2005-06-20

Matrix Representations of Automorphism Groups of Free Groups

Ivan B. Andrus
Brigham Young University - Provo

MATRIX REPRESENTATIONS OF AUTOMORPHISM GROUPS OF FREE GROUPS

by

Ivan Andrus

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Mathematics

Brigham Young University

August 2005

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Ivan Andrus

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Stephen Humphries, Chair
Date	Darrin Doud
Date	Gregory Conner

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Ivan Andrus in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date

Stephen Humphries
Chair, Graduate Committee

Accepted for the Department

Tyler J. Jarvis
Graduate Coordinator

Accepted for the College

G. Rex Bryce, Associate Dean
College of Physical and Mathematical Sciences

ABSTRACT

MATRIX REPRESENTATIONS OF AUTOMORPHISM GROUPS OF FREE GROUPS

Ivan Andrus

Department of Mathematics

Master of Science

In this thesis, we study the representation theory of the automorphism group $\mathrm{Aut}\,(F_n)$ of a free group by studying the representation theory of three finite subgroups: two symmetric groups, S_n and S_{n+1}, and a Coxeter group of type B_n, also known as a hyperoctahedral group. The representation theory of these subgroups is well understood in the language of Young Diagrams, and we apply this knowledge to better understand the representation theory of $\mathrm{Aut}\,(F_n)$. We also calculate irreducible representations of $\mathrm{Aut}\,(F_n)$ in low dimensions and for small n.

Table of Contents

1 Introduction

We will denote by

$$F_n = \langle x_1, x_2, \ldots, x_n \mid \rangle$$

the free group of rank n. In this paper we are interested in studying $\mathrm{Aut}\,(F_n)$, the group of automorphisms of F_n. That is the group of $\alpha : F_n \to F_n$ such that α is a bijective homomorphism of groups. Our goal is to understand the representation theory of $\mathrm{Aut}\,(F_n)$ for small values of n. In other words we wish to understand the irreducible homomorphisms $\rho : \mathrm{Aut}\,(F_n) \to \mathrm{GL}_k(\mathbb{C})$. As is customary when studying $\mathrm{Aut}\,(F_n)$, automorphisms shall act on the right. However when studying S_n, and $\mathrm{Cox}\,(B_n)$, we shall act on the left as is customary for these groups.

Representation theory for finite groups is well developed, but there are few applicable results for infinite groups such as $\mathrm{Aut}\,(F_n)$. To perform our analysis we consider some finite subgroups of $\mathrm{Aut}\,(F_n)$. The most important of these is the Coxeter group of type B_n, denoted $\mathrm{Cox}\,(B_n)$. Other subgroups of interest include two symmetric groups S_n and $\Sigma_n \cong S_{n+1}$.

The Coxeter group is the subgroup of $\mathrm{Aut}\,(F_n)$ which permutes elements of the set

$$\left\{x_1, x_1^{-1}, x_2, x_2^{-1}, \ldots, x_n, x_n^{-1}\right\}$$

while respecting inverses *i.e.,* $\alpha(x_i^{-1}) = \alpha(x_i)^{-1}$ for all $\alpha \in \mathrm{Cox}\,(B_n)$. The hyperoctahedral group, as $\mathrm{Cox}\,(B_n)$ is often called, is generated by the symmetric group S_n, which permutes the indices, together with elements $\varepsilon_i : x_i \mapsto x_i^{-1}$ which take the i-th generator to its inverse and leave the others unchanged. This Coxeter group is also the group of symmetries of the n-cube $[-1, 1]^n$ [2], and has order $2^n n!$. It is sometimes called the signed permutation group for obvious reasons, and is the wreath product $(\mathbb{Z}_2) \wr S_n$ of the cyclic group of order 2 with the symmetric group on n letters. It is also known as the Weyl group of type C.

