqrt{\langle\psi|\psi\rangle}$. Note that quantum mechanics doesn't tell us the state space or state vector of a particular system; it only describes in general the characteristics systems share. In our quantum computing model we regard these physical systems as memory and perform operations on this memory through the appropriate mathematical morphism, which in this case are linear operators. A minor detail is that Hilbert spaces have more properties when they are infinite-dimensional, but in our case we will only ever use finite memory and so our Hilbert spaces are always finite-dimensional.

The simplest quantum mechanical system is the *qubit*. A qubit resides in $\mathbb{C}^2$ and so has a two-dimensional state space. We use $|0\rangle$ and $|1\rangle$ to label the elements of an orthonormal basis for the qubit; in practice we let

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

Note that $|0\rangle$ is not the zero vector 0; we use $|0\rangle$ despite this overlap in reference to the two states of a bit, 0 and 1. What makes the qubit interesting in contrast is that we are not limited to state vectors $|0\rangle$ and $|1\rangle$; we know from linear algebra that we can express any element of the state basis as

$$|\psi\rangle = a\,|0\rangle + b\,|1\rangle\,,$$

where a and b are in the complex numbers. Even with the condition that $|\psi\rangle$ be a unit vector, the state of the qubit is described by four real numbers (that is $x + iy$ where $x, y \in \mathbb{R}$ for both a and b) instead of one bit! In practice we can only physically measure 3 of these values, but nevertheless a qubit can take on far more values than a bit because it can exist in a superposition of the states $|0\rangle$ and $|1\rangle$; that is both a and b can be nonzero at the same time. In our case we will be using the state space $\mathbb{C}\Omega$ with standard basis $\{|\omega\rangle \,|\, \omega \in \Omega\}$; you can imagine we have defined an abelian group structure on Ω and so we are considering the group algebra $\mathbb{C}\Omega$ by linearization.

We are now ready for our next postulate:

Postulate 1.1.2 (Nielsen and Chuang 2010, p. 2.2.2)**.** The evolution of a *closed* quantum system is described by a *unitary transformation*. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi^{'}\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2;

$$|\psi^{'}\rangle = U\,|\psi\rangle\,.$$

With operators we can define the last piece of the bra-ket puzzle. If we consider the vector $|v\rangle$ from a Hilbert space V, and the vector $|w\rangle$ from a Hilbert space W, then we define the outer product notation for a linear operator $|w\rangle\langle v| : V \to W$ as

$$(|w\rangle\langle v|)(|v^{'}\rangle) \coloneqq |w\rangle\langle v|v^{'}\rangle = \langle v|v^{'}\rangle\,|w\rangle\,.$$

We can thus interpret the notation $|w\rangle\langle v|v^{'}\rangle$ in two ways: as a linear operator fed $|v^{'}\rangle$, and as the vector $|w\rangle$ scaled by the inner product of $\langle v|v^{'}\rangle$. In terms of matrix multiplication we see here the associativity property at work; the product of the column and row vectors $|w\rangle\langle v|$ creates a matrix which then acts as a linear operator on $|v^{'}\rangle$, or the product of row and column vectors creates a scalar which then scales $|w\rangle$.

To define a unitary operator, we first need the *adjoint* or *Hermitian conjugate* (Nielsen and Chuang 2010, p. 2.1.6) of an operator. This is the unique operator $A^\dagger$ such that for all vectors $|v_1\rangle, |v_2\rangle \in V$,

$$(|v_1\rangle\,, A\,|v_2\rangle) = (A^\dagger\,|v_1\rangle\,, |v_2\rangle).$$

Here the operation is the inner product; by convention we define $(|v\rangle)^\dagger \coloneqq \langle v|$ so that $(A\,|v_1\rangle)^\dagger\,|v_2\rangle = \langle v_1|A^\dagger|v_2\rangle$. When we have a matrix representing an operator A, the standard definition of the adjoint is to take the conjugate and then the transpose of A: $A^\dagger \coloneqq (A^*)^T$. When $A^\dagger = A$, we call A *Hermitian* or *self-adjoint*. As a broader class we say that an operator A is *normal* if $AA^\dagger = A^\dagger A$; it is immediate that Hermitian operators are normal. Finally, we say an operator U is unitary if $U^\dagger U = I$, where I is the identity operator (or matrix); for finite vector spaces it can be proven that this implies $UU^\dagger = I$, so unitary operators are also normal. We care about unitary products because they preserve inner products; observe that

$$(U\,|v\rangle\,, U\,|w\rangle) = \langle v|U^\dagger U|w\rangle = \langle v|I|w\rangle = \langle v|w\rangle \quad \text{(Nielsen and Chuang 2010, p. 2.36).}$$

Postulate 2 demands that operators be unitary so that the unit condition of the state vector is maintained.

We are now ready for postulate 3, which tells us what happens when an isolated quantum system is measured. To measure a quantum system it must be exposed to an external physical system, and so the rule of unitary evolution from postulate 2 no longer necessarily applies.

Postulate 1.1.3 (Nielsen and Chuang 2010, p. 2.2.3)**.** Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators*. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcomes that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement then the probability that result m occurs is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle \,,$$

and the state of the system after the measurement is

$$\frac{M_m \left|\psi\right\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}.$$

The measurement operators satisfy the *completeness equation*,

$$\sum_m M_m^\dagger M_m = I.$$

The completeness equation expresses the fact that probabilities sum to one:

$$1 = \sum_m p(m) = \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle \,.$$

The takeaways from this postulate are twofold. Measurements in a quantum system can change the state of the state vector; information can be lost by performing a measurement. From the real numbers contained in a state vector — three in the case of the qubit — we can only observe a finite number of states. In fact, if we are trying to distinguish between states that are not orthonormal, then we lose the guarantee of distinguishability. Because some state $|\psi\rangle$ is not orthonormal to some other distinct state $|\phi\rangle$, there is a component of $|\psi\rangle$ that is orthogonal to $|\phi\rangle$ and a component that is parallel to $|\phi\rangle$; therefore when you measure a system with state vector $|\psi\rangle$ there is a nonzero chance of it being misidentified as $|\phi\rangle$. This means that we can only perfectly distinguish between a number of states up to the dimension of the state space. Furthermore, cascaded measurements $\{L_l\}$ and $\{M_m\}$ are equivalent to a single measurement with operators $\{N_{lm}\}$ where $N_{lm} := M_m L_l$ (Nielsen and Chuang 2010, p. 2.57). If we don't care about the evolution of the system over time, there is no reason to not take all the measurements we want to at once.

This leads us to the concept of the POVM, or "positive operator-valued measure." If we perform a measurement on a state vector $|\psi\rangle$ with measurement operators M_m, then we know the probability of outcome m is given by $p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle$. Now if we define

$$E_m := M_m^\dagger M_m,$$

we can conclude that $\{E_m\}$ is a set of positive operators such that $\sum_m E_m = I$ and $p(m) = \langle\psi|E_m|\psi\rangle$. Therefore this set is sufficient to determine the probability of each measurement outcome if we don't care about the post-measurement state, according to Nielsen and Chuang 2010, p. 2.2.6. In fact, we can take this the other direction and define a POVM to be any set of operators such that each operator is positive and the completeness relation $\sum_m E_m = I$ is obeyed. This is legitimate because if we define measurements $M_m := \sqrt{E_m}$, we see that $\sum_m M_m^\dagger M_m = \sum_m E_m = I$, so we have constructed a set of measurements $\{M_m\}$ with POVM $\{E_m\}$. We prefer POVMs to measures as a convenience, because when we do computation we take one measurement at the end and discard the state vector.

Our last postulate tells us how to describe quantum systems made out of multiple distinct systems; this is useful in computation because we model adding memory as composing more physical systems.

Postulate 1.1.4 (Nielsen and Chuang 2010, p. 2.2.8)**.** The state space of a composite physical system is the tensor product of the state spaces of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is in the state $|\psi_i\rangle$, then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.

A tensor product is an operation on vector spaces analogous to "multiplying" them; in the same way that a Cartesian product is an operation on sets that changes what the elements of the resulting set look like (turns them into tuples), the tensor product of the spaces will change the vectors into "tensor products" in a different sense.

The following definitions are from Nielsen and Chuang 2010, p. 2.1.7. Suppose we have Hilbert spaces V and W — although tensor products can be defined on plain vector spaces — of dimension m and n respectively. Then $V \otimes W$ is an mn-dimensional vector space. As an analogue for finite sets $X : |X| = m$ and $Y : |Y| = n$ the Cartesian product has cardinality $|X \times Y| = mn$. The elements of this vector space are linear combinations of tensor products $|v\rangle \otimes |w\rangle$ where $|v\rangle \in V$ and $|w\rangle \in W$. The tensor product is akin to a tuple in that elements from the first vector space go in the first index, elements from the second vector space go in the second index, et cetera. If $\{|i\rangle_n\}$ and $\{|j\rangle_n\}$ are (preferably but not necessarily orthonormal) bases for the spaces V and W then $\{\{|i\rangle \otimes |j\rangle\}_{n,m}\}$ is a basis for $V \otimes W$. We abbreviate tensor products as $|v\rangle |w\rangle$ or $|vw\rangle$ for convenience. For example, if V is a qubit then $V \otimes V$, also notated as $V^{\otimes 2}$ to indicate it has been tensored with itself (akin to squaring) contains the element $|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle = |00\rangle + |11\rangle$. We define the tensor product at the element (vector) level to have the following properties:

1. For an arbitrary scalar z and vectors $|v\rangle \in V$ and $|w\rangle \in W$,

$$z(|v\rangle \otimes |w\rangle) = (z\,|v\rangle) \otimes |w\rangle = |v\rangle \otimes (z\,|w\rangle).$$

2. For an arbitrary $|v_1\rangle, |v_2\rangle \in V$ and $|w\rangle \in W$,

$$(|v_1\rangle + |v_2\rangle) \otimes |w\rangle = |v_1\rangle \otimes |w\rangle + |v_2\rangle \otimes |w\rangle .$$

3. For an arbitrary $|v\rangle \in V$ and $|w_1\rangle, |w_2\rangle \in W$,

$$|v\rangle \otimes (|w_1\rangle + |w_2\rangle) = |v\rangle \otimes |w_1\rangle + |v\rangle \otimes |w_2\rangle .$$

These distributivity laws intuitively make tensor products function like multiplication between vectors. Furthermore, we can define linear operators on tensor products which inherit from linear operators on the tensored spaces; let $|v\rangle \in V, |w\rangle \in W$, and let A and B be linear operators on V and W. Then we define the linear operator $A \otimes B$ in $V \otimes W$ by the equation

$$(A \otimes B)(|v\rangle \otimes |w\rangle) \coloneqq A\,|v\rangle \otimes B\,|w\rangle .$$

This definition of $A \otimes B$ now naturally extends linearly to all the elements of $V \otimes W$; that is

$$(A \otimes B)(\sum_i a_i\,|v_i\rangle \otimes |w_i\rangle) \coloneqq \sum_i a_i A\,|v_i\rangle \otimes B\,|w_i\rangle .$$

It can be shown that $A \otimes B$ thus defined is a well-defined linear operator. Finally, a natural inner product can be inherited from the tensored spaces; we define this product as

$$\left(\sum_i a_i \left| v_i \right\rangle \otimes \left| w_i \right\rangle , \sum_j b_j \left| v_j^{'} \right\rangle \otimes \left| w_j^{'} \right\rangle \right) := \sum_{i,j} a_i^* b_j \left\langle v_i | v_j^{'} \right\rangle \left\langle w_i | w_j^{'} \right\rangle .$$

This function can also be shown to be a well-defined inner product. Equipped with this inner product $V \otimes W$ is now an inner product/Hilbert space, and it inherits our previously established structure of an adjoint, unitarity, normality, Hermiticity, et cetera.

1.1.2 Our Computational Model

We have now covered all the postulates of quantum mechanics! The standard model of quantum computation is simple: we start with a state space $(\mathbb{C}^2)^{\otimes n} \otimes (\mathbb{C}^2)^{\otimes a}$ which corresponds to an input requiring n qubits to store and an extra workspace of a ancilla qubits — usually of constant size — needed as extra space for certain computations. We can assume that the memory begins in any desired state $|\psi\rangle$ as such a state can be initialized by applying a constant number of unitary operators. Unitary operators are then applied to do some kind of computation, and a POVM is taken to measure the output. Instead of worrying about the time or space complexity of this model, we will be concerned with query complexity; this simplifies complexity to the question of how many times a given black box function (the "oracle") is applied as a unitary operator. Because this model is non-uniform — the operators applied change as the input size changes — we describe a quantum algorithm as a recipe to construct the necessary sequence of operations based on the input size. This model lends itself to representation as a circuit diagram, where reading from left to right we initialize qubits as separate inputs, apply unitary operations to qubits, and then perform a measurement; an example is depicted in figure 1.1.

is this true? maybe only takes one?

