Restricting Supercharacters of $U_n(\mathbb{F}_2)$

Set Partition Combinatorics and Colored Hasse Diagrams

Stephen Lewis

An Honors Thesis Submitted to the
Department of Mathematics
University of Colorado

Directed by Professor Nathaniel Thiem

Professor Nathaniel Thiem, Department of Mathematics

Professor Robert Tubbs, Department of Mathematics

Professor Kalyana Mahanthappa, Department of Physics

13 April 2009

Abstract

The character theory of the symmetric group S_n is described beautifully through a striking application of number partitions and tableaux combinatorics. Because $U_n(\mathbb{F}_q)$ plays a role for p-groups similar to the role that S_n plays for finite groups, one might hope for a similar set up in the character theory of $U_n(\mathbb{F}_q)$. However, the character theory of $U_n(\mathbb{F}_q)$ turns out to be wild. Yet, an analogous theory, called supercharacter theory, plays an important role in $U_n(\mathbb{F}_q)$ with set partition combinatorics instead of number partition combinatorics.

In this paper, I will begin with a review of character theory and supercharacter theory, the role of the latter in $U_n(\mathbb{F}_2)$, discuss an algorithm for computation of restriction and the role it may play in theoretical proofs, introduce a role that colored Hasse diagrams can play, and finally address when the coefficient of the trivial character in a restriction is nonzero. Throughout this paper, I will recall the similarities and differences of S_n to $U_n(\mathbb{F}_q)$, and character theory to supercharacter theory.

1 Introduction

For the finite field of q elements $\mathbb{F}_q$, the group $U_n(\mathbb{F}_q)$ is defined by

$$U_n(\mathbb{F}_q) = \left\{ u = \begin{pmatrix} 1 & * & * & \dots & * \\ 0 & 1 & * & \dots & * \\ 0 & 0 & 1 & & * \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} : u \text{ is } n \times n, * \in \mathbb{F}_q \right\}.$$

Just as any finite group can be embedded into some S_n, any p-group can be embedded into some $U_n(\mathbb{F}_q)$. In the character theory of S_n, number partitions and tableaux combinatorics describe the character theory in a graceful way. That is, the conjugacy classes and the irreducible characters of S_n can be indexed by partitions of n, and the action of S_n on a module, tensor products of characters, and restriction of characters may all be understood through the tableaux combinatorics. While the hope may be for a similar setup in the character theory of $U_n(\mathbb{F}_q)$, the character theory turns out to be wild. Recently, however, an analogous theory called supercharacter theory displays that in $U_n(\mathbb{F}_q)$, set partitions of $\{1, 2, \dots, n\}$ play a similar role to number partitions in S_n in character theory.

In the following section, I will address preliminary knowledge, including basic character theory, general supercharacter theory, construction of a supercharacter theory of $U_n(\mathbb{F}_q)$, and how set partitions of $\{1, 2, \dots, n\}$ index the supercharacters of $U_n(\mathbb{F}_2)$ in a natural way which is not only computable, but aesthetically pleasing and combinatorially natural. Although these constructions extend naturally to $U_n(\mathbb{F}_q)$, I will leave these combinatorics alone in this paper. In the next section, I will discuss an algorithm for restriction of these supercharacters, beginning with a theoretical construction which lends itself well to proofs, and a concrete implementation which lends itself well to computation. In the final section, I will build colored partial orders out of the set partitions of $\{1, 2, \dots, n\}$, and use these to give a precise classification for when the coefficient of the trivial character is nonzero in a restriction of these supercharacters.

2 Preliminaries

In this section, I will review necessary parts of character theory, define and introduce some general properties of supercharacter theories, construct a supercharacter theory on $U_n(\mathbb{F}_q)$, and give some of the previous work done in this theory.

2.1 Character Theory

Characters define modules up to isomorphism. To understand the character theory of a group, we need to understand primarily the irreducible characters and the conjugacy classes. In this section, I review some techniques of character theory.

First, there is a natural inner product on characters of G given by

$$\begin{aligned}\langle \chi, \psi \rangle_G &= \dim\left(\mathrm{Hom}_G(V_\chi, V_\psi)\right) \\ &= \frac{1}{|G|}\sum_{g\in G} \chi(g)\overline{\psi(g)}\end{aligned}$$

where V_χ and V_ψ are the modules defined (up to isomorphism) by χ and ψ.

Let G be a group and $H \leqslant G$ be a subgroup. Restriction of characters maps characters of G to characters of H in the following way.

$$\begin{array}{ccc} \{\text{characters of } G\} & \to & \{\text{characters of } H\} \\ \chi & \mapsto & \mathrm{Res}^G_H(\chi): \begin{array}{ccc} H & \to & \mathbb{C} \\ h & \mapsto & \chi(h) \end{array} \end{array}.$$

In other words, restriction of a character χ of a group G to a subgroup H gives a character denoted $\mathrm{Res}^G_H(\chi)$ which is the same character as χ, except with a restricted domain.

There is also a natural notion of induction from the subgroup H up to G. Perhaps most cleanly, it can be defined as the adjoint functor to restriction. That is, induction is defined by the relation

$$\left\langle \chi, \mathrm{Res}^G_H(\psi)\right\rangle_H = \left\langle \mathrm{Ind}^G_H(\chi), \psi\right\rangle_G .$$

Given two characters of G, χ and ψ, their tensor product denoted $\chi \otimes \psi$ is the character associated to the module tensor product $V_\chi \otimes V_\psi$. Because the tensor product of modules is associative, commutative, and distributive over direct sum (all up to a natural isomorphism), the tensor product of characters is associative, commutative, and distributive over addition. Another well known fact which I will use frequently is that restriction and tensor products commute. See, for example, [17].

Proposition 2.1. *If χ and ψ are characters of G, and $H \leqslant G$, then*

$$\mathrm{Res}^G_H(\chi \otimes \psi) = \mathrm{Res}^G_H(\chi) \otimes \mathrm{Res}^G_H(\psi).$$

2.2 Supercharacter Theory

The idea of a supercharacter theory was initially motivated by work done in $U_n(\mathbb{F}_q)$ by André and Yan. Although the character theory of $U_n(\mathbb{F}_q)$ is wild, a slightly less precise version of the character theory led to a computable theory. A character theory can be viewed abstractly as the partition of G into it's conjugacy classes $\mathrm{Cl}(G)$ and a set of irreducible characters $\mathrm{Irr}(G)$. With this in mind, I give the general definition of a supercharacter theory first introduced by Persi Diaconis and Martin Isaacs [11].

Given a group G, a *supercharacter theory* of G is a set of superclasses $\mathcal{K}$ and a set of supercharacters $\mathcal{X}$, such that

(a) the set $\mathcal{K}$ is a partition of G, and each part is a union of conjugacy classes,

(b) the set $\mathcal{X}$ is a set of characters such that each irreducible character appears as a nonzero summand of exactly one supercharacter,

(c) for $\chi \in \mathcal{X}$ and $x, y \in K \in \mathcal{K}$, $\chi(x) = \chi(y)$,

(d) $|\mathcal{K}| = |\mathcal{X}|$,

(e) the identity element 1 of G is in its own superclass, and the trivial character $\mathbb{1}$ of G is a supercharacter.

Notice, for example, that $\mathcal{K} = \mathrm{Cl}(G)$ and $\mathcal{X} = \mathrm{Irr}(G)$ define a supercharacter theory on G. This is the finest supercharacter theory, as it is the character theory. However, often supercharacter theories are coarser. In general, supercharacter theories seem to be finicky, and classifying all supercharacter theories of a given group is still an open question.

Let $\hat{\mathcal{X}}$ denote an indexing set of the supercharacters. In any supercharacter theory coarser than $\mathcal{K} = \mathrm{Cl}(G)$ and $\mathcal{X} = \mathrm{Irr}(G)$, the only characters which can be computed are the characters χ which decompose as

$$\chi = \sum_{\gamma \in \hat{\mathcal{X}}} c^{\gamma} \chi^{\gamma}$$

for $c^{\gamma} \in \mathbb{Z}_{\geqslant 0}$. On the other hand, the theory does give a complete description of these characters. A supercharacter theory is often very good approximation to a character theory. For example, in the supercharacter theory which we will investigate in $U_n(\mathbb{F}_q)$, there is a very simple criterion for determining whether a given supercharacter is an irreducible character. So in addition to approximating the character theory, one gets a slice of the actual character theory along the way.

Supercharacters are also an orthogonal set of vectors in the character space. This can be seen easily by the fact that if χ and ψ are irreducible, then

$$\langle \chi, \psi \rangle_G = \delta_{\chi\psi}$$

and that supercharacters are sums of disjoint irreducible characters. Notice, however, that a supercharacter χ is normal (in the sense that $\langle \chi, \chi \rangle = 1$) if and only if it is irreducible.

2.3 Construction of Supermodules and Superclasses of $U_n(\mathbb{F}_q)$

In this section I give the construction of a supercharacter theory of $U_n(\mathbb{F}_q)$ first introduced by André and Yan [1]. I will begin the construction with some technical necessities. Let the ring $\mathfrak{n}$ be defined as

$$\mathfrak{n} = U_n(\mathbb{F}_q) - 1 = \left\{ \eta = \begin{pmatrix} 0 & * & * & \dots & * \\ 0 & 0 & * & \dots & * \\ 0 & 0 & 0 & & * \\ \vdots & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix} : \eta \text{ is } n \times n, * \in \mathbb{F}_q \right\}.$$

Considered as a vector space, the dual space of $\mathfrak{n}$ is $\mathfrak{n}^*$, given by

$$\mathfrak{n}^* = \left\{ \lambda : \mathfrak{n} \to \mathbb{F}_q \middle| \lambda \text{ is } \mathbb{F}_q\text{-linear} \right\}.$$