A result of Nielsen [15] tells us that $\mathrm{Aut}\,(F_n)$ is generated by four elements, denoted P, Q, σ, U, and defined as follows:

$$\begin{array}{cccc} P : x_1 \mapsto x_2 & Q : x_1 \mapsto x_2 & \sigma : x_1 \mapsto x_1^{-1} & U : x_1 \mapsto x_1 x_2 \\ x_2 \mapsto x_1 & x_2 \mapsto x_3 & x_2 \mapsto x_2 & x_2 \mapsto x_2 \\ x_3 \mapsto x_3 & x_3 \mapsto x_4 & x_3 \mapsto x_3 & x_3 \mapsto x_3 \\ \vdots & \vdots & \vdots & \vdots \\ x_n \mapsto x_n & x_n \mapsto x_1 & x_n \mapsto x_n & x_n \mapsto x_n \end{array} \tag{1}$$

In fact, Nielsen gives a presentation of $\mathrm{Aut}\,(F_n)$ with generators P, Q, σ, and U, and the defining relationships given in tables 1.1 and 1.2 (with $\rightleftarrows$ signifying 'commutes with'). We shall use these relations extensively throughout this paper.

Table 1.1: Relations not involving U

$$1 = P^2; \tag{P.1}$$

$$1 = Q^n; \tag{P.2}$$

$$1 = (QP)^{n-1}; \tag{P.3}$$

$$P \rightleftarrows Q^{-i} P Q^i \qquad i = 2, 3, \ldots, \lfloor n/2 \rfloor \tag{P.4}$$

$$1 = \sigma^2; \tag{P.5}$$

$$\sigma \rightleftarrows Q^{-1} P Q; \qquad n \neq 2 \tag{P.6}$$

$$\sigma \rightleftarrows QP; \tag{P.7}$$

$$\sigma \rightleftarrows Q^{-1} \sigma Q; \tag{P.8}$$

Table 1.2: Relations involving U

$$U \rightleftarrows Q^{-2}PQ^2; \qquad n \neq 2,3 \tag{U.1}$$

$$U \rightleftarrows QPQ^{-1}PQ; \qquad n \neq 2 \tag{U.2}$$

$$U \rightleftarrows Q^{-2}\sigma Q^2; \qquad n \neq 2 \tag{U.3}$$

$$U \rightleftarrows Q^{-2}UQ^2; \qquad n \neq 3 \tag{U.4}$$

$$U \rightleftarrows \sigma U \sigma; \tag{U.5}$$

$$U \rightleftarrows PQ^{-1}\sigma U \sigma QP; \tag{U.6}$$

$$U \rightleftarrows PQ^{-1}PQPUPQ^{-1}PQP; \qquad n \neq 2 \tag{U.7}$$

$$UQ^{-1}U = \left(PQ^{-1}UQ\right)^2 UQ^{-1} \qquad n \neq 2 \tag{U.8}$$

$$1 = (P\sigma PU)^2; \tag{U.9}$$

$$U = PUP\sigma U\sigma P\sigma. \qquad n \neq 1 \tag{U.10}$$

Later Neumann [14] gave presentations of $\mathrm{Aut}(F_n)$ with fewer generators, but we shall work with Nielsen's presentation for simplicity. We note that P, and Q, together with the relations (P.1) through (P.4) are a presentation of S_n, while P, Q and σ and the relations (P.1) through (P.8) define $\mathrm{Cox}(B_n)$ (though the action is considered to be on the right.) We will often denote by P, Q, σ, and U their images under a representation ρ.

Any representation of $\mathrm{Aut}(F_n)$ gives, by restriction, a representation of $\mathrm{Cox}(B_n)$, so we begin by studying the representation theory of $\mathrm{Cox}(B_n)$. From elementary representation theory there exists a finite set $R = \{\rho_1, \rho_2, \ldots, \rho_{N(n)}\}$ of irreducible representations of $\mathrm{Cox}(B_n)$ which have the property that any representation $\rho : \mathrm{Cox}(B_n) \to \mathrm{GL}_k(\mathbb{C})$ can be written, up to a change of basis, as the direct sum of (possibly multiple) copies of elements of R. Here $N(n)$ is the number of conjugacy classes of $\mathrm{Cox}\,(B_n)$.