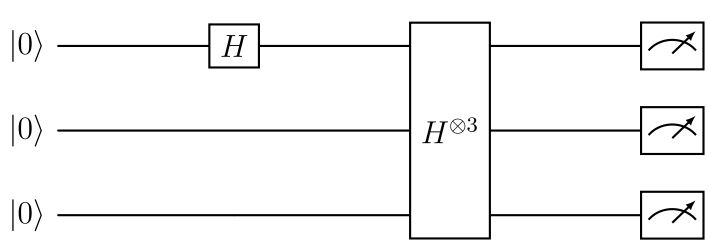


Figure 1.1: The state $|000\rangle \in (\mathbb{C}^2)^{\otimes n}$ gets passed through the unitary operators $H \otimes I \otimes I$ and $H^{\otimes 3}$, then is measured by a POVM.

The model we employ in our research is similar, but instead uses the state space $\mathbb{C}\Omega \otimes \mathbb{C}^N$; here we are taking the group ring $\mathbb{C}\Omega$, where Ω is a G-set with free abelian group structure, as a single register. You can imagine it as a $|\Omega|$-dit because $\mathbb{C}\Omega$ has $|\omega\rangle \in \Omega$ as a basis, but endowed with extra structure because the action of G on Ω naturally corresponds to a group of unitary operators on $\mathbb{C}\Omega$. The space $\mathbb{C}^N$ is simply the ancilla register but regarded as one N-dit rather than multiple qubits. Note that the suffix "-dit" here indicates a register of a given dimension (d for dimension), whereas qubit refers to a register of dimension 2 (as "bi-" means 2). To understand why we use this model, we will need an introduction to representation theory.

1.2 Introductory Representation Theory

Our results and model employ a field of abstract algebra known as representation theory, which is concerned with correspondences between groups and linear operators. By turning group elements into (unitary) linear operators, we will be able to use them as gates in our quantum computing model. Our introduction here will be focused on laying the groundwork for the representation theory of S_n, the subject in representation theory relevant to our results.

Our first definition will be specific to the set of invertible $d \times d$ matrices with entries in $\mathbb{C}$. This is the full complex matrix algebra: a vector space of matrices over $\mathbb{C}$ whose multiplication is given by ordinary matrix multiplication.

Definition 1.2.1 (Sagan 2001, Definition 1.2.1)**.** A *representation* (more specifically a *matrix representation*) of a group G is a group homomorphism

$$\rho : G \to \mathrm{GL}_d$$

where GL_d is the complex general linear group of degree d; that is, the set of invertible $d \times d$ matrices from $\mathbb{C}^d$ to $\mathbb{C}^d$.

To break down the definition of a group homomorphism, this means that

$$\rho(e) = I_d$$

where e is the identity of the group, and I_d is the identity matrix of degree d in $\mathbb{C}$, and

$$\rho(g)\rho(h) = \rho(gh) \text{ for all } g, h \in G.$$

This means that multiplying group elements and then taking their matrix representation is equivalent to taking their matrix representations and performing matrix multiplication on them.

Note that representations can be defined for other fields like $\mathbb{R}$ or $\mathbb{Z}/P\mathbb{Z}$, and doing so changes the core theorems of the theory; we won't be concerned with this because our quantum mechanical postulates demand we use $\mathbb{C}$. A point of confusion with representations is that they are defined to be *homomorphisms*, not *monomorphisms*. We might intuitively imagine that a matrix representation is equivalent to the group just with matrices instead of group elements, but group elements (and hence group structure) can be lost by mapping nontrivial elements of the group to the identity matrix in the linear group (that is, a representation can have nontrivial kernel). Furthermore, additional representations can be created by choosing a different degree or performing a change of basis on a given representation. A small degree can possibly restrict the possible representations to exclude monomorphisms. On the other hand, a change of basis shouldn't fundamentally change the linear transformation a matrix represents. We can consider representations ρ, ϕ equivalent (through an equivalence relation) if there exists an invertible matrix T such that $\rho(g) = T^{-1}\phi(g)T$ for all $g \in G$.

This relationship between matrices and linear transformations can be further used to avoid a choice of basis; we now introduce G-modules.

Definition 1.2.2 (Sagan 2001, Definition 1.3.1)**.** Let V be a vector space and G be a group. Then V is a *G-module* if there is a group homomorphism

$$\rho : G \to \mathrm{GL}(V),$$

where $\mathrm{GL}(V)$ is the set of general linear transformations of the vector space V. Equivalently, V is a G-module if there is a multiplication $g \cdot v$ such that

1. $g \cdot v \in V$,
2. $g(cv + dw) = c(gv) + d(gw)$,
3. $(gh)v = g(hv)$, and
4. $ev = v$

for all $g, h \in G$, $v, w \in V$, and scalars $c, d \in \mathbb{C}$. That is, there is a left action $G \circlearrowleft V$ on V that distributes over addition.

These two definitions are equivalent because acting by a group element g is equivalent to applying the linear transformation $\rho(g)$. G-modules and matrix representations are closely related. We use any matrix representation X with degree matching the dimension of V to construct a G-module by defining the linear action $g \cdot v := X(g)v$. Likewise, if we have a G-module we can define a matrix representation X by choosing a basis B for the linear transformations $\rho(g)$ and then construct $X(g)$. An important difference between representations and modules is that we have a choice of vector space when making a module. We can use this fact (and the fact that actions are part of our module definition) to naturally represent arbitrary group actions.

Example 1.2.3. We turn a group action $G \circlearrowright X$ into a G-module by first making the vectors of our space be formal linear combinations of elements of the set: $[x] = \sum_{x_i \in X} \lambda_i x_i$. The group action then can be linearly extended to act on $\mathbb{C}[X]$, the vector space generated by X over $\mathbb{C}$:

$$\begin{gathered} G \circlearrowright \mathbb{C}[X] \\ (g, [x]) \mapsto \sum_{x_i \in X} \lambda_i (g \cdot x_i) \end{gathered}$$

Now we can define a G-module

$$\rho : G \to \mathrm{GL}(|X|, \mathbb{C}[X])$$

where $\rho(g)$ is the linear transformation induced by the action of G on X; if we take the standard basis of $\mathbb{C}[X]$ to be the set X, then $\rho(g)$ can be constructed as a matrix which permutes each x_i according to the image of the action of g on each x_i.

We will be concerned with finding *irreducible* representations/modules, which are fundamental to understanding the representation theory of a group because for representations over $\mathbb{C}$ and similarly nice fields, any representation can be represented as a product of irreducible representations. We will concern ourselves with irreducible modules primarily because they are easier to work with and are in close correspondence with irreducible representations. To do so we first need submodules.

Definition 1.2.4 (Sagan 2001, Definition 1.4.1)**.** Let V be a G-module. A *submodule* of V is a subspace W that is closed under the action of G; that is,

$$w \in W \implies gw \in W \text{ for all } g \in G.$$

Example 1.2.5. Any G-module V has the submodules $W = V$ as well as $W = \{0\}$, where 0 is the zero vector. These two submodules are called trivial, and any other submodules are called nontrivial.

Definition 1.2.6 (Sagan 2001, Definition 1.4.1)**.** A nonzero G-module V is *reducible* if it contains a non-trivial submodule W. Otherwise V is said to be irreducible.

We will need the following definition in our analysis of S_n to demonstrate that we have found a complete set of irreducible non-isomorphic S_n-modules.

Definition 1.2.7 (Sagan 2001, Definition 1.6.1)**.** Let V and W be G-modules. Then a *G-homomorphism* is a linear transformation $\Theta : V \to W$ such that

$$\Theta(g \cdot v) = g \cdot \Theta(v)$$

for all $g \in G$ and $v \in V$.
A *G-isomorphism* is a bijective G-homomorphism.

Finally, we introduce a main theorem that we will need for the representation theory of S_n. This theorem tells us that any representation can be constructed out of irreducible representations. To understand it we first need the notion of a direct sum.

Definition 1.2.8 (Sagan 2001, Definition 1.5.1)**.** Let V be a vector space with subspaces U and W. Then V is the *(internal) direct sum of U and W*, written $V = U \oplus W$, if every $v \in V$ can be written uniquely as a sum

$$v = u + w, \quad u \in U, w \in W.$$

In the same way that the tensor product is like multiplication for vector spaces, the direct sum is analogous to the addition of vector spaces.

Theorem 1.2.9 (Sagan 2001, Theorem 1.5.3; Maschke's theorem)**.** *Let G be a finite group and let V be a nonzero G-module. Then*

$$V = W_1 \oplus W_2 \oplus \cdots \oplus W_k$$

where each W_i is an irreducible G-submodule of V.

We tack on one extra definition for the tensor product of representations; while we won't need it for the representation theory of the symmetric group, we will use it in our problem statement and our results.

Definition 1.2.10 (Sagan 2001, Definition 1.11.1)**.** Let G and H have matrix representations X and Y, respectively. The tensor product representation, $X \otimes Y$, assigns to each $(g, h) \in G \times H$ the matrix

$$(X \otimes Y)(g, h) = X(g) \otimes Y(h).$$

1.2.1 Character Theory

It turns out that much of the information of a representation can be obtained from the trace of its corresponding matrices. This is shocking and useful because the information of a matrix with n^2 entries can be obtained from the sum of just n entries. We now define the character, which stores all the traces of a representation.

Definition 1.2.11 (Sagan 2001, Definition 1.8.1)**.** Let $X(g)$ where $g \in G$ be a matrix representation. Then the *character* of X is

$$\chi(g) = \mathrm{tr}(X(g)),$$

where tr denotes the trace of a matrix. In other words χ is the function

$$\chi : G \to \mathbb{C}$$

where

$$g \mapsto \mathrm{tr}(X(g)).$$

If V is a G-module, then its character is the character of a matrix representation X corresponding to V. If matrices X and Y both correspond to V, then we know $Y = TXT^{-1}$ for some T. Thus, for all $g \in G$,

$$\mathrm{tr}(Y(g)) = \mathrm{tr}(TX(g)T^{-1}) = \mathrm{tr}(X(g)),$$

since trace is invariant under conjugation. So X and Y have the same character and our notion of module character is well-defined.

The following proposition describes the basic properties of a character:

Proposition 1.2.12 (Sagan 2001, Proposition 1.8.5)**.** *Let X be a matrix representation of a group G of degree d with character χ.*

1. $\chi(\epsilon) = d$.

2. *If K is a conjugacy class of G then*

$$g, h \in K \implies \chi(g) = \chi(h).$$

3. *If Y is a representation of G with character ψ, then*

$$X \cong Y \implies \chi(g) = \psi(g)$$

for all $g \in G$.

Note that we proved 3 *in our definition of a character.*

This next proposition is similar in flavor and characterizes the relationship between irreducible representations and their corresponding group.

Proposition 1.2.13 (Sagan 2001, Proposition 1.10.1)**.** *Let G be a finite group and suppose that the set V_i ranging over i forms a complete list of distinct irreducible G-modules. Then*

1. $\sum_i (\mathrm{dim} V_i)^2 = |G|$, *and*

2. *the number of V_i equals the number of conjugacy classes of G.*

1.3 Representation Theory of the Symmetric Group

1.3.1 Motivation

We will consider the symmetric group S_n and its representations; in doing so, we will find how these representations correspond to mathematical objects known as Young diagrams.

Definition 1.3.1. A *Young diagram* is a pictorial presentation of an *integer partition.* We define an integer partition λ of an integer $n \in \mathbb{Z}$ to be a sequence of integers, indexed by λ_i, such that:

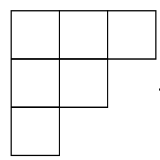
1. the sequence is (weakly) decreasing:

$$\forall i \in \mathbb{N}, \lambda_i \geq \lambda_{i+1}.$$

2. The sequence sums to n:
$$\sum_{i \in \mathbb{N}} \lambda_i = n.$$

3. And each index is nonnegative:
$$\forall i \in \mathbb{N}, \lambda_i \geq 0.$$

We usually leave out the trailing zeros of an integer partition, and denote an integer partition of n with the notation $\lambda \vdash n$ or $(\lambda_1, \lambda_2, \cdots) \vdash n$. A Young diagram represents an integer partition as a picture by drawing λ_i boxes in the i-th row, from top to bottom. For example, the integer partition $(3, 2, 1) \vdash 6$ is represented by the Young diagram


.