There is a natural two-sided action of $U_n(\mathbb{F}_q)$ on $\mathfrak{n}$ given by matrix multiplication. There is also a natural two sided action of $U_n(\mathbb{F}_q)$ on $\mathfrak{n}^*$. For $u, v \in U_n(\mathbb{F}_q), x-1 \in \mathfrak{n}, \lambda \in \mathfrak{n}^*$, $u\lambda v$ is the function given by

$$(u\lambda v)(x-1) = \lambda(u^{-1}(x-1)v^{-1}).$$

The rings $\mathfrak{n}$ and $\mathfrak{n}^*$ with $U_n(\mathbb{F}_q)$ actions then give the modules and partition of $U_n(\mathbb{F}_q)$ which will serve as a supercharacter theory. For $\lambda \in \mathfrak{n}^*$, define the vector space V^λ as

$$V^\lambda = \mathbb{C}\text{-span}\{v_\mu \big| \mu \in U_n(\mathbb{F}_q)\lambda\},$$

where the vectors of V^λ are formal sums of the vectors v_μ over $\mathbb{C}$, and $U_n(\mathbb{F}_q)\lambda$ is the set of all one-sided actions of $U_n(\mathbb{F}_q)$ on λ. That is,

$$U_n(\mathbb{F}_q)\lambda = \left\{ \mu \in \mathfrak{n}^* \middle| \mu = u\lambda \text{ for some } u \in U_n(\mathbb{F}_q) \right\}.$$

This vector space is made into a $U_n(\mathbb{F}_q)$-module by first fixing some nontrivial group homomorphism $\vartheta : \mathbb{F}_q^+ \to \mathbb{C}^\times$ from the additive group of $\mathbb{F}_q$ to the multiplicative group of $\mathbb{C}$. Then $U_n(\mathbb{F}_q)$ acts on V^λ by

$$uv_\mu = \vartheta \circ \mu(u^{-1} - 1)v_{u\mu}.$$

Through this construction, the modules are indexed by the elements of the dual space. However, many of these modules are isomorphic. It turns out that

$$V^\lambda \cong V^\mu \text{ if and only if } \lambda \in U_n \mu U_n = \{u\mu v \big| u, v \in U_n\} \text{ [11]}.$$

In other words, two modules are isomorphic when λ and μ are in the same two-sided orbits of $U_n(\mathbb{F}_q)$ on $\mathfrak{n}^*$. The characters of these modules will be the supercharacters of our theory. I will return to this fact later, and re-index the modules and their characters by new objects (set partitions of $\{1, 2, \ldots, n\}$) to get a unique characterization.

Partition $U_n(\mathbb{F}_q)$ in the following way; let $u, v \in U_n(\mathbb{F}_q)$ be in the same part if there are $x, y \in U_n(\mathbb{F}_q)$ such that

$$x(u-1)y = v-1.$$

In other words, u and v are in the same part if $u-1, v-1$ are in the same two sided orbits of $U_n(\mathbb{F}_q)$ on $\mathfrak{n}$. In Section 2.5, I will characterize the two sided orbits of $\mathfrak{n}^*$ and $\mathfrak{n}$ so that we may understand the constructed modules and partition, but first I introduce some helpful combinatorics.

2.4 Arcs and Set Partitions of $\{1, 2, \ldots, n\}$

I have been using the word partition of a set in the normal sense of the word: a collection of pairwise disjoint sets which union to the whole set. But combinatorially, it will be useful to have a slightly different notion of a set partition of $\{1, 2, \ldots, n\}$. This is analogous to the representation theory of S_n, where a number partition of n is best thought of as n boxes stacked into a corner instead of a weakly decreasing tuple $(i_1, i_2, \ldots, i_k)$ where $n = \sum_{j=1}^{k} i_j$.

An *arc* $i \frown l$ of $\{1, 2, \ldots, n\}$ is an ordered pair $(i, l) \in \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$ such that $i < l$. To indicate two arcs $i \frown l$ and $l \frown k$, I may write $i \frown l \frown k$, a third arc $k \frown j$ by $i \frown l \frown k \frown j$, etc... Two arcs $i \frown l$ and $j \frown k$ are said to *conflict* (or have a conflict) if $i = j$ or $l = k$ (or both).

A *set partition* μ of $\{1, 2, \ldots, n\}$ is a set of arcs of $\{1, 2, \ldots, n\}$ such that no arcs in μ conflict. In other words, a set of arcs is a set partition if no distinct arcs share one endpoint. With abusive terminology, I will call a multi set M whose members are arcs of $\{1, 2, \ldots, n\}$ a set partition if no arcs conflict (with the association that an arc will conflict with itself it appears twice).

It should be noted that this notion of set partition agrees with the more standard notion of partition. It is easy to see that by ordering the elements of each part $P_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,m}\}$ so that $x_{i,1} < x_{i,2} < \cdots, < x_{i,m}$, that the map

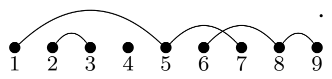
$$\{P_1, \cdots, P_k\} \leftrightarrow \{x_{1,1} \frown x_{1,2} \frown \cdots \frown x_{1,m_1}, \cdots, x_{k,1} \frown x_{k,2} \frown \cdots \frown x_{k,m_k}\}$$

is a bijection between partitions of $\{1, 2, \ldots, n\}$ in the traditional sense and set partitions of $\{1, 2, \ldots, n\}$ in the sense introduced here.

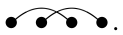
I will often indicate a set or multi-set of arcs (often, but not always, a set partition) μ of $\{1, 2, \ldots, n\}$ by a diagram with n vertices and an arc going between two vertices i and l if $i \frown l \in \mu$. For example, the set partition of $\{1, 2, \ldots, 9\}$ given by

$$\{1 \frown 5 \frown 7, 2 \frown 3, 6 \frown 8 \frown 9\} \longleftrightarrow \{1, 5, 7\} \cup \{2, 3\} \cup \{4\} \cup \{6, 8, 9\}$$

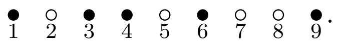
would be indicated by



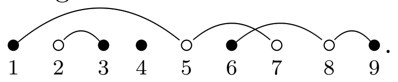
A *crossing* of a set partition μ is a pair of arcs $i \frown l, j \frown k$ such that $i < j < l < k$. In the partition above, $5 \frown 7, 6 \frown 8$ is the only crossing. Diagrammatically, crossings look like



To indicate a subset $S \subset \{1, 2, \ldots, n\}$, I may use n dots in a row, where the i^{th} dot is colored black if $i \in S$ or white if $i \notin S$. For example, the subset $\{1, 3, 4, 6, 9\} \subset \{1, 2, \ldots, 9\}$ may be indicated by



To indicate both a set or multi set of arcs (usually, but not necessarily a set partition) and a subset S, I may just superimpose the two diagrams. For example, with S and μ as before, I may use the diagram



In the rest of this paper, I will tend to not write the numbers underneath the diagram. This is both to save space, and to emphasize the structure of the arcs over the numbers they indicate.

2.5 Supercharacter Theory of $U_n(\mathbb{F}_2)$

In this section I review a better description of the modules and partition given before. First, I will build the two-sided orbits of $\mathfrak{n}$ and $\mathfrak{n}^*$ in $U_n(\mathbb{F}_2)$ and show that they are each indexed by set partitions of $\{1, 2, \ldots, n\}$, use this to index the characters and partition given above, and give some basic computations of this supercharacter theory. It should be noted that the indexing and combinatorics extend naturally to $U_n(\mathbb{F}_q)$, with slightly more finicky combinatorics; see André [1]. For convenience of notation, I will use the convention that $U_n = U_n(\mathbb{F}_2)$ for the rest of this paper, writing explicitly $U_n(\mathbb{F}_q)$ when intending a more general q.

First, let's index the partition of U_n. Recall that I defined the partition $\mathcal{K}$ by u and v are in the same part of $\mathcal{K}$ if and only if there are $x, y \in U_n$ such that

$$x(u-1)y = v - 1.$$

Notice that multiplication on the left by $x \in U_n$ corresponds to row reducing upward in the matrix, and that multiplication on the right by $y \in U_n$ corresponds to row reducing to the right. Since $u-1, v-1 \in \mathfrak{n}$ have zeros along the main diagonal, the entries furthest down and to the left then will determine the two sided orbits. For example, consider the matrix

$$u - 1 = \begin{pmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Because the two sided orbits include any row reduction up or right on this matrix, all matrices of the form

$$u' - 1 = \begin{pmatrix} 0 & 0 & 1 & d & c \\ 0 & 0 & 0 & 1 & b \\ 0 & 0 & 0 & 0 & a \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

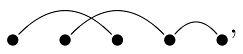
are in the orbit of this matrix. Similarly, for the matrix

$$v - 1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

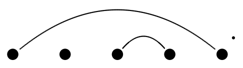
the two sided orbit would consist of all matrices of the form

$$v' - 1 = \begin{pmatrix} 0 & 0 & 0 & b & bc+1 \\ 0 & 0 & 0 & a & ac \\ 0 & 0 & 0 & 1 & c \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

These orbits then suggest indexing by the outer most nonzero entries of the matrix. A natural way of doing so is to use a set of arcs where an arc $i \frown l$ indicates that there is a 1 in the i, l entry of the matrix, and no 1 to left in the ith row and no 1 below it in the lth column. For example, with u as before, the orbit of $u - 1$ would be indicated by

,

and with v as before, the orbit of $v - 1$ would be indicated by

.

These examples show why and how, in general, set partitions index the superclasses. For a more rigorous construction, see [21].

A similar trick will uniquely index the supermodules (and so the supercharacters). Recall that

$$V^\lambda \cong V^\mu \text{ if and only if } \lambda \in U_n \mu U_n,$$

where the action of U_n on $\mathfrak{n}^*$ is for $u, v, x \in U_n$,

$$(u\lambda v)(x-1) = \lambda(u^{-1}(x-1)v^{-1}).$$

Identify a function λ with the matrix (a_{ij}) where $a_{ij} = \lambda(e_{ij})$, and e_{ij} is the $n \times n$ matrix with a 1 in the ith row and jth column, and 0 in every other entry. The question is then the same question as before in disguise, and again one can index the orbits by set partitions of $\{1, 2, \ldots, n\}$ in the same way.

So for any set partition μ of $\{1, 2, \ldots, n\}$, I will uniquely define the character χ^μ as the character of any module V^λ where the outermost nonzero entries of λ are given by the arcs of μ.

It should be noted that these constructions hold almost exactly for $U_n(\mathbb{F}_q)$, except that the nonzero entries which are furthest left and down may be one of many possibilities. So the arcs must be labeled by specific entries, and the superclasses and supercharacters are indexed by set partitions with $\mathbb{F}_q^+$ labels on the arcs (such as $i \overset{a}{\frown} l$).