Any representation ρ of $\mathrm{Aut}\,(F_n)$ is uniquely determined by $\rho\left(\mathrm{Cox}\,(B_n)\right)$ and $\rho\,(U)$. Thus we need only find all possible matrices $M = \rho(U)$ which satisfy the equations of table 1.2 for any given representation of $\mathrm{Cox}\,(B_n)$, or determine that no such matrix exists.

In an effort to understand the representations of $\mathrm{Aut}\,(F_n)$, we shall begin by studying the representation theory of S_n and $\mathrm{Cox}\,(B_n)$. From there we shall place restrictions on the representations which can be extended to representations of $\mathrm{Aut}\,(F_n)$. We enumerate the irreducible representations of $\mathrm{Aut}\,(F_n)$ in low dimensions, and give some general results concerning these representations.

One of the motivations for this paper is to understand the relationship between representations of the subgroups $\mathrm{Cox}\,(B_n)$, S_n, and Σ_n. This is determined in part by theorems 2.59 and 2.65.

2 General Results of Representation Theory

In this chapter we summarize some general results of representation theory that will be useful in our analysis of representations of $\mathrm{Aut}\,(F_n)$ and $\mathrm{Cox}\,(B_n)$.

Definition 2.1. *Let $\mathbb{F}$ be a field and V a vector space over $\mathbb{F}$. For a group G, an $\mathbb{F}$-representation (often called simply a representation when $\mathbb{F}$ is understood) of G is a homomorphism $\rho : G \to \mathrm{GL}(V)$ from G to the general linear group of V. In this paper we will only consider the case $\mathbb{F} = \mathbb{C}$ the complex numbers. Often we shall refer to the vector space V itself as a representation.*

Note that what we call a representation is sometimes called a linear representation.

Definition 2.2. *A matrix representation of a group G is a finite dimensional representation.*

Definition 2.3. *Let V be a representation of a group G. A subrepresentation W of V is a subspace of V which is invariant under G.*

Definition 2.4. *A representation ρ of G is said to be irreducible if it contains no proper subrepresentations.*

Definition 2.5. *Two matrix representations ρ_1, ρ_2 are said to be equivalent if they are conjugate,* i.e., *there exists an invertible matrix S such that $\rho_1 = S^{-1}\rho_2 S$. Such a conjugation is a change of basis, and we shall only define representations up to conjugation.*

Theorem 2.6. *A representation ρ of a finite group G is determined up to equivalence by the traces of its elements [7, 2.14].*

Trace is a conjugacy invariant, which implies that trace is a class function, *i.e.,* a function which is constant on conjugacy classes. Other properties of trace also show that for representations U and V of G

$$\chi_{U\oplus V} = \chi_U + \chi_V; \tag{2}$$

$$\chi_{U\otimes V} = \chi_U \cdot \chi_V. \tag{3}$$

Theorem 2.7. *The number of distinct irreducible representations of a finite group G is equal to the number of conjugacy classes [7, 2.30].*

Definition 2.8. *Given a group G and a representation $\rho : G \to \mathrm{GL}(V)$, we define the character χ_ρ of ρ by*

$$\chi_\rho(g) = \mathrm{tr}\left(\rho(g)\right). \tag{4}$$

Thus χ_ρ is a function $\chi_\rho : G \to \mathbb{C}$, or more generally character $\chi_\rho : G \to \mathbb{F}$ maps into an arbitrary field $\mathbb{F}$.

Theorem 2.9. *A representation ρ of a group G is determined up to equivalence by its character χ_ρ [7, 2.14]. This is a restatement of Theorem 2.6.*

Theorem 2.10. *The characters $\chi_{\rho_1}, \chi_{\rho_2}, \ldots, \chi_{\rho_N}$ of the irreducible representations are known as the irreducible characters and form a basis for the class functions on G [7, 2.30].*

Definition 2.11. *The set of class functions on G admits an inner product $\langle \cdot, \cdot \rangle$ given by*

$$\langle f, h \rangle = \frac{1}{|G|} \sum_{g\in G} \overline{f(g)}\, h(g),$$

for f, h class functions on G.