We have previously stated that the conjugacy classes of a group are in bijection with the irreducible modules of G. It is a fundamental abstract algebra fact that conjugacy classes of S_n are in bijection with the cycle types of S_n, and because cycle types are unordered lists we can map each cycle type to its unique weakly decreasing ordering. This forms a bijection between cycle types of S_n and integer partitions of n. Composing bijections, we can conclude that each integer partition of n uniquely corresponds to an irreducible representation of S_n!

Our goal will be to find these irreducible representations by creating their corresponding irreducible S_n-modules.

1.3.2 Tableaux and Tabloids

In order to use tableaux in S_n-modules, we'll need to define the action of S_n on a tableau of a partition of size n.

Definition 1.3.2. A *Young Tableau* is a numbering of a Young diagram of size n with the numbers $\{1 \cdots n\}$ with no repeats allowed.

There is no ordering condition, unlike with a further refinement we will use later called standard Young tableaux (hereafter referred to as SYT).

The action of $\sigma \in S_N$ on a tableau T of size n is to replace the box i in T with the box $\sigma(i)$ in $\sigma \cdot T$.

Definition 1.3.3 (Fulton 1997, Pg. 84)**.** The *row group* of T, denoted $R(T)$, is the set of permutations which permute the entries of each row among themselves. If $\lambda = (\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k < 0)$, then $R(T)$ is a product of symmetric groups $S_{\lambda_1} \times S_{\lambda_2} \times \cdots \times S_{\lambda_k}$. In fact, if row 1 of T contains the numbers $\{j_1 \cdots i_1\} \subseteq [n]$, row 2 contains $\{j_2 \cdots i_2\}$, et cetera, then $R(T)$ is the product of the permutations of each set — $S_{\{j_1 \cdots i_1\}} \times S_{\{j_2 \cdots i_2\}} \times \cdots \times S_{\{j_k \cdots i_k\}}$.

We analogously define the column group of T, $C(T)$, to be the set of permutations which permute the entries of the columns among themselves. This is equivalent to the row group of the transpose of the tableau.

These subgroups of S_n are compatible with the action of S_n on T in the following way:

$$R(\sigma \cdot T) = \sigma \cdot R(T) \cdot \sigma^{-1} \text{ and } C(\sigma \cdot T) = \sigma \cdot C(T) \cdot \sigma^{-1}.$$

Example 1.3.4. For the tableau below, $(143) \in R(T)$ but $(12) \notin R(T)$. Analogously $(143) \notin C(T)$ and $(12) \in C(T)$.

$$\begin{array}{|c|c|c|}\hline 1 & 4 & 3 \\ \hline 2 & 5 \\ \cline{1-2}\end{array}$$

If we perform the action $(12)(34) \cdot T$ we now see that $\sigma(143)\sigma^{-1} = (12)(34)(143)(12)(34) = (243) \in R(\sigma \cdot T)$. Intuitively conjugation is swapping elements of the underlying set undergoing a bijection.

$$\begin{array}{|c|c|c|}\hline 2 & 3 & 4 \\ \hline 1 & 5 \\ \cline{1-2}\end{array}$$

We now define tabloids, which will be used in our construction of S_n-modules.

Definition 1.3.5 (Fulton 1997, Pg. 85)**.** A *tabloid* is an equivalence class of tableaux where two tableaux of the same shape are equivalent if their rows contain the same values. The tabloid containing T is denoted $\{T\}$. Two tableaux are in the same class ($\{T\} = \{T'\}$) when $T' = \sigma \cdot T$ for some $\sigma \in R(T)$.

Example 1.3.6. Tabloids are notated as tableaux without vertical lines to convey that changing the ordering of values within a row doesn't change the tabloid.

$$\begin{array}{ccc}\hline 1 & 4 & 3 \\ \hline 2 & 5 & \\ \cline{1-2}\end{array} = \begin{array}{ccc}\hline 4 & 1 & 3 \\ \hline 2 & 5 & \\ \cline{1-2}\end{array} \neq \begin{array}{ccc}\hline 4 & 5 & 3 \\ \hline 2 & 1 & \\ \cline{1-2}\end{array}$$

1.3.3 The modules M^λ and S^λ

Before defining M^λ, we must first linearly extend S_n to the group ring $\mathbb{C}[S_n]$, the formal linear combinations of permutations with coefficients in $\mathbb{C}$ — $\sum x_\sigma \sigma$. Multiplication in the group ring is determined by composition. Note that we can restrict a $\mathbb{C}[S_n]$-module to an S_n-module by restricting $\mathbb{C}[S_n]$ to the set

$$\{1_{\mathbb{C}}\sigma | \sigma \in S_n\}$$

which is isomorphic to S_n, so we still obtain our desired S_n-module and can construct any desired matrix representations of S_n.

Definition 1.3.7 (Fulton 1997, Pg. 86)**.** We let M^λ denote the complex vector space with basis the set of tabloids $\{T\}$ for a given partition λ of size n.

Since S_n acts on the set of tabloids, it acts on M^λ, and we can linearly extend this to an action of $\mathbb{C}[S_n]$ on M^λ; namely

$$(\sum x_\sigma \sigma) \cdot (\sum x_T \{T\}) = \sum \sum x_\sigma x_T \{\sigma \cdot T\}.$$

Therefore M^λ is a $\mathbb{C}[S_n]$-module.

We now need some special elements to construct our desired submodule of M^λ:

Definition 1.3.8 (Fulton 1997, Pg. 86)**.** Given a tableau T, we define the element $b_T \in \mathbb{C}[S_n]$:

$$b_T = \sum_{q \in C(T)} \operatorname{sgn}(q) q.$$

This is a *Young symmetrizer*; there are two others we have left undefined as they are used in the symmetric (column-wise) definition of tabloids outside of our scope.

Definition 1.3.9 (Fulton 1997, Pg. 86)**.** For each tableau (not tabloid!) T of shape λ there is an element $v_T \in M_\lambda$ defined by the formula

$$v_T = b_T \cdot \{T\} = \sum \operatorname{sgn}(q) \{q \cdot T\}.$$

Note that changing the tableau T might not change the tabloid $\{T\}$ but could change b_T, resulting in a different element in M^λ.

We can now define the subspace S^λ:

Definition 1.3.10 (Fulton 1997, Pg. 87)**.** The *Specht module*, denoted S^λ, is the subspace of M^λ spanned by the elements v_T, as T varies over all tableaux of λ.

To prove that S^λ is actually a module, we need to show that it is closed under the action of $\mathbb{C}[S_n]$; namely, we will show that $\sigma \cdot v_T = v_{\sigma \cdot T}$ for all tableaux T and $\sigma \in S_n$.

Proof. Recall that $C(\sigma \cdot T) = \sigma \cdot C(T) \cdot \sigma^{-1}$; we first observe that

$$\begin{aligned}
\sigma \cdot v_T &= \sigma \cdot b_T \cdot \{T\} \\
&= \sigma \cdot \sum_{q \in C(T)} \operatorname{sgn}(q)\{q \cdot T\} \\
&= \sum_{q \in C(T)} \operatorname{sgn}(q)\{\sigma \cdot q \cdot T\}.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
v_{\sigma \cdot T} &= b_{\sigma \cdot T}\{\sigma \cdot T\} \\
&= \sum_{q \in C(\sigma \cdot T)} \operatorname{sgn}(q)\{q \cdot \sigma \cdot T\} \\
&= \sum_{q \in \sigma \cdot C(T) \cdot \sigma^{-1}} \operatorname{sgn}(q)\{q \cdot \sigma \cdot T\} \\
&= \sum_{q \in C(T)} \operatorname{sgn}(\sigma q \sigma^{-1})\{\sigma q \sigma^{-1} \cdot \sigma \cdot T\} \\
&= \sum_{q \in C(T)} \operatorname{sgn}(\sigma q \sigma^{-1})\{\sigma \cdot q \cdot T\}
\end{aligned}$$

Recall that $A_n \trianglelefteq S_n$, so conjugation does not change the sign of a permutation; these two expressions are therefore equal. □

Because $\sigma \cdot v_T = v_{\sigma \cdot T} \in S^\lambda$, S^λ is closed under S_n. Furthermore S^λ is closed under $\mathbb{C}[S_n]$, as we can now calculate

$$(\sum x_\sigma \sigma) \cdot v_T = \sum x_\sigma v_{\sigma \cdot T} \in S^\lambda.$$

Because S^λ is a subspace of M^λ, we have thus proven that

Theorem 1.3.11. *S^λ is a $\mathbb{C}[S_n]$ submodule of M^λ.*

In fact, we know that $S^\lambda = \mathbb{C}[S_n] \cdot v_T$ for *any given* tableau T, because we can create any desired element in S^λ like so:

$$\sum_{\{T'\}\in A \subseteq \text{ set of tabloids}} x_{T'} v_{T'} = \left(\sum x_{T'} \sigma_{T'}\right) \cdot v_T,$$

where $\sigma_{T'} \cdot v_T = v_{\sigma_{T'} \cdot T} = v_{T'}$.

1.3.4 Irreducibility and completeness of the S^λ-modules

We now need to show that the modules S^λ have our desired properties: every S^λ is irreducible, no two S^λ are isomorphic, and any irreducible representation is isomorphic to some S^λ. Once we do this we will have shown that the set of modules S^λ over partitions of a given size n are in bijection with the irreducible representations of S_n (up to isomorphism)!

The idea of our proof will be to show that each S^λ is indecomposable (contains no nontrivial submodules) and distinct by putting a linear order on the tableaux of any shape and of size n. Maschke's theorem tells us that indecomposability is equivalent to irreducibility, and we will use some properties of the ordering to show that each S^λ is disjoint to $S^{\lambda'}$ when $\lambda < \lambda'$ and $\lambda \neq \lambda'$.

We begin by defining the lexicographical and dominance orderings:

Definition 1.3.12 (Fulton 1997, pg. 36)**.** The *lexicographic* ordering on partitions of size n, denoted $\lambda \leq \lambda'$, means that for the first i for which $\lambda_i \neq \lambda'_i$ (if any), has $\lambda_i < \lambda'_i$ (in the standard ordering on integers). It is a linear order.

Definition 1.3.13. The *dominance* ordering on partitions of size n, denoted $\lambda \trianglelefteq \lambda'$ or "λ' dominates λ", means that $\sum_{1\leq j\leq i} \lambda_j \leq \sum_{1\leq j\leq i} \lambda'_j$ for all $1 \leq i \leq \infty$.

The intuition here is that partitions with a few long rows dominate partitions with many short rows; note, however, that this is not a linear order. For example, we see for the partitions $\lambda = (4,1,1,1) \vdash 7$ and $\lambda' = (3,3,1) \vdash 7$ that

$$\begin{aligned}
&4 \nleq 3, \text{ so } \lambda \ntrianglelefteq \lambda' \\
&3 \leq 4, \ 3+3 \leq 4+1 \text{ so } \lambda' \ntrianglelefteq \lambda
\end{aligned}$$

so λ and λ' are incomparable.

We can now state the following lemma:

Lemma 1.3.14 (Fulton 1997, Lemma 7.1)**.** *Let T and T' be tableaux of shape λ and λ' respectively, each of size n. Note that they can have different shape! Assume that λ does not strictly dominate λ'. Then exactly one of the following occurs:*

1. *There are two distinct integers that occur in the same row of T' and the same column of T.*

2. *$\lambda = \lambda'$, and there is some p' in $R(T)$ and some q in $C(T)$ such that $p' \cdot T' = q \cdot T$.*

Proof. Suppose 1 is false. The entries of the first row of T' must occur in different columns of T, so there is a $q_1 \in C(T)$ so that these entries occur in the first row of $q \cdot T$.

The entries of the second row of T' occur in different columns in T, and so they also occur in different columns of $q_1 \cdot T$, so there is a $q_2 \in C(q_1 \cdot T) = C(T)$ that:

1. Doesn't move entries in $q_1 \cdot T$ that are also in the first row of T'.

2. Moves entries in the second row of T' that are in $q_1 \cdot T$ into the second row of $q_1 \cdot T$.

We can repeat this process to obtain $q_1, \ldots, q_k$ such that the entries in the first k rows of T' occur in the first k rows of $q_k \cdot q_{k-1} \cdot \cdots \cdot q_1 \cdot T$. The actions of $q_1, \ldots, q_k$ don't change the shape of T, so we can deduce that

$$\sum_{1 \leq j \leq k} \lambda'_j \leq \sum_{1 \leq j \leq k} \lambda_j$$

because each row of λ must have at least enough boxes to contain all the entries of the corresponding row of λ'. This holds for all k, so by definition λ dominates λ'.

We assumed that λ doesn't strictly dominate λ', so the only possibility is that $\lambda = \lambda'$. We can thus take k to be the number of rows in λ and let $q = q_k \cdots q_1$ so that $q \cdot T$ and T' have the same entries in each row. We can now conclude that there is some $p' \in R(T)$ such that $p' \cdot T' = q \cdot T$. □

We now define our needed linear ordering; note that this ordering is on all tableaux of size n, not just partitions!