For example, since $\gamma = \emptyset$ has no conflicts, it is the trivial partition of $\{1, 2, \ldots, n\}$. Then χ^γ is the character of the module V^λ where λ is the trivial function $\lambda(n) = 0$ for all $n \in \mathfrak{n}$, so

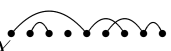
$$\chi^\gamma = \chi^{\bullet\,\bullet\,\cdots\,\bullet} = \mathbb{1},$$

the trivial character.

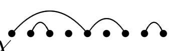
The following is a well known fact (for example, see [13]).

Proposition 2.2. *Let μ be a set partition of $\{1, 2, \ldots, n\}$. Then the supercharacter χ^μ is irreducible if and only if there are no crossings in μ.*

For example, the supercharacter

χ 

is not an irreducible character, but

χ 

is an irreducible character

For $S \subseteq \{1, 2, \ldots, n\}$, define the subgroup $U_S \leqslant U_n$ by

$$U_S = \{u \in U_n \big| u_{ij} \neq 0 \text{ only if } i, j \in S\}.$$

This subgroup is analogous to the subgroup S_T of S_n, $T \subset \{1, 2, \ldots, n\}$, given by

$$\begin{aligned} S_T &= \{w \in S_n \big| w(i) = j \neq i \text{ only if } i, j \in T\} \\ &= \{w \in S_n \big| w(i) = i \text{ for all } i \notin T\}. \end{aligned}$$

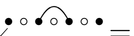
Let $m \leq n$. Then for any $S \subset \{1, 2, \ldots, n\}$ with $|S| = m$, $U_S \cong U_m$. In other words, the subsets S of size m give the natural ways in which U_m embeds into U_n, the same way that subsets T of size m give the natural ways that S_m embeds into S_n. Restriction from U_n to U_S of a character indexed by a single arc set partition can be computed with the following theorem by Thiem [19].

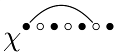
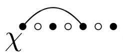
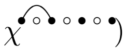
Theorem 2.3. *Let $S \subseteq \{1, 2, \ldots, n\}$. Then*

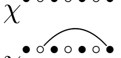
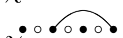
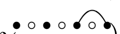
$$\mathrm{Res}^{U_n}_{U_S}(\chi^{i\frown l}) = \begin{cases} 2^{|\{i<k<l \mid k\notin S\}|}\chi^{i\frown l}, & \text{if } i, l \in S,\\ 2^{|\{i<k<l \mid k\notin S\}|}\left(\mathbb{1} + \displaystyle\sum_{i<j<l,\, j\in S} \chi^{j\frown l}\right), & \text{if } i \notin S, l \in S,\\ 2^{|\{i<k<l \mid k\notin S\}|}\left(\mathbb{1} + \displaystyle\sum_{i<k<l,\, k\in S} \chi^{i\frown k}\right), & \text{if } i \in S, l \notin S,\\ 2^{|\{i<k'<l \mid k'\notin S\}|}\left((|S\cap\{i,\ldots,l\}|+1)\mathbb{1} + \displaystyle\sum_{\substack{i<j'<k'<l\\ j',k'\in S}} \chi^{j'\frown k'}\right), & \text{if } i, l \notin S. \end{cases}$$

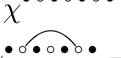
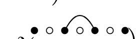
Diagrammatically, restriction can be thought of as removing the dots not in S, and resolving any arcs with an endpoint on a removed dot.

Example 1. Fix $n = 7$, and $S = \{1, 3, 5, 7\} \subset \{1, 2, \ldots, 7\}$. Then,

$\mathrm{Res}^{U_n}_{U_S}\left(\chi^{3\frown 5}\right) = 2^{|\{4\}|}\chi$  $= 2\chi^{3\frown 5}$

$\mathrm{Res}^{U_n}_{U_S}\left(\chi^{1\frown 6}\right) = 2^{|\{2,4\}|}\chi$  $= 4\,(\mathbb{1} + \chi$  $+ \chi$  $)$

$\mathrm{Res}^{U_n}_{U_S}\left(\chi^{2\frown 7}\right) = 2^{|\{2,4\}|}\chi$  $= 4\,(\mathbb{1} + \chi$  $+ \chi$  $)$

$\mathrm{Res}^{U_n}_{U_S}\left(\chi^{2\frown 6}\right) = 2^{|\{4\}|}\chi$  $= 2\,((|S\cap\{3,4,5\}|+1)\mathbb{1} + \chi$  $)$

The tensor product of two supercharacters defined by a single arc was computed by Yan [21].

Theorem 2.4. *For $i < k$, $j < l$ and $\{i,k\} \neq \{j,l\}$,*

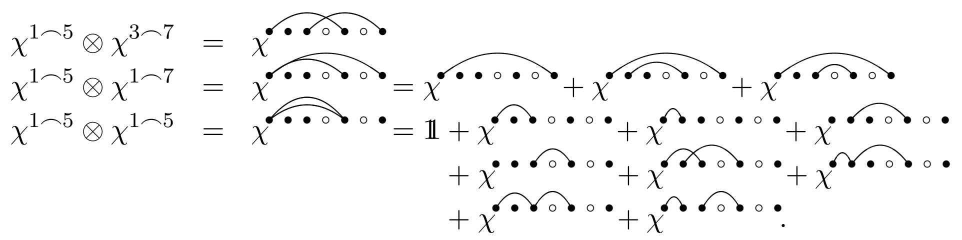
$$\chi^{i\frown k}\otimes\chi^{j\frown l} = \begin{cases} \chi^{\{i\frown k, j\frown l\}}, & \text{if } \{i,k\}\cap\{j,l\} = \emptyset,\\ \chi^{i\frown j\frown l}, & \text{if } i<j=k<l,\\ \chi^{i\frown l} + \displaystyle\sum_{i<j'<k} \chi^{\{j'\frown k, i\frown l\}}, & \text{if } i=j<k<l,\\ \chi^{i\frown l} + \displaystyle\sum_{j<k'<l} \chi^{\{i\frown l, j\frown k'\}}, & \text{if } i<j<k=l, \end{cases}$$

Or, if $\{i,k\} = \{j,l\}$, consider the following case. For $i < l$,

$$\chi^{i\frown l}\otimes\chi^{i\frown l} = \mathbb{1} + \sum_{i<j'<l}\chi^{i\frown j'} + \sum_{i<k'<l}\chi^{k'\frown l} + \sum_{i<j',k'<l}\chi^{\{i\frown j', k'\frown l\}}.$$

Diagrammatically, the tensor product of two supercharacters can be seen as a superposition of their arcs, where conflicts are resolved by the rules above.

Example 2. Fix $n = 7$, and $S = \{1, 2, 3, 5, 7\} \subset \{1, 2, \ldots, 7\}$. Then in U_S,



The previous theorems have direct analogues in $U_n(\mathbb{F}_q)$, and so the combinatorics of $U_n(\mathbb{F}_2)$ are a good stepping stone for the generalized combinatorics of $U_n(\mathbb{F}_q)$.

An immediate consequence of this theorem is that tensor products commute with disjoint unions. Or more precisely:

Corollary 2.5. *Let μ be a set partition of $\{1, 2, \ldots, n\}$. If ν and γ are set partitions of $\{1, 2, \ldots, n\}$ such that $\nu \cup \gamma = \mu$ and $\nu \cap \gamma = \emptyset$, then*

$$\chi^{\mu} = \chi^{\nu} \otimes \chi^{\gamma}.$$

Recall that restriction commutes with tensor products. These facts suggest a method for computing the restriction of supercharacters.

3 An Algorithm For Restriction

In this section I discuss computing restriction in the supercharacter theory of $U_n = U_n(\mathbb{F}_2)$. Restriction of a supercharacter defined by a single arc is done with with Theorem 2.3, and the tensor product of two supercharacters each defined by a single arc is done with Theorem 2.4. The general method of the following implementation is the natural one. That is, if $\mu = \{i_1 \frown j_1, i_2 \frown j_2, \cdots, i_m \frown j_m\}$, then we write

$$\mathrm{Res}^{U_n}_{U_S}(\chi^{\mu}) = \mathrm{Res}^{U_n}_{U_S}(\chi^{i_1 \frown j_1}) \otimes \mathrm{Res}^{U_n}_{U_S}(\chi^{i_2 \frown j_2}) \otimes \cdots \otimes \mathrm{Res}^{U_n}_{U_S}(\chi^{i_m \frown j_m})$$

and we compute each restriction individually, then compute the tensor products iteratively. However, because in general restriction of each $\chi^{i \frown l}$ may produce many copies of the same $\chi^{j \frown k}$, and the necessary tensor products may create many copies of the same χ^{γ}, these conflicts may propagate many times over. So a general implementation must be recursive.

The first section details a theoretical construction of the algorithm which will be useful in theoretical proofs, and which leads intuitively to the implementation. The following sections detail the implementation of the algorithm, and are not important in the theoretical usage (though they may lend intuition if the construction seems obtuse). The function has been implemented in Python, which is a fairly intuitive language, and is compatible with SAGE open source mathematics software.

3.1 A Theoretical Construction

First, it is useful to review multi-sets for this construction. A multi-set M can be thought of as a "set" where repeated elements must be counted repeated times. For example, as sets,

$$\{1,2,2,3,3,3,4,6,6\} = \{1,2,3,4,6\},$$

but as multi sets they are distinct. A union of multi-sets M and N is the multi set $M \cup N$, where an element $m \in M \cup N$ appears the amount that it does in M plus the amount that it does in N. For example, the multi set union of $\{1,2,2,4,5\}$ with $\{1,1,2,3,5\}$ is

$$\{1,2,2,4,5\} \cup \{1,1,2,3,5\} = \{1,1,1,2,2,2,3,4,5,5\}.$$

Also, I will often make the following association implicitly that the coefficient $c_\mu^\gamma \in \mathbb{Z}_{\geqslant 0}$ is defined by

$$\operatorname{Res}_{U_S}^{U_n}(\chi^\mu) = \sum_{\gamma \in \hat{\mathcal{X}}} c_\mu^\gamma \chi^\gamma.$$

The algorithm to compute $\operatorname{Res}_{U_S}^{U_n}(\chi^\mu)$ begins with two pieces of information;

$$\mu = \{i_1 \frown j_1, i_2 \frown j_2, \cdots, i_m \frown j_m\}$$

a partition of $\{1,2,\ldots,n\}$, and $S \subset \{1,2,\ldots,n\}$ a subset. By manipulating,

$$\begin{aligned}\operatorname{Res}_{U_S}^{U_n}(\chi^\mu) &= \bigotimes_{j\frown k \in \mu} \operatorname{Res}_{U_S}^{U_n}\left(\chi^{j\frown k}\right) = \bigotimes_{j\frown k\in\mu} \sum_{\gamma\in\hat{\mathcal{X}}} c_{j\frown k}^\gamma \chi^\gamma \\ &= \sum_{\Gamma = \bigcup\{\gamma_1,\gamma_2,\ldots,\gamma_m\}} C_\Gamma \bigotimes_{\gamma_i\in\Gamma} \chi^{\gamma_i},\end{aligned}$$

where each Γ is a multi set union obtained from selecting one $c_\mu^\gamma \chi^\gamma$ from each $\operatorname{Res}_{U_S}^{U_n}(\chi^\mu) = \sum_\gamma c_\mu^\gamma \chi^\gamma$ in the tensor product, and $C_\Gamma = \Pi_{\gamma\in\Gamma} c_\mu^\gamma$.