Theorem 2.12. *If $\rho_1, \rho_2, \ldots, \rho_N$ are the irreducible representations of a group G then the corresponding characters $\chi_{\rho_1}, \chi_{\rho_2}, \ldots, \chi_{\rho_N}$ form an orthonormal basis with respect to this inner product [7, 2.30]. In particular*

$$\left\langle \chi_{\rho_i}, \chi_{\rho_j} \right\rangle = \delta_{ij} = \begin{cases} 1 & i = j; \\ 0 & i \neq j. \end{cases} \tag{5}$$

Corollary 2.13. *If ρ is a representation of a group G, then ρ is irreducible if and only if $\langle \chi_\rho, \chi_\rho \rangle = 1$.*

Definition 2.14. *If H is a subgroup of G, then any representation V of G will restrict to a representation of H. We will denote this $\mathrm{Res}_H^G V$ or simply $\mathrm{Res}\, V$ if H and G are understood.*

Definition 2.15. *If H is a subgroup of G, then any representation of H can be used to induce a representation of G. It is determined by the action of G on cosets of H. We denote this $\mathrm{Ind}\, V = \mathrm{Ind}_H^G V$ [7, pg.33].*

Theorem 2.16. *The character of a representation induced from a representation W from H to G, can be calculated from the following formula*

$$\chi_{\mathrm{Ind}\, W}(g) = \sum_{g\sigma = \sigma} \chi_W(s^{-1} g s) \qquad s \in \sigma \text{ arbitrary}$$

involving the character of W and cosets σ of H in G [7, 3.18].

Theorem 2.17. *(Frobenius Reciprocity). If W is a representation of H, and U a representation of G, then*

$$\langle \chi_{\mathrm{Ind}\, W}, \chi_U \rangle_G = \langle \chi_W, \chi_{\mathrm{Res}\, U} \rangle_H$$

describes the relationship between the induced and restricted characters [7, 3.20]

Definition 2.18. *If V is a vector space over $\mathbb{C}$ with basis $\{v_1, v_2, \ldots, v_n\}$, then the k-th exterior power $\bigwedge^k V$ is defined as the quotient of the k-th tensor power $V^{\otimes k}$ and the ideal generated by elements $v_{i_1} \otimes v_{i_2} \otimes \cdots \otimes v_{i_k}$ such that $i_j = i_\ell$ for some k, ℓ. The 0-th exterior power $\bigwedge^0 V = \mathbb{C}$ is defined to be $\mathbb{C}$ itself, and the first exterior power $\bigwedge^1 V = V$ is the vector space V.*

The k-th exterior power has basis $\{v_{i_1} \wedge v_{i_2} \wedge \cdots \wedge v_{i_k}\}$ such that

$$v_{\alpha(i_1)} \wedge v_{\alpha(i_2)} \wedge \cdots \wedge v_{\alpha(i_k)} = \operatorname{sgn}(\alpha) v_{i_1} \wedge v_{i_2} \wedge \cdots \wedge v_{i_k}$$

for all $\alpha \in S_n$. The exterior powers are often thought of in terms of this basis.

Lemma 2.19. *Given an irreducible representation U, and a representation $V \neq U$, then V is irreducible if and only if $\langle \chi_{U \oplus V}, \chi_{U \oplus V} \rangle = 2$.*

Proof. We note that $\langle \chi_V, \chi_U \rangle$ is the number of times that U appears as a direst summand in V, and $\langle \chi_V, \chi_V \rangle$ is the number of irreducible components in V.