Definition 1.3.15 (Fulton 1997, pg. 84-85)**.** We denote the linear ordering $T < T'$ on tableaux to mean either:

1. The shape of T' is larger than the shape of T in the lexicographical order.
2. T and T' have the same shape, and the largest entry that is in a different box in the two numberings occurs earlier in the column word of T' than in the column word of T.

   The column word is obtained by listing the entries of each column from bottom to top, reading columns from left to right.

Example 1.3.16. This ordering puts the standard tableaux of shape $(3, 2)$ in the following order:

$$\begin{array}{|c|c|c|}\hline 1&2&3\\ \hline 4&5\\ \cline{1-2}\end{array} > \begin{array}{|c|c|c|}\hline 1&2&4\\ \hline 3&5\\ \cline{1-2}\end{array} > \begin{array}{|c|c|c|}\hline 1&3&4\\ \hline 2&5\\ \cline{1-2}\end{array} > \begin{array}{|c|c|c|}\hline 1&2&5\\ \hline 3&4\\ \cline{1-2}\end{array} > \begin{array}{|c|c|c|}\hline 1&3&5\\ \hline 2&4\\ \cline{1-2}\end{array}.$$

We demonstrate the first inequality by observing that 4 is the largest entry in a different box between the two SYT, and that the first SYT has column word 41523 while the second has column word 31524. We see that 4 occurs earlier in the column word of the first SYT, so the first SYT is "larger" in this ordering.

An important property of this ordering for SYT T and any $p \in R(T), q \in C(T)$ is that

$$p \cdot T < T \text{ and } q \cdot T > T.$$

The symbol "$<$" does not denote a strict poset relation, so it could be that $p \cdot T = T$.

This is true because the largest element moved by a row permutation must be moved left, pushing it closer to the front in the column word, while the largest element moved by a column permutation must be moved up, pushing it further back in the column word.

Example 1.3.17. In the SYT below, any nontrivial row permutation will swap at least two elements. Because every element to the right of a given entry in a row is larger in a SYT, this swap will move the larger element to the left. Analogously, elements lower in a column are larger, so a column permutation will move the larger element up.

$$(15) \cdot \begin{array}{|c|c|c|c|c|}\hline 1&2&3&4&5\\ \hline\end{array} = \begin{array}{|c|c|c|c|c|}\hline 5&2&3&4&1\\ \hline\end{array}$$

We need the following lemmas, given as exercises on Fulton 1997, Page 86, for the next 2 proofs:

Lemma 1.3.18. *For all $q' \in C(T)$,*

$$q' \cdot b_T = \operatorname{sgn}(q') b_T.$$

Proof. First we compute

$$q' \cdot b_T = q' \cdot \sum_{q \in C(T)} \operatorname{sgn}(q)\{q \cdot T\} = \sum_{q \in C(T)} \operatorname{sgn}(q)\{q' \cdot q \cdot T\}.$$

We know from abstract algebra that $q'C(T) = C(T)$ because $q' \in C(T) \leq S_n$; the function of applying q' to every element of $C(T)$ is a bijection. Therefore the composition of q' with every element of $C(T)$ will only reorder the addends in the sum. If q' is odd, it will map each permutation to a permutation with opposite sign; if it is even, it will maintain the parity of each permutation. This is equivalent to multiplying each permutation by the sign of q', so

$$\sum_{q \in C(T)} \operatorname{sgn}(q)\{q' \cdot q \cdot T\} = \sum_{q \in C(T)} \operatorname{sgn}(q)\operatorname{sgn}(q')\{q \cdot T\} = \operatorname{sgn}(q') \sum_{q \in C(T)} \operatorname{sgn}(q)\{q \cdot T\} = \operatorname{sgn}(q')b_T.$$

□

Lemma 1.3.19.

$$b_T \cdot b_T = |C(T)| \cdot b_T,$$

where $\cdot$ here is multiplication in the group ring.

Proof. We linearly extend the result we just proved:

$$b_T \cdot b_T = (\sum_{q \in C(T)} \operatorname{sgn}(q)q) \cdot b_T = |C(T)|\operatorname{sgn}(q)\operatorname{sgn}(q) \cdot b_T = |C(T)| \cdot b_T.$$

□

We now state and prove another lemma which applies our previous lemma to the M^λ-module:

Lemma 1.3.20 (Fulton 1997, Lemma 7.2)**.** *Let T and T' be numberings of shapes λ and λ' respectively, and assume that λ does not strictly dominate λ'.*

1. *If there is a pair of integers in the same row of T' and the same column of T, then $b_T \cdot \{T'\} = 0$.*

2. *If there is no such pair, then $b_T \cdot \{T'\} = \pm v_T$.*

Proof. If there is such a pair of integers, let $t \in S_n$ be the transposition that swaps them. Then $b_T \cdot t = -b_T$, since t is in the column group of T, and transpositions have odd sign. On the other hand, $t \cdot \{T'\} = \{T'\}$ by the definition of a tabloid, as t is in the row group of T'. Therefore

$$b_T \cdot \{T'\} = b_T \cdot (t \cdot \{T'\}) = (b_T \cdot t) \cdot \{T'\} = -b_T \cdot \{T'\}$$

so $b_T \cdot \{T'\} = 0$.

If there is no such pair of integers, then let p' and q be as in the second case of our first lemma. Then

$$\begin{aligned} b_T \cdot \{T'\} &= b_T\{p' \cdot T'\} = b_T \cdot \{q \cdot T\} \\ &= b_T \cdot q \cdot \{T\} = \operatorname{sgn}(q) b_T \cdot \{T\} = \operatorname{sgn}(q) v_T. \end{aligned}$$

□

Finally, we have the tools to prove the main theorem:

Theorem 1.3.21 (Fulton 1997, Pg. 87-88)**.** *For each partition λ of n, S^λ is an irreducible representation of S_n. Every irreducible representation of S_n is isomorphic to exactly one S^λ.*

Proof. First, we note that no v_T is 0 by definition, so the modules S^λ are all nonzero (nontrivial subspaces of M^λ). We wish to prove the following statements, for a given tableau T of λ:

$$b_T \cdot M^\lambda = b_T \cdot S^\lambda = \mathbb{C} \cdot v_T \neq \{0\}. \tag{1.3.22}$$

$$b_T \cdot M^{\lambda'} = b_T \cdot S^{\lambda'} = \{0\} \text{ if } \lambda < \lambda' \text{ and } \lambda \neq \lambda', \tag{1.3.23}$$

where $\{0\}$ is the trivial zero subspace. We begin with the first equation.

The first equality follows as such, using our result that $b_T \cdot b_T = |C(T)| \cdot b_T$:

$$\begin{aligned} b_T \cdot b_T = |C(T)| \cdot b_T &\iff \\ \frac{1}{|C(T)|} \cdot b_T \cdot b_T = b_T&; \\ b_T \cdot M^\lambda &= \frac{1}{|C(T)|} \cdot b_T \cdot b_T \cdot M^\lambda \\ &= \frac{1}{|C(T)|} \cdot b_T S^\lambda \\ &= b_T \cdot \frac{1}{|C(T)|} \cdot S^\lambda \\ &= b_T \cdot S^\lambda \end{aligned}$$

where we can commute $\frac{1}{|C(T)|}$ because it is only a scalar; furthermore $\frac{1}{|C(T)|} \cdot S^\lambda = S^\lambda$ because vector spaces are invariant under scaling. The second equality follows because we know that for any T, T' tableaux of λ, we know by Lemma 1.3.14 there are not two distinct integers that appear in the same row of T' and in the same column of T, so by Lemma 1.3.20 $b_T \cdot \{T'\} = \pm v_T$. This means that $b_t \cdot M^\lambda = \mathbb{C} \cdot v_T$, because every element in M^λ is mapped to a scaling of v_T, and any we can obtain any scaling of v_T via $b_T \cdot x\{T\} = xv_T \in \mathbb{C} \cdot v_T$.

For the second equation, the first equality follows from our argument regarding the second equation. The second equality follows because by assumption we are in case 1 of Lemma 1.3.14 and therefore case 1 of Lemma 1.3.20, so $b_T \cdot \{T'\} = 0$ for all $\{T'\} \in M^\lambda$, and so $b_T \cdot M^\lambda = 0$.

Now by Maschke's theorem we know that $S^\lambda = W_1 \oplus W_2 \oplus \cdots \oplus W_k$ for irreducible submodules W_j. We see that

$$\mathbb{C} \cdot v_T = b_T \cdot S^\lambda = b_T \cdot W_1 \oplus b_T \cdot W_2 \cdots \oplus b_T \cdot W_k,$$

so one of the modules W_j must contain v_T. If some W_i contains v_T, then by the definition of a submodule $W_i = \mathbb{C}[S_n] \cdot v_T = S^\lambda$, so the other submodules must be the zero submodule and so S^λ is irreducible.

Furthermore we prove that $S^\lambda \ncong S^{\lambda'}$ for any $\lambda < \lambda'$ and $\lambda \neq \lambda'$. We assume there exists a module isomorphism Θ between S^λ and $S^{\lambda'}$. By the definition of a module isomorphism, $\Theta(b_T \cdot S^{\lambda'}) = b_T \cdot \Theta(S^{\lambda'})$, but we just showed that

$$\Theta(b_T \cdot S^{\lambda'}) = \Theta(0) = 0 \neq b_T \cdot S^\lambda = b_T \cdot \Theta(S^{\lambda'})$$

which is a contradiction. Therefore $S^\lambda \ncong S^{\lambda'}$, and because $<$ is a linear ordering, we can conclude that no two distinct S^λ are isomorphic.

Finally, we cite Sagan 2001, Proposition 1.10.1, specifically the result that the number of irreducible modules/representations equals the number of conjugacy classes of the group. We have previously discussed how the conjugacy classes of S_n are in bijection with cycle types of S_n, which in turn are in bijection with partitions of n. Because there is one Specht module S^λ for each partition λ of size n, we can conclude that there are exactly as many S^λ as there are irreducible representations of S_n. That is, the set of modules S^λ is all of the irreducible modules/representations of S_n up to isomorphism. $\square$

1.3.5 A basis for S^λ

With the main result aside, we will provide a basis for the S^λ-modules using Young tableaux with a certain restriction.

Definition 1.3.24 (Sagan 2001, Definition 2.5.1)**.** A tableau T is *standard* if the rows and columns of T are increasing sequences. In this case we also say that the corresponding tabloid and polytabloid are standard.

Example 1.3.25. The tableau

$$T = \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 4 & 6 & \multicolumn{1}{c}{} \\ \cline{1-2} 5 & \multicolumn{2}{c}{} \\ \cline{1-1}\end{array}$$

is standard, but

$$T = \begin{array}{|c|c|c|}\hline 1 & 2 & 3 \\ \hline 5 & 4 & \multicolumn{1}{c}{} \\ \cline{1-2} 6 & \multicolumn{2}{c}{} \\ \cline{1-1}\end{array}$$

is not.

Proposition 1.3.26 (Fulton 1997, Pg. 88)**.** *The elements v_T, as T varies over the* standard *tableaux of λ, form a basis for S^λ.*

Proof. The element v_T is a linear combination of $\{T\}$, with coefficient 1 (as the trivial permutation has even sign), and elements $\{q \cdot T\}$, for $q \in C(T)$, with coefficients ± 1. Recall that when T is a SYT, $q \cdot T < T$, and furthermore this relation is strict when q is nontrivial. To find solutions to the equation $\sum_{T \in \text{ SYT of } \lambda} x_T v_T = 0$, we can look at the largest v_T with nonzero coefficient x_T. We know that the $\{T\}$ component cannot be canceled out by some other $v_{T'}$, because it must be that $\{T\} \neq \{T'\}$ in order for the relation $v_T > v_{T'}$ to be strict, and furthermore $\{T\} \neq \{q \cdot T'\}$ because we know $\{T\} > \{T'\} > \{q \cdot T'\}$ and at least the first relation is strict. Thus $\{T\}$ cannot be canceled out by some other term, so it must be that $x_T = 0$; but then all $x_T = 0$ because if any x_T is nonzero there will be some largest nonzero v_T. We can thus conclude that the elements $v_T \in S^\lambda$ as T varies over the SYT are linearly independent.