Then consider a sequence of multi sets

$$\Gamma = M_0, M_1, \cdots, M_k = \nu$$

such that $M_k = \nu$ is a set partition of $\{1,2,\ldots,n\}$ and for $i \in \{1,2,\ldots,k\}$

$$M_i = \big(M_{i-1} - \{j \frown k, j' \frown k'\}\big) \cup \delta$$

where

1. $\chi^{j\frown k} \otimes \chi^{j'\frown k'}\big|_{\chi^\delta} \neq 0$
2. $j \frown k, j' \frown k'$ conflict.

In other words, the process is to start by selecting a χ^{γ_i} from each restriction, and looking at Γ as a multi set of arcs. In general, this will not necessarily form a set partition. To try and resolve this, select two arcs $i \frown l$ and $j \frown k$ which conflict, and compute the tensor product $\chi^{i\frown l} \otimes \chi^{j\frown k} = \sum_{\delta\in\hat{\mathcal{X}}} c_\delta \chi^\delta$. Then select a δ with $c_\delta \neq 0$ and create the multi set union $(\Gamma - \{i \frown l, j \frown k\}) \cup \delta$. In general, however, this will not be a set partition, so we repeat the process until we end up at a set partition M_k. Note that at every stage, there are potentially several δ which could be chosen. So once a sequence terminates at M_k, back up to M_{k-1} and choose another possible δ. In other words, follow a sequence until it terminates, back up one step and choose a different delta, follow that sequence until it terminates, ...

The final answer is obtained by the sum

$$\sum_{\Gamma}\sum_{\nu=M_k} C_\Gamma \chi^\nu,$$

where the by the indexing in the last sum $\nu = M_k$, I mean a term in the sum for each multi-set sequence ending at ν and beginning at Γ.

In summary, the algorithm is:

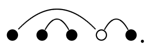
0. Construct each Γ. Choose one.

1. Follow a sequence $\Gamma = M_0, M_1, \ldots, M_k = \nu$ as described before.

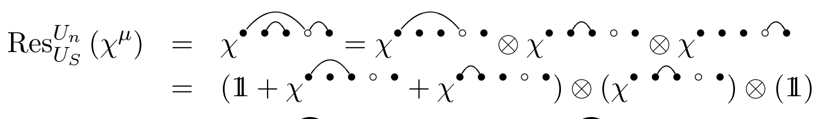
2. Back up to the last multi-set M_i where there are δ which have not been incorporated in a sequence, and choose a δ. Go back to 1. with $\Gamma' = M_i \cup \delta$. If none exist, choose another Γ not yet used. If none exist, go to step 3.

3. Sum all of the ν obtained weighted by the C_Γ to get a final answer.

Example 3. Let $S = \{1, 2, 3, 5\} \subset \{1, 2, 3, 4, 5\}$ and let

$$\mu = \{1 \frown 4 \frown 5, 2 \frown 3\} =$$ .

Then



$$= \bigotimes_{\gamma\in\Gamma=\emptyset\cup\{2\frown 3\}\cup\emptyset} \chi^\gamma + \bigotimes_{\gamma\in\Gamma=\{1\frown 3\}\cup\{2\frown 3\}\cup\emptyset} \chi^\gamma + \bigotimes_{\gamma\in\Gamma=\{1\frown 2\}\cup\{2\frown 3\}\cup\emptyset} \chi^\gamma$$

In this simple example, each $C_\Gamma = 1$ and the sequences are very short. In the first term where $\Gamma = \{2 \frown 3\}$ there is one sequence $\Gamma = M_0 = \nu$. In the last term where $\Gamma = \{1 \frown 2 \frown 3\}$ there is also only one sequence $\Gamma = M_0 = \nu$. In both of these sequences, Γ was a set partition. However, in the middle term $\Gamma = \{1 \frown 3, 2 \frown 3\}$

is not a set partition. The only choice for conflicting arcs is $1 \frown 3, 2 \frown 3$. First, we compute

$$\chi^{\overparen{\bullet\,\overparen{\bullet\,\bullet}}\,\circ\,\bullet} = \chi^{\overparen{\bullet\,\bullet\,\bullet}\,\circ\,\bullet}$$

so we construct each possible M_1 iteratively from each δ (there is, again, only one). So for $\delta = \{1 \frown 3\}$,

$$M_1 = (\Gamma - \{1 \frown 3, 2 \frown 3\}) \cup \{1 \frown 3\} = \{1 \frown 3\}.$$

So M_1 is a set partition, and the sequence terminates. Since there are no other terms χ^δ, in the sequence above, we have computed all sequences and our answer is the sum of the final terms of all sequences, weighted by the C_Γ (all equal to 1),

$$\mathrm{Res}_{U_S}^{U_n}(\chi^\mu) = \chi^{\bullet\,\overparen{\bullet\,\bullet}\,\circ\,\bullet} + \chi^{\overparen{\bullet\,\bullet\,\bullet}\,\circ\,\bullet} + \chi^{\overparen{\bullet\,\bullet}\overparen{\,\bullet}\,\circ\,\bullet}.$$

3.2 Helper Functions

The following functions have been added as helpers, and allow cleaner statements. The `addX` function takes as input a key value (in this case a set partition) `key`, a dictionary `X`, and a value to store (in this case a coefficient) value. The function simply adds `num` to the existing value for `X[key]` if it exists. Otherwise, it creates the key value `key` and assigns `X[key] = num`.

```
def addX(key, X, num):
    if key in X:
        X[key] += num
    else:
        X[key] = num
```

The function `dictord` takes as inputs numbers `i,j,k,l`, and returns a dictionary ordering of the pairs `(i,j)` and `(k,l)` as a tuple `((a,b),(c,d))`. For example, `dictord(2,4,2,3)` returns `((2,3),(2,4))`.

```
def dictord(i, j, k, l):
    if i < k:
        return ((i,j),(k,l))
    if k < i:
        return ((k,l),(i,j))
    if j < l:
        return ((i, j),(k,l))
    return ((k,l),(i,j))
```

3.3 Set Partitions and Arcs

The function `partsToArcs` is a tool for changing a partition (in the traditional sense) of $\{1, 2, \ldots, n\}$ into a set partition of $\{1, 2, \ldots, n\}$. It takes as input a list `parts`, whose elements are lists, which are the ordered parts of the set partition (note that singleton parts need not be included). The function first creates a dictionary called `arcs`. Then, it iterates through the parts `x` in `parts`, and for each `x`, it iterates through the values in `x` storing the first value as `a` and throwing it away, storing the new first value as `b`, and adding the arc `(a,b)` with coefficient 1. For example, `partsToArcs([[1,2,5][3,4,8][7,9]])` returns a dictionary `X` where `X[(1,2)] = X[(2,5)] = X[(3,4)] = X[(4,8)] = X[(7,9)] = 1` are all of the keys and values.

```
def partsToArcs(parts):
    arcs = dict()
    for x in parts:
        while len(x) > 1:
            a = x.pop(0)
            b = x[0]
            arcs[a,b] = 1
    return arcs
```

3.4 Tensor Products

Computing tensor products of sums of supercharacters will be carried out with the following general idea: iterate through all sums in the tensor product, at each stage picking and removing the first two sums, computing their product, and adding their product back into the list, until only one sum is left. This sum will be the final answer.

To get there, we first need machinery to compute the tensor product of arbitrary supercharacters. To begin, we need a way to compute the product of two supercharacters, both of which are defined by set partitions with a single arc. Then by repeating this process, and checking each time to see if the current tensor product corresponds to a supercharacter, we can compute arbitrary tensor products of supercharacters.

In order to multiply two supercharacters each indexed by a set partition with one arc, we use the function `arcTimes`. The function takes as inputs positive integers `i,k,j,l,m`, the set `S` $\subset \{1, 2, \ldots, n\}$, and a dictionary `X`, in order to compute the product $m\left(\chi^{i\frown k} \otimes \chi^{j\frown l}\right)$.

First, for ease of future needs, we order `(i,k)` and `(j,l)`. Then, the computations follow on a case by case basis from Theorem 2.4, and are implemented without trickery. For example, if I give the function the inputs

```
arcTimes(1,5,2,5,3,set([1,2,4,5]),X)
```

where `X` is an empty dictionary, then the function returns nothing and fills `X` in with the keys and values

```
X[((1,5),(2,4))]=X[(1,5)]=3.
```

It should be noted, however, that in python, a set is indicated by the word `set` in front of a list, and that `A & B` is the intersection of sets `A` and `B`.

```
def arcTimes(i, k, j, l, m, S, X):
    ((i,k),(j,l)) = dictord(i,k,j,l)
    if not (i,k) == (j,l):
        if len(set([i,k]) & set([j,l])) == 0:
            addX(((i,k),(j,l)),X,m)
        elif i < j == k < l:
            addX(((i,j),(k,l)),X,m)
        elif i == j < k < l:
            addX((i,l), X, m)
            for j_ in xrange(i+1,k):
                if j_ in S:
                addX(dictord(j_,k,i,l),X,m)
        else:
            addX((i,l),X,m)
            for k_ in xrange(j+1,l):
                if k_ in S:
                    addX(dictord(j,k_,i,l),X,m)
    else:
        addX(1, X, m)
        for j_ in xrange (i+1,l):
            if j_ in S:
                addX((i,j_), X, m)
        for k_ in xrange (i+1,l):
            if k_ in S:
                addX((k_,l), X, m)
        for j_ in xrange (i+1,l):
            if j_ in S:
                for k_ in xrange(i+1,l):
                    if k_ in S:
                        addX(dictord(i,j_,k_,l),X,m)
```

Let's take what may seem like a detour, but will help with larger tensor products. I now give a natural method for finding the "biggest problem" with the function `biggestProb`. This function will minimize the number of repeated computations in the recursion. The function `biggestProb` takes as input a list of arcs `x`, and returns a tuple `((i,l),(j,k),a,b,c,d)`. In this tuple, $\{(i,l),(j,k)\}$ is not a set partition, $a \geqslant b$ are the sizes of the arcs, i.e. $a = l - i$ and $b = k - j$, and there is no pair of arcs with sizes $a' \geqslant b'$ where $(a, b) < (a', b')$ as a dictionary ordering. If `arcs` does define a set partition, then there is no conflict, and `biggestProb` returns `((0,0),(0,0),0,0,0,0)` to indicate this.