$$\begin{aligned} \langle \chi_{U \oplus V}, \chi_{U \oplus V} \rangle &= \langle \chi_V + \chi_U, \chi_V + \chi_U \rangle \\ &= \langle \chi_V, \chi_V \rangle + \langle \chi_V, \chi_U \rangle + \langle \chi_U, \chi_V \rangle + \langle \chi_U, \chi_U \rangle \\ &= \begin{cases} 1 + 0 + 0 + 1 = 2 & \text{if } V \text{ is irreducible} \\ y + x + x + 1 > 2 & \text{if } V \text{ is not irreducible (since } y > 1) \end{cases} \end{aligned}$$ ∎

Definition 2.20. *Let $\{e_1, e_2, \ldots, e_n\}$ be the standard basis for $\mathbb{C}^n$. The permutation representation of S_n is defined by $\alpha \cdot e_i = e_{\alpha(i)}$ for all $\alpha \in S_n$.*

Definition 2.21. *We note that $e_1 + e_2 + \cdots + e_n$ is fixed by the permutation representation of S_n, and so determines a subrepresentation which we call the trivial representation. The trivial representation is clearly irreducible since it is 1-dimensional.*

Definition 2.22. *The standard representation, V, of S_n is defined to be the complementary subspace of the trivial representation in the permutation representation.*

Theorem 2.23. *If V denotes the standard representation of S_n, then* $\dim V = n-1$ *and V is irreducible.*

Proof. This is a special case of the next theorem. ∎

Theorem 2.24. *For V the standard representation of S_n, the exterior powers $\bigwedge^k V$ of V are all irreducible for $k = 0, 1, \ldots, n-1$ [7, 3.12]. We will give this proof.*

Proof. Let V be the standard representation of S_n, and U be the trivial representation. Then $V \oplus U = \mathbb{C}^n$, the permutation representation, and $\chi_{V\oplus U} = \chi_V + \chi_U$. From Lemma 2.19 it suffices to show that $\langle \chi_{\mathbb{C}^n}, \chi_{\mathbb{C}^n} \rangle = 2$, since U is irreducible and $V \neq U$

From basic properties of exterior products [7, B.1] we know that

$$\bigwedge^k (V \oplus U) = \left(\bigwedge^0 V \otimes \bigwedge^k U\right) \oplus \left(\bigwedge^1 V \otimes \bigwedge^{k-1} U\right) \oplus \cdots \oplus \left(\bigwedge^k V \otimes \bigwedge^0 U\right)$$

Since U is 1-dimensional, $\bigwedge^j U = 0$ if $j \geq 2$. It is equivalent to the base field for both $j = 0, 1$. This leads to

$$\bigwedge^k (V \oplus U) = \bigwedge^{k-1} V \oplus \bigwedge^k V.$$

A basis for $\bigwedge^k \mathbb{C}^n$ is given by elements of the form

$$e_{i_1} \wedge e_{i_2} \wedge \cdots \wedge e_{i_k}$$

for all $1 \leq i_1 < i_2 < \cdots < i_k \leq n$. Here, as before, e_i is the i-th standard basis vector for $\mathbb{C}^n$.

Let $A = \{1, 2, \ldots, n\}$ and let $B \subset A$ be a subset of A with k elements. If we now define for $\alpha \in S_n$

$$\{\alpha\}_B = \begin{cases} 1 & \text{if } \alpha(B) = B \text{ and } \alpha\big|_B \text{ is even;} \\ -1 & \text{if } \alpha(B) = B \text{ and } \alpha\big|_B \text{ is odd;} \\ 0 & \text{if } \alpha(B) \neq B. \end{cases}$$

then we see that $\chi(\alpha) = \sum_{B} \{\alpha\}_B$. Thus we have

$$\begin{aligned}
\langle \chi, \chi \rangle &= \frac{1}{n!} \sum_{\alpha \in S_n} \left(\chi(\alpha)\right)^2 \\
&= \frac{1}{n!} \sum_{\alpha \in S_n} \left(\sum_{B} \{\alpha\}_B \right)^2 \\
&= \frac{1}{n!} \sum_{\alpha \in S_n} \sum_{B} \sum_{C} \{\alpha\}_B \{\alpha\}_C \\
&= \frac{1}{n!} \sum_{B} \sum_{C} \sum_{\substack{\alpha : \alpha(B) = B \\ \alpha(C) = C}} \operatorname{sgn}\left(\alpha\big|_B\right) \operatorname{sgn}\left(\alpha\big|_C\right).
\end{aligned}$$