To show that these elements span S^λ, we again cite Sagan 2001, Proposition 1.10.1, specifically the result that

$$\sum_i (\text{dim} V_i)^2 = |G|$$

where the V_i are a complete set of irreducible G-modules. In our case this means that

$$\sum_{\lambda} (\dim S^{\lambda})^2 = n!.$$

The Robinson-Schensted algorithm in Sagan 2001, Theorem 3.1.1 provides a bijection between the symmetric group S_n and pairs of standard Young tableaux; while its proof is outside our scope it proves combinatorially that

$$\sum_{\lambda} (f^{\lambda})^2 = n!,$$

where f^{λ} is the number of SYT of the partition λ of size n. We can thus conclude

$$n! = \sum_{\lambda} (\dim S^{\lambda})^2 = \sum_{\lambda} (f^{\lambda})^2 = n!.$$

It follows that $\dim(S^{\lambda}) = f^{\lambda}$ for all λ, and because f^{λ} counts the number of SYT of shape λ this means that the elements v_T as T varies over SYT must span S^{λ}. $\square$

1.4 Oracle Formalism

We are now equipped to state our oracle problem and key theorem formally.

Definition 1.4.1 (Copeland and Pommersheim 2021, Section 2)**.** A *classical oracle problem* is a tuple (Y, Ω, π, f) where

1. Y is a set of hidden information.
2. Ω is a set of inputs algorithms can query.
3. π is a function $\pi : Y \to \mathrm{Sym}(\Omega)$, where $\mathrm{Sym}(\Omega)$ is the group of permutations of Ω.
4. And $f : Y \to X$ is the function to learn; it is known to the algorithm.

A classical computer has access to $\pi(y)$ for some unknown $y \in Y$ by spending one query to learn $\pi(y) \cdot \omega$. The goal is to determine $f(y)$. The average case case success probability is the probability of correctly outputting $f(y)$ assuming y is sampled uniformly from Y.

Definition 1.4.2 (Copeland and Pommersheim 2021, Section 2)**.** A *quantum oracle problem* is a tuple (Y, V, π, f) where

1. Y is a set of hidden information.
2. V is a Hilbert space appearing as a register in our quantum circuit.
3. π is a function $\pi : Y \to U(V)$, where $U(V)$ is the set of unitary operators of V.
4. And $f : Y \to X$ is the function to learn.

A quantum computer spends one query to input a state $|\psi\rangle \in V$ to $\pi(y)$ to acquire the state $\pi(y)\,|\psi\rangle$. Any classical oracle problem (Y, ω, π, f) determines a quantum oracle problem via linearization: oracles will act on the Hilbert space $\mathbb{C}\Omega$ by permutation matrices. Henceforth when we provide an instance of a classical oracle problem we are also providing an instance of a quantum oracle problem.

Definition 1.4.3 (Copeland and Pommersheim 2021, Section 2)**.** A *symmetric oracle problem* is a classical/quantum oracle problem where

1. we require that the hidden information Y be a group G.
2. Ω or V are as they were before.
3. π is as before, but now also is a homomorphism to the group $\mathrm{Sym}(\Omega)$ in the classical case and to the group $U(V)$ in the quantum case.
4. And f is as it was before.

When $\pi : G \to U(V)$ is a homomorphism, it is a representation as we have defined before; it is therefore natural to regard V as a (left) $\mathbb{C}G$-module where $\mathbb{C}G$ is the group algebra of G (spanned by the orthonormal basis $\{g|g \in G\}$ where we leave out kets to indicate their use as an action), due to the relationship between representations and modules.

Definition 1.4.4 (Copeland and Pommersheim 2021, Section 2)**.** A *coset identification problem* is a symmetric oracle problem where the function to be learned $f : G \to X$ is constant on left cosets of a subgroup $H \leq G$ and distinct on distinct cosets. We also assume f is surjective. The typical example is when $X = \{gH|g \in G\}$ is the set of left cosets of H and $f(g) = gH$. We cite without proof the fact that worst case and average case success probabilities are equal in the coset identification problem.

Definition 1.4.5 (Copeland and Pommersheim 2021, Section 2)**.** The *exact* query complexity of a learning problem, denoted γ, is the minimum number of queries needed by an algorithm to compute $f(y)$ with zero probability of error. The *bounded* error query complexity, denoted γ^{bdd}, is the minimum number of queries needed by an algorithm to compute $f(y)$ with probability $\geq 2/3$. The bounded error query complexity is often studied for a family of problems growing with a parameter n and so changing the constant $2/3$ above to any number strictly greater than $1/2$ will only change the query complexity by a constant factor mostly ignored in asymptotic analysis.

Definition 1.4.6 (Copeland and Pommersheim 2021, Section 4)**.** *Symmetric oracle discrimination* is a coset identification problem a la the typical example where $X = \{gH | g \in G\}$, $f(g) = gH$, and $H = \{e\}$, so $f(g) = ge = g$. This means an element g is hidden and the goal is to find g. We have fixed f, so such a problem is determined by a choice of finite group G, a G-set Ω or a vector space $\mathbb{C}\Omega$, and an action $G \circlearrowright \Omega$ or a (finite-dimensional) unitary representation $\pi : G \to U(V)$.

Our problem is thus two instances of symmetric oracle discrimination where, in the first case,

1. $G = S_n$.
2. Ω is the set of k elements subsets of $[n] = \{1, 2, \cdots, n\}$, denoted Ω_k.
3. $S_n \circlearrowright \Omega_k$ is the natural action of S_n:

$$\sigma \circlearrowright \{\omega_1, \omega_2, \cdots, \omega_k\} \mapsto \{\sigma(\omega_1), \sigma(\omega_2), \cdots, \sigma(\omega_k)\}.$$

We will refer to this problem as k-element subset discrimination, and the relevant module will be called the k-element module, and the relevant action the k-element action. In the second case,

1. $G = S_n$.
2. Ω is the set of regular partitions of $[n]$ into b parts of size a, so $ab = n$. We denote this set as Ω_p.
3. $S_n \circlearrowright \Omega_p$ is the natural action of S_n:

$$\begin{aligned}&\sigma \circlearrowright \{\{\omega_1, \omega_2, \cdots, \omega_a\}, \{\omega_{a+1}, \omega_{a+2}, \cdots, \omega_{2a}\}, \cdots, \{\omega_{(a-1)b+1}, \omega_{(a-1)b+2}, \cdots, \omega_{ab}\}\} \mapsto \\ &\{\{\sigma(\omega_1), \sigma(\omega_2), \cdots, \sigma(\omega_a)\}, \{\sigma(\omega_{a+1}), \sigma(\omega_{a+2}), \cdots, \sigma(\omega_{2a})\}, \cdots, \\ &\{\sigma(\omega_{(a-1)b+1}), \sigma(\omega_{(a-1)b+2}), \cdots, \sigma(\omega_{ab})\}\}.\end{aligned}$$

We will likewise refer to the associated objects here with the "regular partition" prefix. Finally, we introduce the theorem which our results rely upon.

Theorem 1.4.7. *Suppose G is a finite group and $\pi : G \to U(V)$ a unitary representation of G. Then an optimal t-query algorithm to solve symmetric oracle discrimination succeeds with probability*

$$P_{opt} = \frac{d_{V^{\otimes t}}}{|G|}$$

where

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

Note that $I(V^{\otimes t})$ is the set of irreducible constituent characters of the character $V^{\otimes t}$.

Proof Outline 1.4.8. In a paper separate to the one we cite, Copeland and Pommershiem proved a similar theorem for a single query algorithm, where only the irreducible consituent characters of V were considered. In Copeland and Pommersheim 2021, Copeland and Pommershiem prove that nonadaptive (all oracle queries made in parallel) quantum algorithms have equivalent strength to adaptive (arbitrary unitary transformations allowed between queries) algorithms. Furthermore, they prove that a t-query nonadaptive algorithm has equivalent success probability to a a single query algorithm which takes the tensor product of the representation $\pi : G \to U(V)$ with itself multiple times to the representation $\pi^{\otimes t} : G \to U(V^{\otimes t})$. We can thus substitute $\pi^{\otimes t} : G \to U(V^{\otimes t})$ in for the single query theorem to find the optimal t-query success chance.

This theorem tells us that we can determine the query complexity of a symmetric oracle discrimination problem just by understanding how the consituent irreducibles of the representation $\pi : G \to U(V)$ change as n grows, and how the repeated tensor product of these constituents introduces new irreducible representations. To make our work computationally possible we will use the characters of these irreducible representations to do our analysis rather than the representations themselves.

Chapter 2

Methods

We approached this problem by analyzing the query complexity and the constituent irreducibles of each action for small values of n. This was done via computation in the language *GAP – Groups, Algorithms, and Programming, Version 4.14.0* 2024, which offers native support for group and representation theory calculations. Computations using representations themselves are slow and space-intensive, so we study the representations we are interested in via their characters. Each action is created using built-in objects, then used to create a permutation character corresponding to the module $\mathbb{C}\Omega$, where Ω is the G-set of the action. With this character we can compute the query complexity and bounded query complexity with a simple algorithm, offered here in pseudocode:

Require: $permutationCharacter$ is the permutation character of $G \circlearrowright \Omega$

```
procedure QueryComplexity(permutationCharacter, G)
    queries ← 0
    boundedQueries ← 0
    ▷ tensorCharacter stores V^{⊗t}, while permutationCharcter stores V.   ◁
    tensorCharacter ← permutationCharacter
    repeat
        queries ← queries + 1
        constituents ← set of constituent characters of tensorCharacter
        probabilityOfSuccess ← ∑_{χ∈constituents} χ(e)^2/|G|
        if probabiltyOfSuccess ≥ 2/3 and boundedQueries = 0 then
            boundedQueries ← queries
        tensorCharacter ← tensorCharacter ⊗ permutationCharacter
    until constituents = set of irreducible characters of G
```

In addition, GAP offers a special character table for the symmetric group which uses the *Murnaghan-Nakayama Rule* to recursively associate a partition with each irreducible character of S_n. Using this table we were able to store the corresponding partitions of the constituent characters present after each query, which we then visualized in Python. Both these partitions and the behavior of the query complexity as n increased helped us make conjectures regarding the general query complexity of these actions.

Chapter 3

Results

We first decompose the k-element module $\mathbb{C}\Omega_k$, and in the special case $k = 1$ we also decompose $V \otimes \mathbb{C}\Omega_1$ for any S_n-module V. The latter result is assumed in Copeland and Pommersheim 2021, but no proof is provided or easily accessible online. Our other results will revolve around conjectures regarding the quantum query complexity of each oracle problem and the behavior of the tensor products of representations leading to this behavior.

We will also introduce the *base size* of $S_n \circlearrowright \Omega$ for the k-element action and the regular partition action as found in Mecenero and Spiga 2023 and Morris and Spiga 2021. The base size as defined in Copeland and Pommersheim 2021 is the length of the smallest tuple $(\omega_1, \cdots, \omega_t) \in \Omega^t$ with the property that $(g \cdot \omega_1, \cdots, g \cdot \omega_t) = (\omega_1, \cdots, \omega_t)$ if and only if $g = 1$, which is to say the tuple has trivial pointwise stabilizer. This definition corresponds to the non-adaptive (queries all made at the same time) classical query complexity; by the orbit stabilizer theorem, there is a bijective function which, given $g \cdot x$, returns the coset gG_{Ω^t}. Because the Cartesian product has trivial pointwise stabilizer—that is, $G_{\Omega^t} = e$—this is just the element g we aimed to find.

This, in fact, proves that the base size is also the *adaptive* classical query complexity, where the algorithm is allowed to consider $g \cdot \omega_i$ before choosing ω_{i+1} to query. Any adaptively chosen sequence $(\omega_1, \ldots, \omega_t)$ that suffices to identify some $g \in G$ in fact suffices to identify *every* element of G. Indeed, to rule out all other candidates one needs a trivial pointwise stabilizer, and the orbit–stabilizer argument then recovers g uniquely. Base size is thus equivalent to classical query complexity for symmetric oracle discrimination, and the results of Valle and Roney-Dougal 2023 and Morris and Spiga 2021 will allow us to compare the classical and quantum query complexity of our oracle problems. Because of this equivalence, we will sometimes refer to the quantum query complexity as the *quantum base size*, denoted γ. In fact, they will play an important role in our conjectures, because our data indicates that the classical and quantum query complexities are the same for k-element discrimination.

3.1 Decompositions

To decompose the k-element module, we will need more theorems from the representation theory of the symmetric group. Like before, we will be using Young tableaux; however, we are now interested in a less restrictive version of standard Young tableaux:

Definition 3.1.1 (Sagan 2001, p. 2.9.1)**.** A tableau is *semistandard* if the nodes of the Young diagram of the partition λ are filled with positive integers, repetitions allowed; additionally, its rows must be weakly increasing and its columns must be strongly increasing. The latter condition follows suit with standard tableaux, while now repetitions and integers greater than the number of nodes in the diagram are allowed. The *type* of a semistandard tableau is the tuple $\mu = (\mu_1, \mu_2, \cdots, \mu_m)$, where μ_i equals the number of the integer i in the tableau.