Our first step is to create the variable `prob` as the tuple above, and let `l` denote the size of `x` for convenience. If the length is 1, then we're done, and we return the tuple `prob` full of zeros to indicate this. Otherwise, the function continues into nested for loops which iterate over all pairs of indices `i` and `j` from `1` to `l`. First, the function picks `i` and makes sure that `x[i]` didn't happen to be the trivial character `1` (which we could ignore). Then, the function selects a `j` less than `i` and checks to see if `x[j]` is the trivial character. If it isn't, then the function checks to see if the $\{x[i], x[j]\}$ is a proper set partition. That is, it checks if their initial vertices coincide, or if their end vertices coincide. If they do not, then the function moves on to the next step in the loop. If they do, then the function sets m to be the size of the larger arc, and p to be the size of the smaller arc. The function then compares their sizes to the sizes of the arcs currently stored in `prob`, and if the new sizes `(m,p)` are larger than the old sizes `prob[2],prob[3]` by a dictionary ordering, then we assign to `prob` the new "biggest problem," `(x[j],x[i],m,p,i,j)`. In the end, the function return the tuple `prob`. For example, let a list of arcs be

```
arcs = [(1,3),(1,5),(2,3),(2,7),(3,10)].
```

Then `prob(arcs)` returns `((2,3),(2,7),5,1,3,4)`, where 3 and 4 are the indices of the arcs `(2,3)` and `(2,7)` in `arcs`.

```
def biggestProb(x):
    prob = ((0,0), (0,0), 0, 0, 0, 0)
    l = len(x)
    if l == 1:
        return prob
    for i in xrange(1,l):
        if x[i] != 1:
            for j in xrange(0,i):
                if x[j] != 1:
                    if x[i][0] == x[j][0] or x[i][1] == x[j][1]:
                        m = max(x[i][1] - x[i][0], x[j][1]-x[j][0])
                        p = min(x[i][1] - x[i][0], x[j][1]-x[j][0])
                        if (m,p) > (prob[2], prob[3]):
                            prob = (x[j],x[i],m,p,i,j)
    return prob
```

I now give the algorithm to compute the tensor product of supercharacters defined by arbitrary set partitions of $\{1, 2, \ldots, n\}$. To do this, I have implemented a function `bigTimes` which takes as input a list of arcs (i.e., the multi sets Γ) called `arcs`, the product of the coefficients of all of the arcs `arcval`, the set `S`, and a dictionary `Sum`. The idea here will be to find the biggest problem, tensor those two arcs together, and recurse down each term in the sum. This is the implementation of the method for generating the sequences of multi sets.

The function begins by letting `p` denote the tuple returned by `biggestProb`. Then, if there is no problem and our list of arcs is a valid set partition (`p[2]==0`), then the function stores the partition and coefficient as a term in `Sum`. Some care is taken to format the set partition in the proper way (most of the following code is simply symbolic manipulation), but in this case the function does so and then returns nothing.

```
def bigTimes(arcs, arcval, S, Sum):
    p = biggestProb(arcs)
    if p[2] == 0:
        if len(arcs) == 1:
            addX(arcs[0], Sum, arcval)
        else:
            if 1 in arcs:
                arcs.remove(1)
            if len(arcs) == 1:
                addX(arcs[0], Sum, arcval)
            else:
                if 1 in arcs:
                    arcs.remove(1)
                if len(arcs) == 1:
                    addX(arcs[0], Sum, arcval)
                else:
                    arcs.sort()
                addX(tuple(arcs), Sum, arcval)
        return
```

Otherwise, there are arcs `p[0]` and `p[1]` which conflict. First, the function defines a dictionary called `temp`, and uses it to store the tensor product $arcval\left(\chi^{p[0]} \otimes \chi^{p[1]}\right)$. Then, the function throws out the arcs `p[0]` and `p[1]` (whose indices are `p[4]` and `p[5]`). One must be a little careful. For example, if the indices were 3 and 4, then one should pop `arcs[4]` first, then `arcs[3]`. If one popped `arcs[3]` first, then the arc which was in `arcs[4]` would now be `arcs[3]`, and throwing out `arcs[4]` would throw out the wrong arc, or even exceed the bounds of the list `arcs`. Because of this, the function first throws out the maximum of the two values first, and then the minimum. The function then iterates through the terms in the sum `temp`, which recall was the tensor product of the conflicting arcs, and for each term in the sum makes a new list of arcs `newarcs`, with what remains of `arcs`, and the arcs from the current term of `temp`, and then recurses by calling itself with the `newarcs`, the coefficient `item[1]`, the set `S`, and the dictionary `Sum`.

```
    temp = dict()
    arcTimes(p[0][0], p[0][1], p[1][0], p[1][1], arcval, S, temp)
    arcs.pop(max(p[4],p[5]))
    arcs.pop(min(p[4],p[5]))
    for item in temp.iteritems():
        newarcs = list(arcs)
        if item[0] == 1:
            newarcs.append(item[0])
        elif type(item[0][0]) == int:
            newarcs.append(item[0])
        else:
            for x in item[0]:
                newarcs.append(x)
        bigTimes(newarcs,item[1], S, Sum)
```

Now that we can multiply any amount of supercharacters together, we turn out attention to sums of supercharacters. The function takes as inputs a list of dictionaries `Y`, which in which each dictionary is a sum of supercharacters, and each item in the list is tensored together. Here, we will simply iterate over the sums `tempsum1` (initially set to `Y[0]`) and the next term in our tensor product of sums `Y[j]`, as if they are polynomials, and tensor supercharacters pairwise, collecting terms in `tempsum2` as we go. Then, when we have finished one iteration, we will set `tempsum1 = tempsum2` and reiterate. Some checking must be done to see if a term is the trivial character, and whether to combine tuples in the right way, but most important is that for any two nontrivial supercharacters, the function generates a list [arcs] which is a list whose entries are each arc counted once for each time it appeared in `tempsum1` or `Y[j]`. When the function finishes, it simply returns `tempsum1`.

```
def SumTensor(Y, S, q):
    tempsum1 = dict()
    tempsum2 = dict()
    tempsum1 = Y[0]
    for j in xrange(1, len(Y)):
        for x in tempsum1.iterkeys():
            for y in Y[j].iterkeys():
                if x == 1:
                    addX(y, tempsum2, tempsum1[x] * Y[j][y])
                elif y == 1:
                    addX(x, tempsum2, tempsum1[x] * Y[j][y])
                else:
                    arcs = list()
                    if type(x[0]) == int:
                        arcs.append(x)
                    else:
                        for x_ in x:
                            arcs.append(x_)
                    if type(y[0]) == int:
                        arcs.append(y)
                    else:
                        for y_ in y:
                            arcs.append(y_)
                    bigTimes(arcs, tempsum1[x] * Y[j][y], S, tempsum2)
        tempsum1 = tempsum2
        tempsum2 = dict()
    return tempsum1
```

3.5 Restriction

With restriction, we shall wish to employ a trick similar to tensor products. We take a set partition μ, compute the restriction of each $\chi^{i \frown l}$ for $i \frown l \in \mu$, and then we will tensor the resulting sums together.

The first step of restricting each $\chi^{i \frown l}$ is done in `restrictArc`. The function takes as input positive integers `i` and `l`, the set `S`, the coefficient `m`, `n` and `q` of $U_n(\mathbb{F}_q)$, and a dictionary `coeff` to store the values. The function follows straight from Theorem 2.4 and has been implemented without trickery.

```
def restrictArc(i, l, S, m, n, q, coeff):
    if i in S and l in S:
        t = len([k for k in set(range(i+1, l)) if not k in S])
        addX((i,l), coeff, q**t * m)
    elif not i in S and l in S:
        t = len([k for k in set(range(i+1, l)) if not k in S])
        c = q**t * m
        addX(1, coeff, c)
        for j in S:
            if i < j and j < l:
                addX((j,l), coeff, c)
    elif i in S and not l in S:
        t = len([k for k in set(range(i+1, l)) if not k in S])
        c = q**t * m
        addX(1, coeff, c)
        for j in S:
            if i < j and j < l:
```

```
            addX((i,j), coeff, c)
    else:
        t = len([k for k in set(range(i+1, l)) if not k in S])
        q_ = q**t
        c = q_ * (len(S & set(range(i,l+1))) * (q-1) + 1) * m
        addX(1, coeff, c)
        for j in S & set(range(i+1, l)):
            for k in S & set(range(j+1, l)):
                addX((j,k), coeff, q_ * (q-1) * m)
```

We now consider larger restrictions than a single arc. The idea here is that the restriction of each arc gives a sum of supercharacters, and the tensor products can be computed with existing machinery. To be more precise,

$$\mathrm{Res}(\chi^{\mu}) = \mathrm{Res}(\chi^{\mu_1}) \otimes \cdots \otimes \mathrm{Res}(\chi^{\mu_m})$$

where $\mu_1, \cdots, \mu_m$ are all of the arcs in μ counted once. The function takes as inputs `X`, `S`, `n`, and `q` as usual. It then restricts each arc, appending the resulting sum in a dictionary to a list `Y`.

```
def Restrict(X, S, n, q):
    Y = list()
    dummy = dict()
    for item in X.iteritems():
        if item[0] == 1:
            Y.append( makeDict(([1], item[1])) )
        else:
            restrictArc(item[0][0], item[0][1], S, item[1], n, q, dummy)
            Y.append(dummy)
            dummy = dict()
      return Y
```

The final just puts it all together.