In the above sums B and C range over all k-element subsets of A. A permutation α that fixes both B and C can be written as the product of four commuting permutations, one on $B \cap C$, one on $B \setminus (B \cap C)$, $C \setminus (B \cap C)$, and finally one on $A \setminus (B \cup C)$. Thus if we let ℓ be the cardinality of $B \cap C$, the above equation becomes

$$\begin{aligned}
\langle \chi, \chi \rangle &= \frac{1}{n!} \sum_{B} \sum_{C} \sum_{a \in S_\ell} \sum_{b \in S_{k-\ell}} \sum_{c \in S_{k-\ell}} \sum_{d \in S_{n-2k+\ell}} \left(\operatorname{sgn} a\right)^2 \left(\operatorname{sgn} b\right) \left(\operatorname{sgn} c\right) \\
&= \frac{1}{n!} \sum_{B} \sum_{C} \ell!(n-2k+\ell)! \left(\sum_{b \in S_{k-\ell}} \operatorname{sgn} b \right) \left(\sum_{c \in S_{k-\ell}} \operatorname{sgn} c \right)
\end{aligned}$$

We now note that the last two sums are zeros unless $k - \ell = 0$ or 1. Since $S_0 = S_1 = \langle id \rangle$, $\sum_{g \in S_{k-\ell}} \operatorname{sgn} g = 1$ in these cases. This allows us to write the sum over C in two terms as below.

$$\begin{aligned}
\langle \chi, \chi \rangle &= \frac{1}{n!} \sum_{B, k=\ell} k!(n-k)! + \frac{1}{n!} \sum_{B, k-\ell=1} (k-1)!(n-k-1)! \\
&= \frac{1}{n!} \binom{n}{k} k!(n-k)! + \frac{1}{n!} \binom{n}{k-1} (k-1)!(n-k-1)! \\
&= 1 + 1 = 2 \quad \blacksquare
\end{aligned}$$

Definition 2.25. *The complex group algebra $\mathbb{C}G$ of a group G is the algebra consisting of formal sums $\sum_{i=1}^{N} \alpha_i g_i$ with $\alpha_i \in \mathbb{C}$ and $g_i \in G$, and N finite. Addition* *is component-wise, and multiplication is defined by* $(\alpha_1 g_1)(\alpha_2 g_2) = (\alpha_1\alpha_2)(g_1 g_2)$*, extending linearly to sums.*

Left $\mathbb{C}G$-modules correspond exactly to representations of G, and minimal left ideals in $\mathbb{C}G$ correspond to irreducible representations of G. Thus all work in studying the representation theory of G can be done in the group algebra [7, pg. 36–37]. We shall use the group algebra in sections 2.1 and 2.2 when developing the representation theory of S_n and $\mathrm{Cox}\,(B_n)$.

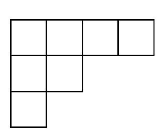
2.1 Young Diagrams

Irreducible representations of S_n are in one to one correspondence with conjugacy classes, which in turn are in one to one correspondence with partitions of n. We shall examine an algorithm for determining an irreducible representation of S_n given a partition of n. This is explained in lecture 4 of [7].

Definition 2.26. *A partition* λ *of a natural number* n *is an ordered* k*-tuple of positive integers* $(\lambda_1, \lambda_2, \ldots, \lambda_k)$ *such that* $\sum_{i=1}^{k} \lambda_i = n$ *and* $\lambda_i \geq \lambda_{i+1}$ *for all* i*. We may sometimes write a partition* λ *as*

$$n = \lambda_1 + \lambda_2 + \cdots + \lambda_N, \qquad \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq 1.$$

Definition 2.27. *For each partition there is an associated Young diagram, also known as a Young frame or Ferrers diagram. A Young diagram consists of* x_i *boxes in the* i*-th row, starting at the top and moving down. For example, the following Young diagram corresponds to the partition* $x = (4, 2, 1)$*.*



We will often abuse notation and refer to a partition and its associated Young diagram interchangeably.