Example 3.1.2. The tableau

$$\begin{array}{|c|c|c|}\hline 1 & 1 & 2 \\ \hline 2 & 6 \\ \cline{1-2}\end{array}$$

is semistandard with type $(2, 2, 0, 0, 0, 1)$, whereas

$$\begin{array}{|c|c|c|}\hline 2 & 1 & 1 \\ \hline 6 & 2 \\ \cline{1-2}\end{array}$$

is not a semistandard tableau.

Definition 3.1.3 (Sagan 2001, p. 2.11.1)**.** The *Kostka numbers* $K_{\lambda\mu}$ are the number of semistandard tableaux of a partition λ with type μ.

Theorem 3.1.4 (Sagan 2001, p. 2.11.2, Young's Rule)**.** *The multiplicity of the Specht module S^λ in the permutation module M^μ is equal to the number of semistandard tableaux of shape λ and content μ; that is,*

$$M^\mu \cong \bigoplus_\lambda K_{\lambda\mu} S^\lambda.$$

Thanks to the following (unrelated) corollary, we will only have to calculate the Kostka numbers $K_{\lambda\mu}$ when $\lambda \trianglelefteq \mu$, as they are otherwise zero.

Corollary 3.1.5 (Sagan 2001, p. 2.4.7)**.** *The permutation modules decompose as*

$$M^{\mu} = \bigoplus_{\lambda \trianglelefteq \mu} m_{\lambda\mu} S^{\lambda}$$

where $m_{\lambda\mu}$ is a generic coefficient with diagonal multiplicity $m_{\mu\mu} = 1$.

The following lemma will enable us to use Young's rule to decompose our module into irreducibles.

Lemma 3.1.6. *Suppose $\mathbb{C}\Omega_k$ is the k-element module of n. Then*

$$\mathbb{C}\Omega_k \cong M^{(n-k,k)}$$

as modules.

Proof. For a given partition $\lambda = (n-k, k) \vdash n$, we define the function $f : \Omega_k \to \{\{t_1\}, \{t_2\}, \cdots \{t_j\}\}$ where $\{t_1\}, \{t_2\}, \cdots, \{t_j\}$ is a complete set of λ-tabloids:

proof of isomorphism by an explicit function is maybe less elegant than desired

$$\{\omega_1, \omega_2, \cdots, \omega_k\} \mapsto \begin{array}{cccc} \hline \omega_{k+1} & \omega_{k+2} & \cdots & \omega_n \\ \hline \omega_1 & \omega_2 & \cdots & \omega_k \\ \hline \end{array}.$$

The values $\omega_{k+1}, \omega_{k+2}, \cdots, \omega_n$ come from the set complement $[n] \setminus \{\omega_1, \omega_2, \cdots, \omega_k\} = \{\omega_{k+1}, \omega_{k+2}, \cdots, \omega_n\}$. This function linearly extends to a function $F : \mathbb{C}\Omega \to M^{n-k,k}$. Note that the rows of M^{λ} can be listed in any order and produce an isomorphic module. This means that M^{λ} is defined for *ordered* partitions λ , so this function is still well-defined if $k > n-k$. We observe that if

says this in the book but i don't know if this is convincing

$$f(\{\omega_1, \omega_2, \cdots, \omega_k\}) = f(\{\omega'_1, \omega'_2, \cdots, \omega'_k\}),$$

then

$$\{\omega_1, \omega_2, \cdots, \omega_k\} = \{\omega'_1, \omega'_2, \cdots, \omega'_k\};$$

because a $(n-k, k)$ tabloid is determined by the integers in its bottom row—the order of which doesn't matter due to the equivalence relation—and, hence, f is injective. Furthermore $|\Omega_k| = \binom{n}{k}$, and $|\{\{t_1\}, \{t_2\}, \cdots, \{t_j\}\}| = \binom{n}{k}$ because the tabloid is determined by choosing the integers in its bottom row, so f is a bijection and by

this feels really ugly, the bijection seems linear extension F is a vector space isomorphism. Finally, we observe that

$$
\begin{aligned}
\sigma \cdot f(\{\omega_1, \omega_2, \cdots, \omega_k\}) &= \sigma \cdot \begin{array}{cccc} \hline \omega_{k+1} & \omega_{k+2} & \cdots & \omega_n \\ \hline \omega_1 & \omega_2 & \cdots & \omega_k \\ \hline \end{array} \\
&= \begin{array}{cccc} \hline \sigma\cdot\omega_{k+1} & \sigma\cdot\omega_{k+2} & \cdots & \sigma \cdot \omega_n \\ \hline \sigma \cdot \omega_1 & \sigma \cdot \omega_2 & \cdots & \sigma \cdot \omega_k \\ \hline \end{array} \\
&= f(\sigma \cdot \{\omega_1, \omega_2, \cdots, \omega_k\})
\end{aligned}
$$

so by linear extension F is indeed a module isomorphism. $\square$

We are now ready to decompose $\mathbb{C}\Omega_k$ using Young's rule.

Lemma 3.1.7. *Suppose $\mathbb{C}\Omega_k$ is the k-element module of n. Then*

$$\mathbb{C}\Omega_k \cong \bigoplus_{j=0}^{k} S^{(n-j,j)}.$$

Proof. We first recall that $\mathbb{C}\Omega_k \cong M^{(n-k,k)}$, and apply corollary 3.1.5. For $K_{\lambda(n-k,k)}$ to be nonzero, λ must dominate $(n-k,k)$. If λ has parts in some third or more part which sum to nonzero h, then $n-k+k-h = n-h \not\geq n-k+k = n$, so it cannot dominate $(n-k,k)$. Furthermore, it cannot have a first part smaller than $n-k$, as then $n-k-h \not\geq n-k$. The only partitions left are the partitions $(n-j,j)$ for $j \in \mathbb{Z} : 0 \leq j \leq k$. These dominate $(n-k,k)$ as $n-j \geq n-k$ and $n-j+j = n \geq n-k+k = n$. Looking at the semistandard tableaux of shape $(n-j,j)$ with content $(n-k,k)$, we see that the only way to form a valid semistandard tableau is to place the $n-k$ 1's into the first row and to fill the rest of the slots with the k 2's:

$$\begin{array}{|c|c|c|c|c|c|c|c|} \hline 1 & 1 & \cdots & 1 & 2 & 2 & \cdots & 2 \\ \hline 2 & 2 & \cdots & 2 \\ \cline{1-4} \end{array}.$$

We can thus conclude by Young's rule that

$$\mathbb{C}\Omega_k \cong M^{(n-k,k)} \cong \bigoplus_{j=0}^{k} S^{(n-j,j)}.$$

$\square$

Our next result will rely on the concept of *restricted* and *induced* representations, where new representations are created using a group and its supergroup or subgroup.

Definition 3.1.8 (Fulton and Harris 2004, Section 3.3)**.** If $H \subseteq G$, any (module) representation of G restricts to a representation of H by restricting the action $G \circlearrowright V$ to $H \circlearrowright V$; likewise for a matrix representation X with G as its domain, the restriction $\mathrm{Res}_H^G(X)$ is a function with H as its domain where $\mathrm{Res}_H^G(X(h)) = X(h)$.

Suppose V is a (module) representation of G, and $W \subseteq V$ is a subspace which is H-invariant. For any g in G, the subspace $g \cdot W = \{g \cdot w : w \in W\}$ depends only on the left coset gH of g modulo H, since $gh \cdot W = g \cdot (h \cdot W) = g \cdot W$; for a coset σ in G/H, we write $\sigma \cdot W$ for this subspace of V. We say that V is induced by W if every element in V can be written uniquely as a sum of elements in such translates of W, i.e.,

$$V = \bigoplus_{\sigma \in G/H} \sigma \cdot W.$$

In this case we write $\mathrm{Ind}_H^G(W = V)$.

We assume the existence and uniqueness of induced representations as the proof is outside of our scope. We now need the following lemma which is left as an exercise in Fulton and Harris 2004, Exercise 3.16.

Lemma 3.1.9. *Suppose that U is a (module) representation of G and W a representation of H. Then*

$$U \otimes \mathrm{Ind}(W) = \mathrm{Ind}(\mathrm{Res}(U) \otimes W).$$

In particular, $\mathrm{Ind}(\mathrm{Res}(U)) = U \otimes P$*, where P is the permutation representation of G on G/H.*

Proof. We observe that

$$\begin{aligned}
\mathrm{Ind}_H^G((\mathrm{Res}_H^G(U) \otimes W)) &= \bigoplus_{\sigma \in G/H} (\mathrm{Res}_H^G(U) \otimes W) \\
&= (\bigoplus_{\sigma \in G/H} \mathrm{Res}_H^G(U)) \otimes (\bigoplus_{\sigma \in G/H} W) \\
&= (\mathrm{Ind}_H^G(\mathrm{Res}_H^G(U))) \otimes (\mathrm{Ind}_H^G(W)) \\
&= U \otimes \mathrm{Ind}_H^G(W).
\end{aligned}$$

If we let W be the trivial representation of H, then as a subspace W will be trivial, and the subspace $g \cdot W$ is determined by the coset $gH \in G/H$. Therefore this induced representation P has basis $\{e_\sigma : \sigma \in G/H\}$, so P is the permutation representation of G/H. By substitution we see that

$$\mathrm{Ind}_H^G((\mathrm{Res}_H^G(U) \otimes \mathbb{C}e_H)) = \mathrm{Ind}_H^G(\mathrm{Res}_H^G(U)) = U \otimes \mathrm{Ind}_H^G(W) = U \otimes P,$$

which is the particular formula we were looking for. □

We will combine this lemma with the following theorem:

Theorem 3.1.10 (Sagan 2001, p. 2.8.3, Branching Rule). *If $\lambda \vdash n$, then*

1. $\mathrm{Res}_{S_{n-1}}^{S_n}(S^\lambda) \cong \bigoplus_{\lambda^-} S^{\lambda^-}$, *and*

2. $\mathrm{Ind}_{S_n}^{S_{n+1}}(S^\lambda) \cong \bigoplus_{\lambda^+} S^{\lambda^+}$,

where λ^- is any partition created by the removal of a node in λ such that the result is a partition, and likewise λ^+ is any partition created by the valid addition of a node to λ.

We now can provide a simple proof of the lemma used on page 13 by Copeland and Pommersheim 2021:

Lemma 3.1.11. *Let V_λ be the Specht module corresponding to the partition $\lambda \vdash n$, and let V be the permutation module $V = S^{(n)} \oplus S^{(n-1,1)}$. Then*

$$V \otimes V_\lambda \cong \bigoplus_{\mu \in \lambda^\pm} (S^\mu)^{m_\mu}$$

for positive m_μ; $\lambda^\pm$ is the set of partitions of n obtained by removing then adding a valid node to λ, and $(S^\mu)^{m_\mu} = S^\mu \oplus S^\mu \oplus \cdots \oplus S^\mu$ m_μ times.

The multiplicity m_μ is left undetermined because it is not relevant to the quantum base size.

Proof. We observe that V is the permutation representation of S_n/S_{n-1}, so using the

expand on this or prove it maybe

permutation representation case of lemma 3.1.9 we see that

$$\begin{aligned}
V \otimes V_\lambda &\cong \mathrm{Ind}_{S_{n-1}}^{S_n}(\mathrm{Res}_{S_{n-1}}^{S_n}(V_\lambda)) \\
&\cong \mathrm{Ind}_{S_{n-1}}^{S_n}(\bigoplus_{\lambda^-} S^{\lambda^-}) \\
&\cong \bigoplus_{\lambda^-} \mathrm{Ind}_{S_{n-1}}^{S_n}(S^{\lambda^-}) \\
&\cong \bigoplus_{\lambda^-} \bigoplus_{\lambda^+} S^{\lambda^{-+}} \\
&\cong \bigoplus_{\mu \in \lambda^\pm} m_\mu S^\mu
\end{aligned}$$

where the distributivity of induction is proven in Fulton and Harris 2004, Exercise 3.15. □

3.2 Data and Conjectures

The problem of finding the multiplicities of irreducible representations present in the decomposition of the tensor product of two S_n-modules, known as finding the *Kronecker coefficients*, is generally unsolved and very difficult. Formally the Kronecker coefficients $g_{\mu\nu}^\lambda$ of $\mu, \nu, \lambda \vdash n$, as defined in Wang 2021, Definition 2.4.1, appear in the decomposition

$$S^\mu \otimes S^\nu \cong \bigoplus_{\lambda \vdash n} (S^\lambda)^{\oplus g_{\mu\nu}^\lambda}$$

where $(S^\lambda)^{\oplus g_{\mu\nu}^\lambda} = S^\lambda \oplus S^\lambda \oplus \cdots \oplus S^\lambda$ $g_{\mu\nu}^\lambda$ times. In terms of characters, the Kronecker coefficients can be solved for as the projection of the tensor product of χ^μ and χ^ν onto χ^λ:

$$g_{\mu\nu}^\lambda = \langle \chi^\lambda, \chi^\mu \chi^\nu \rangle = \frac{1}{n!} \sum_{\pi \in S_n} \chi^\mu(\pi) \chi^\nu(\pi) \chi^\lambda(\pi),$$

where the inner product of characters $\langle \chi, \phi \rangle$ over the complex numbers is defined as $\frac{1}{|G|} \sum_{g \in G} \chi(g) \overline{\phi(g)}$. Through this latter method we are able to compute the Kronecker coefficients for small partitions, although computing Kronecker coefficients is in general $P\#$ hard (as shown in Bürgisser and Ikenmeyer 2008, Theorem 1) and thus intractable. The difficulty of this problem meant that we were able to find evidence for conjectures regarding the behavior of the quantum query complexity of the k element action but were unable to prove them in general. The regular partition action offered less evidence and presents an even greater challenge to understand.