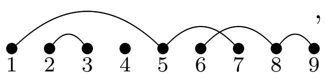
```
def restrictChar(X, S, n, q):
    Y = Restrict(X, S, n, q)
    return SumTensor(Y, X, S, q)
```

4 The Trivial Character in a Restriction

In this section I will investigate less algorithmic interpretations of the set partition combinatorics. First, I will introduce partial order combinatorics on the arcs, and use them to give the final theorem: a necessary and sufficient condition for the coefficient of the trivial character to be nonzero in a restriction. I will continue to use U_n in place of $U_n(\mathbb{F}_2)$, and I will frequently write c_μ^γ to be the coefficient of χ^γ in the sum $\mathrm{Res}_{U_S}^{U_n}(\chi^\mu) = \sum_{\psi \in \hat{\mathcal{X}}} c_\mu^\psi \chi^\psi$.

4.1 Colored Partial Orders

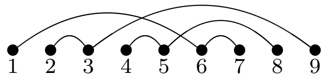
Define the a relation on arcs by $i \frown l \leqslant j \frown k$ when $j \leqslant i < l \leqslant k$. Visually, one arc is less than another if it is entirely underneath it. Let $\mu \subset \{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$ be a set or multi set of arcs (not necessarily a set partition). The *arc ordering* of μ denoted P^μ is the partial ordering on the arcs of μ with the relation given before. In the example from before,



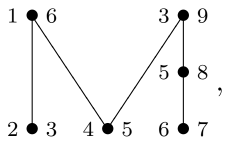
the only nontrivial relation is $2 \frown 3 < 1 \frown 5$. Consider the partition μ of $\{1, 2, \ldots, 9\}$,

$$\{1 \frown 6 \frown 7, 2 \frown 3 \frown 9, 4 \frown 5 \frown 8\}.$$

The diagram of μ is

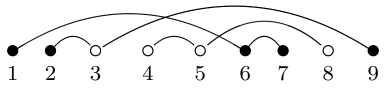


and the Hasse diagram of the arc ordering P^μ is

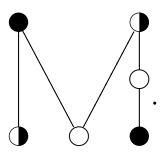


where I've added numbers to indicated which arc each vertex corresponds to.

Let $S \subset \{1, 2, \ldots, n\}$. Let an arc $i \frown l$ be colored ● if $i, l \in S$, ○ if $i, l \notin S$, and ◑ if exactly one of $i \in S$ or $l \in S$. Define P_S^μ to be the digraph with vertices and edges given by the Hasse diagram of P^μ where the vertices are colored by the previous rule. For example, let $S = \{1, 2, 6, 7, 9\}$. Then S and μ have the diagram



and P_S^μ is the colored Hasse diagram



Although I have defined the colored partial ordering as a digraph to avoid pathologies, I will think of it as a partial ordering with colored points. Also, I will think of a *sub-partial order* of P_S^μ as any subgraph of P_S^μ with the same coloring which maintains transitivity (and hence is the colored Hasse diagram of some partial order).

To give motivation to these combinatorics, notice that in some sense what is happening in a restriction only depends on how any individual arc is being restricted (its coloring), and what arcs it may potentially conflict with when restricted (which arcs are underneath it).

4.2 The Coefficient $c_\mu^{\mathbb{1}}$

In this section I will build a series of lemmas regarding the coefficient $c_\mu^{\mathbb{1}}$, culminating in a classification of when $c_\mu^{\mathbb{1}} \neq 0$, and relate it to a well understood combinatorial problem. First, I will give the theorem I am working toward.

Theorem 4.1. *Let μ be a set partition of $\{1, 2, \ldots, n\}$, and $S \subset \{1, 2, \ldots, n\}$. Let*

$$\nu = \{i \frown l \in \mu \big| i \frown l \text{ is not colored } ◑\}.$$

Then $c_\mu^{\mathbb{1}} \neq 0$ if and only if there is a non-induced sub-ordering P of P_S^ν such that

i) every arc in P_S^ν is in P

ii) each ● colored arc is less than a ○ colored arc

iii) and each ○ colored arc has at most one ● colored arc beneath it.

First, a simple lemma regarding the coefficient of $\mathbb{1}$ in a tensor product of supercharacters.

Lemma 4.2. *Let γ and μ be set partitions of $\{1, 2, \ldots, n\}$. Then*

$$\chi^\gamma \otimes \chi^\mu \big|_{\mathbb{1}} \neq 0$$

if and only if $\gamma = \mu$.

Proof. Let g_ν be any member of a superclass ν. Note that $\mathbb{1}$ evaluates to 1 on any element. Also, because supercharacters are from $U_n(\mathbb{F}_2)$ are real [8], $\chi^\nu(g_\nu) = \overline{\chi^\nu(g_\nu)}$. Thus, we have that coefficient of $\mathbb{1}$ in the tensor product is

$$\begin{aligned}\chi^\gamma \otimes \chi^\mu \big|_{\mathbb{1}} &= \langle \chi^\gamma \otimes \chi^\mu, \mathbb{1}\rangle = \frac{1}{|U_n|} \sum_{\nu \in \mathcal{K}} |\mathcal{K}^\nu|\, \chi^\gamma(g_\nu)\chi^\mu(g_\nu)\overline{\mathbb{1}(g_\nu)} \\ &= \frac{1}{|U_n|} \sum_{\nu \in \mathcal{K}} |\mathcal{K}^\nu|\, \chi^\gamma(g_\nu)\chi^\mu(g_\nu) = \frac{1}{|U_n|} \sum_{\nu \in \mathcal{K}} |\mathcal{K}^\nu|\, \chi^\gamma(g_\nu)\overline{\chi^\mu(g_\nu)} \\ &= \langle \chi^\gamma, \chi^\mu \rangle .\end{aligned}$$

Because supercharacters are orthogonal, this is nonzero if and only if $\chi^\gamma = \chi^\nu$. □

The following lemma which does not directly address $c_\mu^{\mathbb{1}}$ will be useful in the next corollary.

Lemma 4.3. *Let μ be a set partition of $\{1,2,\ldots,n\}$ and let $i \frown l$ be an arc of $\{1,2,\ldots,n\}$. Then*

$$\chi^{i\frown l} \otimes \chi^{\mu}\big|_{\chi^{i\frown l}} \neq 0$$

if and only if μ is one of the following:

1. $\mu = \emptyset$
2. $\mu = i \frown l' < i \frown l$ *or* $\mu = i' \frown l < i \frown l$
3. $\mu = \{i \frown l', i' \frown l\}$ *with* $i \frown l', i' \frown l < i \frown l$.

Proof. The necessity is clear. Let us investigate the sufficiency. Suppose

$$\chi^{i\frown l} \otimes \chi^{\mu}\big|_{\chi^{i\frown l}} \neq 0.$$

Note that by inspection of Theorem 2.4, μ must not have any arcs greater than or equal to $i \frown l$. From here, investigate the following cases.

Case 1. $|\mu| = 0$
Then $\mu = \emptyset$, $\chi^\mu = \mathbb{1}$, and nothing needs to be checked.

Case 2. $|\mu| = 1$
Say $\mu = j \frown k$. Then $j \frown k$ must conflict with $i \frown l$, but also $j \frown k < i \frown l$, and nothing more needs to be checked.

Case 3. $|\mu| = 2$
Let $\mu = \{i' \frown l', i'' \frown l''\}$ with $i' < i''$. Then there must be at least one conflict. Suppose there is only one. Without loss of generality, say $i' = i$ (l"=l is the same up to reflection of arcs). Then there are two possibilities: $i'' \frown l'' < i' \frown l'$, or not. In either case, a quick computation will show that

$$\chi^{i\frown l} \otimes \chi^{\mu}\big|_{\chi^{i\frown l}} = 0.$$

Case 4. $|\mu| \geqslant 3$
This case cannot happen. There may only be two conflicts in $\mu \cup \{i \frown l\}$, but there are at least 3 arcs which cannot all be removed in the tensor product. □

The next lemma gives a simple sufficient (though certainly not necessary) condition for $c_\mu^{\mathbb{1}}$ being nonzero. Mainly, if any supercharacter in the restriction is indexed by a set partition with a single arc $i \frown l \notin \mu$, then the coefficient of $\mathbb{1}$ is nonzero.

Lemma 4.4. *Let μ be a partition of $\{1,2,\ldots,n\}$, and let*

$$\mathrm{Res}_{U_S}^{U_n}(\chi^\mu) = \sum_{\psi \in \hat{\mathcal{X}}} c_\mu^\psi \chi^\psi.$$

If $c_\mu^{i\frown l} \neq 0$ for any $i \frown l \notin \mu$, then $c_\mu^{\mathbb{1}} \neq 0$.

Proof. Assume there exist arcs $x \frown y$ such that $x \frown y \notin \mu$ but $c_\mu^{x\frown y} \neq 0$. Let $i \frown l \notin \mu$ be a minimal arc such that $c_\mu^{i\frown l} \neq 0$. Consider any sequence given by the algorithm from before,

$$\Gamma = M_0, M_1, \ldots, M_k = \{i \frown l\}.$$

Consider the first t in the above sequence such that $\chi^{i\frown l}$ is a tensor in M_t and it is a tensor in each successive $M_{t'}$, $t' > t$.

Case 1. $t = 0$

Suppose $t = 0$. Then since $i \frown l \notin \mu$, there was some arc $j \frown k \in \mu$ such that $\mathrm{Res}_{U_S}^{U_n}\left(\chi^{j\frown k}\right)\big|_{\chi^{i\frown l}} \neq 0$. By inspection, Theorem 2.3 implies that $\mathrm{Res}_{U_S}^{U_n}\left(\chi^{j\frown k}\right)\big|_{\mathbb{1}} \neq 0$. So for $\nu = \mu - \{j \frown k\}$,

$$\begin{aligned}
\mathrm{Res}_{U_S}^{U_n}(\chi^\mu) &= \mathrm{Res}_{U_S}^{U_n}(\chi^\nu) \otimes \mathrm{Res}_{U_S}^{U_n}\left(\chi^{j\frown k}\right) \\
&= \mathrm{Res}_{U_S}^{U_n}(\chi^\nu) \otimes \left(c_{j\frown k}^{\mathbb{1}}\mathbb{1} + c_{j\frown k}^{i\frown l}\chi^{i\frown l} + \sum_{\gamma\neq \mathbb{1}, i\frown l} c_{j\frown k}^{\gamma}\chi^{\gamma}\right) \\
&= c_{j\frown k}^{\mathbb{1}}\mathrm{Res}(\chi^\nu) + c_{j\frown k}^{i\frown l}\chi^{i\frown l} \otimes \mathrm{Res}(\chi^\nu) + \sum_{\gamma\neq \mathbb{1}, i\frown l} c_{j\frown k}^{\gamma}\chi^{\gamma} \otimes \mathrm{Res}(\chi^\nu)
\end{aligned}$$

where I have dropped the scripts on $\mathrm{Res}_{U_S}^{U_n}(\chi)$ to conserve space.