3.2.1 The k-element Action

We computed the quantum query complexity of $S_n \circlearrowleft \mathbb{C}\Omega_k$ for small values of n and k as shown in Table 3.1. Of interest are the cases where $n = 2k$, which correspond to the asymptotics of the logarithm of the Dedekind numbers $\binom{n}{\lfloor n/2 \rfloor}$.

maybe don't gain anything from saying this

We now introduce the base size of Ω_k for comparison.

Theorem 3.2.1 (Valle and Roney-Dougal 2023, Mecenero and Spiga 2023)**.** *Let Ω_k be the set of k-element subsets of $[n]$, and let S_n act on Ω_k naturally. Let $b(n,k)$ denote the base size of $S_n \circlearrowleft \Omega_k$.*

1. *When $n \geq \lfloor (k^2+k)/2 \rfloor + 1$,*

$$b(n,k) = \left\lceil \frac{2n-2}{k+1} \right\rceil.$$

2. *When $n \geq 2k$,*

$$b(n,k) \geq \lceil \log_2 n \rceil,$$

with equality when $n = 2k$.

We noticed from our data that $b(n,k) = \gamma(n,k)$ for n,k such that $b(n,k)$ is closed; here γ is the quantum base size. In fact, we observed the maximum number of parts present in the partitions corresponding to the irreducible representations of $(\mathbb{C}\Omega)^{\otimes \ell}$ for each ℓ and found that the number of parts was growing exponentially by 2^ℓ up to $n = 2k$ and then after some stabilization was growing constantly by a factor of $(\dfrac{k+1}{2})\ell$; when this factor was odd the growth each iteration alternated. Additionally, due to the fact that the maximum number of parts present grew strictly, we found that the conjecture

Conjecture 3.2.2.

$$S^{(1^n)} \in \mathrm{Irr}((\mathbb{C}\Omega_k)^{\otimes \ell}) \implies \mathrm{Irr}(\mathbb{C}[S_n]) = \mathrm{Irr}((\mathbb{C}\Omega_k)^{\otimes \ell}),$$

which is to say that the alternating representation $S^{(1^n)}$ is present in the decomposition of $(\mathbb{C}\Omega)^{\otimes \ell}$ only if every other irreducible representation is in $(\mathbb{C}\Omega)^{\otimes \ell}$.

held true in all our data. Notably, this did not hold for the regular partition module. In the corresponding classical base size case, Valle 2024 found that

Theorem 3.2.3. *Let* sgn *be the sign character of S_n and χ the permutation character of $S_n \circlearrowleft \Omega_k$. Then*

$$b(n,k) = \min\{l \in \mathbb{N} : \langle \mathrm{sgn}, \chi^l \rangle \neq 0\}.$$

Table 3.1: Quantum Query Complexity Of Symmetric Oracle Discrimination for S_n Acting On k-Element Subsets of $[n]$

n\k	1	2	3	4	5	6	7	8	9	10	11
1											
2											
3		2									
4		2	4								
5		3	4	4							
6		4	4	4	5						
7		4	4	3	4	6					
8		5	5	3	4	5	7				
9		6	5	4	4	4	6	8			
10		6	6	4	4	4	5	6	9		
11		7	6	4	4	4	4	5	7	10	
12		8	7	5	4	4	4	5	6	8	
13		8	7	5	5	4	4	5	5	6	
14		9	8	6	5	5	4	5	5	6	
15		10	8	6	5	5	4	4	5	5	
16		10	9	6	5	5	5	4	5	5	
17		11	9	7	6	5	5	5	5	5	
18		12	10	7	6	6	5	5	5	5	
19		12	10	8	6	6	5	5	5	5	
20		13	11	8	7	6	6	5	5	5	
21		14	11	8	7	6	6	5	5	5	
22		14	12	9	7	6	6	6	5	5	
23		15	12	9	8	7	6	6	6	5	
24		16	13	10	8	7	6	6			5
25		16	13	10	8	7	7				
26		17	14	10	9	8					
27			14		9	8					
28			15		9	8					
29					10						
30					10						

In greater generality, a homomorphism $\phi : G \to \{1, -1\}$*, for* $G \leq \mathrm{Sym}(\Omega)$ *is* base-controlling *if for every tuple* A *of points of* Ω*,* A *is a base if and only if* $\phi(G_A) = 1$*. If we consider* $\{1, -1\}$ *as one-dimensional vector space, it becomes clear that* ϕ *is an irreducible representation (a one-dimensional representation cannot be reduced), and also an irreducible character as* $\mathrm{tr}(\phi) = \phi$*. If* G *has permutation character* χ *and admits a base controlling homomorphism* ϕ*, then*

$$b(G) = \min\{l \in \mathbb{N} : \langle \phi, \chi^l \rangle \neq 0\}.$$

which is an equivalent statement formulated in character theoretic language. For an irreducible character $\chi_{(i)}$ of a irreducible module $V^{(i)}$ with multiplicity m_i in a module $V \cong m_1 V^{(1)} \oplus m_2 V^{(2)} \oplus \cdots \oplus m_k V^{(k)}$, we have by Sagan 2001, Cor. 1.9.4 that $\langle \chi, \chi^{(i)} \rangle = m_i$; this is to say that an inner product with an irreducible character is a projection upon the given irreducible character. The equivalence lends further credence to our conjecture that

Conjecture 3.2.4. *For the action* $S_n \circlearrowright \Omega_k$*,*

$$b(n, k) = \gamma(n, k).$$

and leads us to a new (mostly unevidenced) conjecture

Conjecture 3.2.5. *If a permutation group* G *admits a base-controlling homomorphism* ϕ*, then*

$$b(G) = \gamma(G).$$

We now focus on the case where $n = 2k$. For our main positive result we need the following theorem:

Theorem 3.2.6 (Dvir 1993)**.** *Define* $h(S^\lambda) = h(\lambda) = (\lambda^t)_1$*, where* λ^t *is the transpose of* λ*. This is the number of parts in* λ*. Furthermore write* $\lambda \cap \mu = (\min(\lambda_1, \mu_1), \min(\lambda_2, \mu_2), \cdots)$*, so* $|\lambda \cap \mu|$ *is the area of the intersection of the diagrams* λ *and of* μ*. Then*

$$\max\{h(S^\nu) | S^\lambda \in \mathrm{Irr}(S^\lambda \otimes S^\mu)\} = |\lambda \cap \mu^t| \leq h(\lambda) \cdot h(\mu).$$

Theorem 3.2.7. *Let* $S_{2k} \circlearrowright \Omega_k^{2k}$ *naturally where* Ω_k^{2k} *is the set of* k*-element subsets of* $[2k]$*. Then*

$$b(2k, k) = \gamma(2k, k).$$

Proof. By Dvir's theorem we know that

$$h(S^\lambda \otimes S^\mu) \leq h(S^\lambda)h(S^\mu).$$

As ℓ increases by 1 each query,

$$2 \cdot \max\{h(S^\lambda) \in \mathrm{Irr}(\Omega_k^{2k})^{\otimes \ell - 1} \geq \max\{h(S^\lambda) \in \mathrm{Irr}(\Omega_k^{2k})^{\otimes \ell}\}.$$

Because we know the sign representation $S^{(1^{2k})}$ must be in the decomposition in order for the decomposition to contain every irreducible module, it takes at least $\ell = \lceil \log_2 2k \rceil$ queries for $S^{(1^{2k})} \in \mathrm{Irr}((\Omega_k^{2k})^{\otimes \ell})$. We thus have a lower bound

$$\lceil \log_2 2k \rceil = b(2k,k) \leq \gamma(2k,k).$$

Because we can always simulate a classical computer on a quantum computer, we also have the upper bound

$$\gamma(2k,k) \leq b(2k,k).$$

Combining these two statements we see that

$$b(2k,k) = \gamma(2k,k).$$

$\square$

One experiment we did was to count how many partitions were showing up in $(\mathbb{C}\Omega)^{\otimes \ell}$ for each ℓ. We found that at each step where $\max\{h((\mathbb{C}\Omega_k^{2k})^{\otimes \ell})\} = 2^\ell$, that all partitions with $h(\lambda \vdash 2k) \leq 2^\ell$ parts appeared in $(\mathbb{C}\Omega)^{\otimes \ell}$. We thus conjecture:

Conjecture 3.2.8. *Let $n = 2k$, and let Ω_k^n denote the set of k element subsets of $[n]$. Then*

$$\mathrm{Irr}((\mathbb{C}\Omega_k^{2k})^{\otimes \ell}) = \{S^\mu |\ \mu \textit{ has at least } 2k - 2^\ell \textit{ columns}\}.$$

If this conjecture is true, then we can neatly prove the following theorem:

Theorem 3.2.9. *Let $\gamma^{\mathrm{bdd}}(n,k)$ denote the bounded quantum base size for S_n acting on k element subsets of n. Then*

$$\gamma^{\mathrm{bdd}}(2k,k) = \log_2(n - 2\sqrt{n} + \Theta(n^{1/6}))$$

queries are necessary and sufficient to succeed with probability $2/3$, and in fact any probability $1 - \epsilon$ for $\epsilon \in (0,1)$.

Proof. In Copeland and Pommersheim 2021, they prove a similar theorem for the case $k = 1$, first proving that

$$\mathrm{Irr}((\mathbb{C}\Omega_1)^{\otimes \ell}) = \{S^\mu |\ \mu \text{ has at least } n - \ell \text{ columns}\}.$$

Using the Robinson-Schensted correspondence, they prove that the sum of the square of the constituent irreducible representation dimensions corresponds to the number of permutations with longest increasing subsequence at least $n - \ell$. They then employ the Tracy-Widom distribution of the longest increasing subsequence l_n of a random permutation: formally they cite that for cumulative distribution function $F(x)$ of the Tracy-Widom distribution

$$\lim_{n\to\infty} Prob\left(\frac{l_n - 2\sqrt{n}}{n^{1/6}} \leq x\right) = F(x).$$

Because our conjecture is exactly analogous to their proof, we can use this formula in the same manner as they do with $n - 2^\ell$ instead of $n - \ell$. We find that if we use $\ell = \log_2(n - 2\sqrt{n} + cn^{1/6})$ queries for any real number c then

$$Prob(l_n \geq n - 2^\ell) = 1 - Prob(l_n < n - 2^\ell) = 1 - Prob(l_n < n - 2\sqrt{n} + cn^{1/6}) =$$

$$1 - Prob\left(\frac{l_n - 2\sqrt{n}}{n^{1/6}} < -c\right) \to_{n\to\infty} 1 - F(-c).$$

Thus for any $\epsilon \in (0, 1)$, if we wish to succeed with probability $1 - \epsilon$, it will be necessary and sufficient to use $\ell = \log_2(n - 2\sqrt{n} + cn^{1/6})$ queries, where $c = -F^{-1}(\epsilon)$ and we assume n is sufficiently large. □

We end with a weaker conjecture which, if proven, offers an inductive proof of Conjecture 3.2.8.

Conjecture 3.2.10. *Let $G = S_{2k}$, $2k = 2^j$ for some integer j, and $X = \Omega_k^{2k}$. Then*

$$M^{((2k/2^\ell)^{2^\ell})} \subseteq (\mathbb{C}\Omega_k^{2k})^{\otimes \ell}.$$

What we observed is that $S^{(k,k)} \otimes S^{((2k/2^j)^{2^j})}$ would contain the representation $S^{((2k/2^{j+1})^{2^{j+1}})}$. If, on the other hand, $2k$ isn't a power of 2, then for $2k/2^j \notin \mathbb{Z}$, the rows get shorter so that $\sum \lambda_i = 2k$. For example, if $k = 5$, then $S^{(5,5)} \otimes S^{(5,5)}$ contains the partition $(3^2, 2^2)$ instead of (3^4), and likewise $S^{(5,5)} \otimes (S^{(5,5)})^{\otimes 2}$ contains $(2^2, 1^6)$ instead of (2^8). We can thus formulate a more general conjecture that $M^{((2k/2^\ell)^{2^\ell})}$ with enough rows shortened is contained in $(\mathbb{C}\Omega_k^{2k})^{\otimes \ell}$ for any k. With this more general form we could argue that at each step $(\mathbb{C}\Omega_k^{2k})^{\otimes \ell}$ contains all partitions with $h(\lambda) \leq \max 2^\ell, 2k$, which is equivalent to saying that each partition in the decomposition has at least $2k - 2^\ell$ columns.

3.2.2 The Regular Partition Action

The action on regular partitions posed an even greater challenge than the action on k-elements. One of the main issues was that our method of calculating the query complexity quickly became infeasible due to the explosive growth in the number of regular partitions; as such we lacked enough data to have much confidence in a conjecture about its asymptotic behavior. Even the decomposition of the regular partition module proved to be difficult; combinatorially we can calculate that $\mathbb{C}\Omega_p^{a,b}$ has dimension

$$\frac{\binom{ab}{a,a,a,\cdots,a \text{ b times}}}{b!}$$

where $\binom{n}{k_1,k_2,\cdots,k_m} = \dfrac{n!}{k_1!k_2!\cdots k_m!}$ is the multinomial coefficient. This expression comes from the choice of picking a from ab total numbers to be in a given part b times, and then accounting for the equivalence of regular partitions under permuting the b parts. On the other hand, we have by Sagan 2001, Proposition 2.1.11 that

$$\dim(M^\lambda) = \binom{n}{\lambda_1, \lambda_2, \cdots, \lambda_\ell} = n!/\lambda!$$

which counts the number of λ tabloids. The natural choice would be to consider the module $M^{(a^b)}$, but here a factor of $1/b!$ is missing in its dimension. We suspect that no permutation module of S_{ab} is isomorphic to the module formed by our action, which would imply that the decomposition must be done without Young's rule. The behavior of the decomposition of $\mathbb{C}\Omega_p^{a,b}$ we gathered through our data disproves that it is in general isomorphic to some $M^{\lambda \vdash ab}$. The simplest example would be $\mathbb{C}\Omega_p^{3,2}$ which we have computed to decompose as

$$m_1 \, \yng(4,2) \oplus m_2 \, \yng(6)$$

for some positive m_i. We see that $S^{(4,2)}$ is present, so if we assume $\mathbb{C}\Omega_p^{3,2}$ is some M^μ-module then it must be that $\mu \trianglelefteq (4,2)$. Furthermore we can compute that $(4,2) \trianglelefteq (5,2)$, so by transitivity $\mu \trianglelefteq (5,1)$, but then by corollary 2.4.7 in Sagan 2001 the coefficient $m_j S^{(5,1)}$ must be positive. This is a contradiction, so $\mathbb{C}\Omega_p^{3,2}$ cannot be a M^μ-module for any $\mu \vdash ab$. Experimentally we find that the decomposition of a regular partition module always contains the partition $(ab-2,2)$ and not the partition $(ab-1,1)$, so none of them are M^μ-modules. What follows is a list of various observations on our data:

1. In the decomposition of $\mathbb{C}\Omega_p^{a,b}$, b equals the maximum number of parts present in an irreducible representation of the decomposition.

2. If a or b equals 2, then $\mathbb{C}\Omega_p^{a,b}$ decomposes into all the partitions that can be made by adding two blocks in a row at a time to an initially empty partition until there are ab blocks.

3. For $a \geq b$,
$$\mathbb{C}\Omega_p^{a,b} \subseteq \mathbb{C}\Omega_p^{b,a}.$$

Once both a and b are at least 3, the behavior of the decomposition loses a clear pattern and most of the decompositions compute too slowly to collect sufficient evidence for a conjecture. A final interesting remark is that $\mathbb{C}\Omega_p^{b,a}$ does not in general admit sgn as a sign-controlling homomorphism, as

$$S^{(1^{ab})} \in \mathrm{Irr}((\mathbb{C}\Omega_p^{2,4})^{\otimes 2})$$

but

$$\mathrm{Irr}((\mathbb{C}\Omega_p^{2,4})^{\otimes 2}) \subsetneq \mathrm{Irr}(\mathbb{C}[S_{ab}]).$$

In general we conjecture that at least when $a = 2, b = 2j$, then $S^{(1^{ab})} \in \mathrm{Irr}((\mathbb{C}\Omega_p^{a,b})^{\otimes 2})$. This is due to our observation that $S^{(1^{2k})} \in \mathrm{Irr}(S^{(2^k)} \otimes S^{(k,k)})$ in accordance with our last conjecture regarding the k-element action.

Chapter 4

Discussion

We had difficulty attempting to prove our more general conjectures due to the general difficulty of the Kronecker coefficient problem. That being said, Kronecker coefficients can be solved when one makes greatly restrictive assumptions, so it seems like some of these conjectures could be proven by recourse to character theory or algebra. We hope that these conjectures will kick-start further interest in these problems and allow future researchers to jump into the problem quickly.

Many paths lie open for further research of both empirical and theoretical nature. Our general conjecture that $\gamma(G) = b(G)$ when G admits a base-controlling homomorphism could be tested using Cohen De Valle's example of the group $\mathrm{PSL}_2(7) : 2$ acting sharply 3-transitively from Valle 2024, by constructing the group and its action in GAP and seeing if the character/homomorphism $\phi : G \to \{1, -1\}$ with kernel $\mathrm{PSL}_2(7)$ appears with enough multiplicity before every other irreducible character appears in the decomposition of the permutation character. One idea we explored but decided not to pursue was to use the representation theory of the alternating group to analyze the quantum base size of our two actions restricted to A_n. Using wreath products to create groups and actions was also of interest for finding new symmetric oracle discrimination problems but ultimately fell outside of our scope; it would be very interesting to see if Theorem 1.3 concerning the base size of a wreath product could be of use in determining quantum base size. A final curiosity is if the equivalence of quantum and classical base size for the k-element action could help shed some light on the interpolation of exponential and linear growth that occurs as n grows for fixed k; the interpolation can be observed from the growth of the maximum height of partitions in $(\mathbb{C}\Omega_k)^{\otimes \ell}$.

Zajj offered us a method for proving upper and lower bounds for the quantum base size of Ω_k: First, if we let $A = \Omega_k^\ell$, and let $G_a = \mathrm{Stab}(a)$, then if there is some $a \in A$

such that $G_a = 1$, then the linear span of the orbit $S_n \cdot a$ is naturally isomorphic to the left regular module $\mathbb{C}[S_n]$ and a submodule of $\mathbb{C}A$; since the regular module contains every irreducible representation of S_n, so does $\mathbb{C}[S_n] \subseteq \mathbb{C}A$. This gives us an upper bound of queries ℓ. Alternatively, if for every $a \in A$, G_a has the same number of even and odd permutations, and the action of the sign idempotent $p = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma)\sigma$ on each a sums to 0, then by linear extension $p \cdot \mathbb{C}A = 0$ and therefore the sign representation is not in A. This gives a lower bound on $\gamma(G)$ as the sign character must be present for every irreducible to be present in $\mathbb{C}A$. The upper bound method could be helpful to prove that there is a quantum speedup for the regular partition action, although we don't think that this is likely to be true. On the other hand, the lower bound method might offer a way to prove $b(n, k) = \gamma(n, k)$ in general or in a new special case.

Bibliography

Bhattacharjee, Meenaxi et al. (1998). “Wreath Products”. In: *Notes on Infinite Permutation Groups*. Berlin, Heidelberg: Springer, pp. 67–76. ISBN: 978-3-540-49813-1. DOI: `10.1007/BFb0092558`. URL: `https://doi.org/10.1007/BFb0092558` (visited on 10/17/2024).

Bürgisser, Peter and Christian Ikenmeyer (2008). “The Complexity of Computing Kronecker Coefficients”. In: *DMTCS Proceedings*. Ed. by Krattenthaler et al. Vol. DMTCS Proceedings vol. AJ, 20th Annual International Conference on Formal Power Series and Algebraic Combinatorics (FPSAC 2008). DMTCS Proceedings. Viña del Mar, Chile: Discrete Mathematics and Theoretical Computer Science, pp. 357–368. DOI: `10.46298/dmtcs.3622`. URL: `https://inria.hal.science/hal-01185157` (visited on 04/22/2025).

Copeland, Daniel and Jamie Pommersheim (Mar. 3, 2021). *Quantum Query Complexity of Symmetric Oracle Problems*. DOI: `10.48550/arXiv.1812.09428`. arXiv: `1812.09428`. URL: `http://arxiv.org/abs/1812.09428` (visited on 10/17/2024). Pre-published.

Dvir, Y. (Jan. 1993). “On the Kronecker Product of S_n Characters”. In: *Journal of Algebra* 154.1, pp. 125–140. ISSN: 00218693. DOI: `10.1006/jabr.1993.1008`. URL: `https://linkinghub.elsevier.com/retrieve/pii/S0021869383710082` (visited on 04/18/2025).

Fulton, William (1997). *Young Tableaux: With Applications to Representation Theory and Geometry*. Cambridge University Press. 276 pp. ISBN: 978-0-521-56724-4. Google Books: `U9vZal2HCcoC`.

Fulton, William and Joe Harris (2004). *Representation Theory*. Vol. 129. Graduate Texts in Mathematics. New York, NY: Springer. ISBN: 978-3-540-00539-1 978-1-4612-0979-9. DOI: `10.1007/978-1-4612-0979-9`. URL: `http://link.springer.com/10.1007/978-1-4612-0979-9` (visited on 04/10/2025).

GAP – Groups, Algorithms, and Programming, Version 4.14.0 (2024). The GAP˜Group. URL: `https://www.gap-system.org`.

James, Gordon and Martin Liebeck (2001). *Representations and Characters of Groups*. 2nd ed. Cambridge: Cambridge University Press. ISBN: 978-0-521-81205-4. DOI: 10.1017/CBO9780511814532. URL: https://www.cambridge.org/core/books/representations-and-characters-of-groups/9F525E6ACAC7FFADFDBDECE98C115F40 (visited on 10/01/2024).

Mecenero, Giovanni and Pablo Spiga (Aug. 9, 2023). *A Formula for the Base Size of the Symmetric Group in Its Action on Subsets*. Version 2. DOI: 10.48550/arXiv.2308.02337. arXiv: 2308.02337 [math]. URL: http://arxiv.org/abs/2308.02337 (visited on 04/07/2025). Pre-published.

Morris, Joy and Pablo Spiga (Feb. 20, 2021). *On the Base Size of the Symmetric and the Alternating Group Acting on Partitions*. Version 1. DOI: 10.48550/arXiv.2102.10428. arXiv: 2102.10428 [math]. URL: http://arxiv.org/abs/2102.10428 (visited on 04/07/2025). Pre-published.

Nielsen, Michael A. and Isaac L. Chuang (Dec. 8, 2010). *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press. ISBN: 9780511976667. DOI: 10.1017/CBO9780511976667. URL: https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE (visited on 10/17/2024).

Sagan, Bruce E. (2001). *The Symmetric Group*. Vol. 203. Graduate Texts in Mathematics. New York, NY: Springer. ISBN: 978-1-4419-2869-6 978-1-4757-6804-6. DOI: 10.1007/978-1-4757-6804-6. URL: http://link.springer.com/10.1007/978-1-4757-6804-6 (visited on 12/04/2024).

Valle, Coen del (Sept. 23, 2024). *A Character Theoretic Formula for Base Size*. DOI: 10.48550/arXiv.2409.15153. arXiv: 2409.15153. URL: http://arxiv.org/abs/2409.15153 (visited on 10/17/2024). Pre-published.

Valle, Coen del and Colva M. Roney-Dougal (Aug. 8, 2023). *The Base Size of the Symmetric Group Acting on Subsets*. DOI: 10.48550/arXiv.2308.04360. arXiv: 2308.04360 [math]. URL: http://arxiv.org/abs/2308.04360 (visited on 04/17/2025). Pre-published.

Wang, Ruoyu (2021). *Symmetric Functions and Kronecker Coefficients*. URL: https://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-441279 (visited on 04/22/2025).

Zhandry, Mark (Oct. 28, 2015). *Quantum Oracle Classification - The Case of Group Structure*. DOI: 10.48550/arXiv.1510.08352. arXiv: 1510.08352. URL: http://arxiv.org/abs/1510.08352 (visited on 10/17/2024). Pre-published.