Consider now the term

$$c_{j\frown k}^{i\frown l}\chi^{i\frown l} \otimes \mathrm{Res}(\chi^\nu).$$

By assumption, $i \frown l$ is in each successive $M_{t'}$. Thus, there exits a δ such that:

1. $c_\nu^\delta \neq 0$
2. $\chi^{i\frown l} \otimes \chi^\delta\big|_{\chi^{i\frown l}} \neq 0$

Then by Lemma 4.3, δ must be either $\emptyset, i \frown l', i' \frown l$, or $\{i \frown l', i' \frown l\}$. If $\delta = \emptyset$, then the term $c_{j\frown k}^{\mathbb{1}}\mathrm{Res}(\chi^\nu)$ has a nonzero trivial character. The second two cases, $\delta = i \frown l'$ or $\delta = i' \frown l$ give then in the term $c_{j\frown k}^{\mathbb{1}}\mathrm{Res}(\chi^\nu)$ a nonzero $\chi^{i\frown l'}$ or $\chi^{i'\frown l}$, contradicting the minimality of $i \frown l$. In the final case, $\delta = \{i \frown l', i' \frown l\}$. In any case, one of the terms in the sum

$$\sum_{\gamma\neq \mathbb{1}, i\frown l} c_{j\frown k}^{\gamma}\chi^{\gamma}$$

will create either an arc contradicting minimality, or a trivial character.

Otherwise, $t > 0$, which I will assume for the following cases. Consider the multi set M_{t-1}. There must be a pair $j \frown k, j' \frown k'$ such that $\{j \frown k, j' \frown k'\}$ is not a set partition (i.e., $j = k$ or $j' = k'$), and $\chi^{j\frown k} \otimes \chi^{j'\frown k'}\big|_{\chi^{i\frown l}} \neq 0$. By Theorem 2.4, this can only result from the following cases.

Case 2. $\{j \frown k, j' \frown k'\} = \{i' \frown l', i \frown l'\}$, where, $i' < i$ and $l < l'$.

Write $\psi = M_{t-1} - \{i' \frown l', i \frown l'\}$. Then by reordering the tensor products, we can break things up as

$$\begin{aligned}\left(\bigotimes_{\xi\in\psi}\chi^{\xi}\right)\otimes\left(\chi^{i\frown l'}\otimes\chi^{i'\frown l'}\right) &= \left(\sum_{\gamma\in\hat{\mathcal{X}}}c^{\gamma}\chi^{\gamma}\right)\otimes\left(\chi^{i'\frown l'}+\sum_{i<j<l'}\chi^{i'\frown l',i\frown j}\right)\\ &= \left(\sum_{\gamma\in\hat{\mathcal{X}}}c^{\gamma}\chi^{\gamma}\right)\otimes\chi^{i'\frown l'}\otimes\left(\mathbb{1}+\sum_{i<j<l'}\chi^{i\frown j}\right)\\ &= \left(\sum_{\gamma\in\hat{\mathcal{X}}}c^{\gamma}\chi^{\gamma}\right)\otimes\chi^{i'\frown l'}\otimes\left(\mathbb{1}+\sum_{i<j<l'}\chi^{i\frown j}\right)\\ &= \left(\sum_{\gamma\in\hat{\mathcal{X}}}d^{\gamma}\chi^{\gamma}\right)\otimes\left(\mathbb{1}+\sum_{i<j<l'}\chi^{i\frown j}\right)\end{aligned}$$

So there must be a δ such that $d^{\delta} \neq 0$, such that δ is either $\emptyset, i \frown l', i' \frown l$ or $\{i \frown l', i' \frown l\}$, and the analysis is almost identical to before.

Case 3. $\{j \frown k, j' \frown k'\} = \{i' \frown l', i' \frown l\}$, where, $i' < i$ and $l < l'$.

This case is identical to Case 2 up to a reflection of the arcs.

Case 4. $\{j \frown k, j' \frown k'\} = \{i' \frown l, i' \frown l\}$, where, $i' < i$.

Again, write $\psi = M_{t-1} - \{i' \frown l, i' \frown l\}$. Then break things up similarly to before,

$$\left(\bigotimes_{\xi\in\psi}\chi^{\xi}\right)\otimes\left(\chi^{i'\frown l}\otimes\chi^{i'\frown l}\right)$$

$$= \left(\sum_{\gamma\in\hat{\mathcal{X}}}a_{\gamma}\chi^{\gamma}\right)\otimes\left(\mathbb{1}+\sum_{i'<k<l}\chi^{i'\frown k}+\sum_{i'<j<l}\chi^{j\frown l}+\sum_{i'<j,k<l}\chi^{i'\frown kj\frown l}\right).$$

Then the term $\chi^{i\frown l}$ either came from the sum

$$\sum_{i'<j<l}\chi^{j\frown l},$$

in which case we run an analysis like in Case 1, or it came from the sum

$$\sum_{i'<j,k<l}\chi^{i'\frown kj\frown l},$$

in which case we take the necessary term from $\sum_{i'<k<l}\chi^{i'\frown k}$, and factor out a $\chi^{i'\frown k}$ as before.

Case 5. $\{j \frown k, j' \frown k'\} = \{i \frown l', i \frown l'\}$, where $l < l'$.

This case is identical to Case 3 up to a reflection of the arcs.

Thus, the cases are exhausted and the lemma holds. $\square$

In the next lemma, the result just used helps to prove that, while removing a ◑ colored arc may affect the coefficient of $\mathbb{1}$, it does not affect whether or not it is nonzero.

Lemma 4.5. *Let $S \subset \{1, 2, \ldots, n\}$ and μ be a set partition of $\{1, 2, \ldots, n\}$ with at least one ◑ colored arc, $i \frown l$. Define ν to be the set partition $\nu = \mu - \{i \frown l\}$. Let*

$$\mathrm{Res}^{U_n}_{U_S}(\chi^\mu) = \sum_{\psi \in \hat{\mathcal{X}}} c^\psi_\mu \psi \qquad \textit{and} \qquad \mathrm{Res}^{U_n}_{U_S}(\chi^\nu) = \sum_{\psi \in \hat{\mathcal{X}}} c^\psi_\nu \psi.$$

Then $c^{\mathbb{1}}_\mu = 0$ if and only if $c^{\mathbb{1}}_\nu = 0$.

Proof. Let S, μ, and ν be as described. Suppose that $c^{\mathbb{1}}_\nu \neq 0$. Then

$$\mathrm{Res}^{U_n}_{U_S}(\chi^\mu) = \mathrm{Res}^{U_n}_{U_S}(\chi^{\nu \cup \{i \frown l\}}) = \mathrm{Res}^{U_n}_{U_S}(\chi^\nu) \otimes \mathrm{Res}^{U_n}_{U_S}(\chi^{i \frown l})$$

Since the coefficient of $\mathbb{1}$ in both of $\mathrm{Res}^{U_n}_{U_S}(\chi^\nu)$ and $\mathrm{Res}^{U_n}_{U_S}(\chi^{i \frown l})$ is nonzero, the coefficient of $\mathbb{1}$ in $\mathrm{Res}^{U_n}_{U_S}(\chi^\mu)$ is nonzero.

Assume that $c^{\mathbb{1}}_\mu \neq 0$. If $c^{\mathbb{1}}_\nu \neq 0$, then we're done. Otherwise, $c^{\mathbb{1}}_\nu = 0$. Since $i \frown l$ is colored ◑, without loss of generality say that $i \in S$. Then

$$\mathrm{Res}^{U_n}_{U_S}(\chi^{i \frown l}) = c^{\mathbb{1}}_{i \frown l} \mathbb{1} + \sum_{j \in \{i+1, \cdots, l-1\} \cap S} c^{i \frown j}_{i \frown l} \chi^{i \frown j}.$$

This gives us that

$$\begin{aligned} \mathrm{Res}^{U_n}_{U_S}(\chi^\mu) &= \mathrm{Res}^{U_n}_{U_S}(\chi^\nu) \otimes \left(c^{\mathbb{1}}_{i \frown l} \mathbb{1} + \sum_{j \in \{i+1, \cdots, l-1\} \cap S} c^{i \frown j}_{i \frown l} \chi^{i \frown j} \right) \\ &= c^{\mathbb{1}}_{i \frown l} \mathrm{Res}^{U_n}_{U_S}(\chi^\nu) + \left(\mathrm{Res}^{U_n}_{U_S}(\chi^\nu) \otimes \sum_{j \in \{i+1, \cdots, l-1\} \cap S} c^{i \frown j}_{i \frown l} \chi^{i \frown j} \right). \end{aligned}$$

Now because $c^{\mathbb{1}}_\nu = 0$, but $c^{\mathbb{1}}_\mu \neq 0$, it must be that $d_{\mathbb{1}} \neq 0$ in the sum

$$\mathrm{Res}^{U_n}_{U_S}(\chi^\nu) \otimes \sum_{j \in \{i+1, \cdots, l-1\} \cap S} c^{i \frown j}_{i \frown l} \chi^{i \frown j} = \sum d_\psi \chi^\psi.$$

But, by Lemma 4.2 this implies that there is some $i \frown j$ such that $c^{i \frown j}_\nu \neq 0$. Then, by Lemma 4.4, we have that $c^{\mathbb{1}}_\nu \neq 0$. But $c^{\mathbb{1}}_\nu$ was assumed to be zero. Thus, $c^{\mathbb{1}}_\nu \neq 0$, giving the lemma. $\square$

The following lemma concerns the progression of arcs in a multi set sequence in the algorithm for restriction.

Lemma 4.6. *Let*

$$\Gamma = M_0, M_1, \dots, M_k = \nu$$

be any sequence in the algorithm. Then for each $1 \leqslant t \leqslant k$ there is an injective function $\theta_t : M_t \hookrightarrow M_{t-1}$ such that for each $j \frown k \in M_t$, $j \frown k \leqslant \theta_t(j \frown k)$.

Proof. Let $M_{t-1} = \psi \cup \{j \frown k, j' \frown k'\}$, where $j \frown k$ and $j' \frown k'$ conflict, and $M_t = \psi \cup \delta$ for a fixed δ with $c^\delta \neq 0$ in the sum

$$\chi^{j \frown k} \otimes \chi^{j' \frown k'} = \sum_{\delta \in \hat{\mathcal{X}}} c^\delta \chi^\delta.$$

Then by inspection of Theorem 2.4, there is a such a map from δ to $\{j \frown k, j' \frown k'\}$. Mapping the remaining arcs to themselves build θ_t. □

This leads immediately to the following corollary.

Corollary 4.7. *Let μ be a set partition of $\{1, 2, \dots, n\}$ such that*

$$\mathrm{Res}^{U_n}_{U_S}(\chi^\mu) = \sum_{\nu \in \hat{\mathcal{X}}} c^\nu_\mu \chi^\nu.$$

Then for each ν with $c^\nu_\mu \neq 0$ there is an injective function

$$\vartheta : \nu \hookrightarrow \mu$$

such that

$$j \frown k \leqslant \vartheta(j \frown k).$$

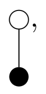
In particular, $|\nu| \leqslant |\mu|$.

Proof. Any composition of the injective functions in Lemma 4.6 leads to an injective function

$$\vartheta' = \theta_k \circ \theta_{k-1} \circ \cdots \circ \theta_1 : \nu \hookrightarrow \Gamma$$

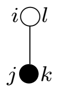
for which $j \frown k \leqslant \vartheta'(j \frown k)$, and by inspection of Theorem 2.3, there is a function from $\Theta : \Gamma \hookrightarrow \mu$ for which $j \frown k \leqslant \Theta(j \frown k)$. Then $\vartheta = \Theta \circ \vartheta'$ is the desired function. □

We now have everything necessary for the final proof of the theorem, which falls out fairly easily from the previous lemmas. Before addressing the proof, I'd like to note what is going on in P^μ_S. First, remove all ◑ colored dots from the Hasse diagram, and fill in the necessary edges to maintain all relations. Then, check if there is a way to remove two element chains of the form



so that eventually all that is left of the Hasse diagram are ○ dots. If so, then the coefficient $c^{\mathbb{1}}_{\mu}$ is nonzero. In some sense, this can be thought of as every ● colored arc needs an ○ colored arc above it to cancel it out, and an ○ colored dot can only cancel out at most one ● colored dot beneath it.

Proof of Theorem 4.1. First, note that by Lemma 4.5, $c^{\mathbb{1}}_{\mu} = 0$ if and only if $c^{\mathbb{1}}_{\nu} = 0$ by successively removing ◑ colored arcs from μ. Now, suppose that such a partial ordering P exists. Then for each two element chain



employ the usual trick of writing tensor product of the two restrictions as a sum of supercharacters. That is,

$$\begin{aligned}\mathrm{Res}^{U_n}_{U_S}\left(\chi^{i\frown l}\right) \otimes \mathrm{Res}^{U_n}_{U_S}\left(\chi^{j\frown k}\right) &= \left(c^{\mathbb{1}}_{i\frown l}\mathbb{1} + c\sum_{m\frown p<i\frown l}\chi^{m\frown p}\right) \otimes c^{j\frown k}_{j\frown k}\chi^{j\frown k} \\ &= \sum_{\gamma\in\hat{\mathcal{X}}} d^{\gamma}\chi^{\gamma}.\end{aligned}$$

Then $d^{\mathbb{1}} \neq 0$ because $j \frown k < i \frown l$ and so $\chi^{j\frown k}$ appears in the sum $\sum_{m\frown p<i\frown l}\chi^{m\frown p}$. So the coefficient of $\mathbb{1}$ is nonzero in the tensor product associated to each two element chain. Also, each single element chain is colored ○, and the coefficient of $\mathbb{1}$ will be nonzero in the restriction of each of these. So, by associating each trivial character $\mathbb{1}$ in each tensor, the coefficient of $\mathbb{1}$ in the restriction is nonzero.

Suppose now that $c^{\mathbb{1}}_{\mu} \neq 0$. Then $c^{\mathbb{1}}_{\nu} \neq 0$. Let

$$\xi = \{j \frown k \in \nu | j \frown k \text{ is colored } ○\}$$

and let

$$\psi = \{j \frown k \in \nu | j \frown k \text{ is colored } ●\}.$$

Then

$$\begin{aligned}\mathrm{Res}^{U_n}_{U_S}\left(\chi^{\nu}\right) &= \mathrm{Res}^{U_n}_{U_S}\left(\chi^{\xi}\right) \otimes \mathrm{Res}^{U_n}_{U_S}\left(\chi^{\psi}\right) \\ &= \left(\sum_{\gamma\in\hat{\mathcal{X}}} c^{\gamma}_{\xi}\chi^{\gamma}\right) \otimes c^{\psi}_{\psi}\chi^{\psi},\end{aligned}$$

where the last equality holds because $\psi \subset S \times S$. Since $c^{\mathbb{1}}_{\nu} \neq 0$, by Lemma 4.2, we have that $c^{\psi}_{\xi} \neq 0$. But then by Lemma 4.7, there is an injective function $\vartheta : \psi \hookrightarrow \nu$ such that $j \frown k \leqslant \vartheta(j \frown k)$. Thus, there is a sub partial ordering given by

$$\begin{tikzcd} \vartheta(j \frown k) \arrow[d, no head] \\ j \frown k \end{tikzcd}$$

which matches all of the criteria *i*, *ii*, and *iii*. □

In general however, the coefficient $c^{\mathbb{1}}_{\mu}$ is greater than these partial orders would indicate. However, they do give a lower bound.

Corollary 4.8. *Using notation as above, the sum of the coefficient of* $\mathbb{1}$ *in each possible* P *is a lower bound on the coefficient* $c^{\mathbb{1}}_{\mu}$.

It might be worrisome that Theorem 4.1 is a useless classification. However, the problem of whether such a sub partial ordering exists is equivalent to a well known combinatorial problem.

Corollary 4.9. *Let* μ *be a set partition of* $\{1, 2, \dots, n\}$ *and* $S \subset \{1, 2, \dots, n\}$. *Construct from* P^{μ}_{S} *the (undirected bipartite) graph* $\mathcal{P}$*:*

1. *Let the vertices of* $\mathcal{P}$ *be*

$$\nu = \{i \frown l \in \mu \,|\, i \frown l \text{ is not colored } ◑\}.$$

2. *Let there be an edge between the vertices* $i \frown l$ *and* $j \frown k$ *if* $i \frown l$ *is colored* ○ *and* $j \frown k$ *is colored* ●, *and* $j \frown k \leqslant i \frown l$.

Then using notation as above, P^{μ}_{S} *has such a sub partial order* P *is equivalent to there being a complete matching from the* ● *colored dots to the* ○ *colored dots in* $\mathcal{P}$.

References

[1] André, C. "Basic characters of the unitriangular group," *J. algebra* **175** (1995), 287–319.

[2] André, C. "Irreducible characters of finite algebra groups," *Matrices and group representations Coimbra, 1998* Textos Mat. Sér B **19** (1999), 65–80.

[3] André, C. "The basic character table of the unitriangular group," *J. algebra* **241** (2001), 437–471.

[4] André, C. "Basic characters of the unitriangular group (for arbitrary primes)," *Proc. Amer. Math. Soc.* **130** (2002), 1934–1954.

[5] André, C; Neto, A. "Super-characters of finite unipotent groups of types B_n, C_n and D_n," August 2006 preprint.

[6] André, C; Nicolás, A. "Supercharacters of the adjoint group of a finite radical ring," August 2006 preprint.

[7] Arregi, J; Vera-Lopez, A. "Computing in unitriangular matrices over finite fields." *Linear algebra and Appl.* **387** (2004), 193–219.

[8] Arias-Castro E; Diaconis, P; Stanley, R. "A super-class walk on upper-triangular matrices," *J. algebra* **278** (2004), 739–765.

[9] Bergeron, N; Hohlweg, C; Rosas, M; Zabrocki, M. "Grothendieck bialgebras, partition lattices, and symmetric functions in noncommutative variables," *Electron. J. Combin.* **13** (2006).

[10] Boyarchenko, M; Drinfeld, V. "A motivated introduction to character sheaves and the orbit method for unipotent groups in positive chracteristic." Preprint (2006), arXiv.0609769.

[11] Diaconis, P; Isaacs, M. "Supercharacters and superclasses for algebra groups," To appear in *Trans. Amer. Math. Soc.*, 2006.

[12] Diaconis, P; Saloff-Coste, L. "Comparison techniques for random walk on finite groups," *Ann. Probab.* **21** (1993), 2131–2156.

[13] Diaconis, P; Thiem, N. "Supercharacter formulas for pattern groups." To appear in *Trans. Amer. Math. Soc.*, 2007.

[14] Macdonald, I.G. *Symmetric functions and Hall polynomials. Second edition. With Contributions by A. Zelevinsky.* Oxford Mathematical Monographs. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1995.

[15] Marberg, E; Thiem, N. "Superinduction for pattern groups." A 2007 preprint, , arXiv:0712.1228..

[16] Sagan, B; Rosas, M. "Symmetric functions in noncommuting variables," *Trans. Amer. Math. Soc.* **358** (2004), 215–232.

[17] Thiem, N. "Branching rules in the ring superclass functions of unipotent upper-triangular matrices," A 2008 preprint, 2008 preprint.

[18] Thiem, N; Venkateswaran, V. "Restricting supercharacters of the finite group of unipotent uppertriangular matrices," A 2007 preprint, arXiv:0712.1237.

[19] Thiem, N; Vinroot, C.R. "On the characteristic map of finite unitary groups," *Adv. Math.* **210** (2007), 707–732.

[20] Wolf, M.C. “Symmetric functions of noncommuting elements,” *Duke Math. J.* **2** (1936), 626 637.

[21] Yan, N. *Representation theory of the finite unipotent linear groups*, Unpublished Ph.D. Thesis, Department of mathematics, University of Pennsylvania, 2001.

[22] Yan, N. “Representations of finite unipotent linear groups by the method of clusters,” 2006 preprint.