Definition 2.28. *We define the lexicographical ordering on the set of partitions of* n *(and hence Young Diagrams) by* $\lambda > \mu$ *if the first non-zero* $\lambda_i - \mu_i$ *is positive.*

Definition 2.29. *A Young tableau consists of a Young diagram which is labeled with the integers* $1, 2, \ldots, n$ *in some order such that each integer appears only once. Below are two examples of Young tableaux corresponding to the Young diagram above.*

1	2	3	4
5	6		
7			

5	2	1	6
3	7		
4			

We now proceed to explain an algorithm for calculating a representation of S_n from a partition of n. Although the algorithm involves choosing a Young tableau, the results depend only on the Young diagram involved (up to a change of basis).

Starting with a Young tableau λ, define the row group $\mathrm{Row}(x)$, to be the permutations that fix the rows of λ as sets. Similarly define the column group $\mathrm{Col}(x)$ to be those which fix the columns, that is

$$\mathrm{Row}(x) = \{\alpha \in S_n \mid \alpha \text{ fixes rows of } \lambda\} \ ;$$
$$\mathrm{Col}(x) = \{\beta \in S_n \mid \beta \text{ fixes columns of } \lambda\} \ .$$

Now define elements $a, b, c \in \mathbb{C}S_n$ of the complex group algebra by

$$a = \sum_{\alpha \in \mathrm{Row}(x)} \alpha;$$
$$b = \sum_{\beta \in \mathrm{Col}(x)} \mathrm{sgn}(\beta)\, \beta;$$
$$c = ab.$$

The element c is known as a Young symmetrizer.

Theorem 2.30. *With c as defined above, $\mathbb{C}S_n c$ is an irreducible representation of S_n with action defined by*

$$\alpha \cdot (\gamma c) = \alpha\gamma c \quad \textit{for } \gamma \in \mathbb{C}S_n,\ \alpha \in S_n.$$

This is found in [7, 4.3]

Example 2.31. *Consider the partition $x = (2, 1)$ of 3. With the associated Young tableau,* $\begin{array}{|c|c|}\hline 1 & 2 \\ \hline 3 & \\ \cline{1-1}\end{array}$ *we have* $\mathrm{Row}(x) = \langle (12) \rangle$, *and* $\mathrm{Col}(x) = \langle (13) \rangle$. *Then*

$$\begin{aligned} a &= (1) + (12) \\ b &= (1) - (13) \\ c &= \big((1) + (12)\big)\big((1) - (13)\big) \\ &= (1) - (13) + (12) - (12)(13) \\ &= (1) - (13) + (12) - (123). \end{aligned}$$

We now consider an element c'

$$\begin{aligned} c' = (13)c &= (13) - (13)(13) + (13)(12) - (13)(123) \\ &= (13) - (1) + (132) - (23). \end{aligned}$$

It is easy to verify that

$$\begin{aligned} (12)c &= c; & (12)c' &= -c - c'; \\ (13)c &= c'; & (13)c' &= c; \\ (23)c &= -c - c'; & (23)c' &= c'; \\ (123)c &= -c - c'; & (123)c' &= c; \\ (132)c &= -c'; & (132)c' &= -c - c'. \end{aligned}$$

Thus $\text{Span}_{\mathbb{C}}\{c, c'\}$ *is fixed by* S_3*, and there is an irreducible representation* ρ *determined by the action of* S_3 *on* $\text{Span}_{\mathbb{C}}\{c, c'\}$*. We have as generators:*

$$\rho_{(12)} = \begin{bmatrix} 1 & 0 \\ -1 & -1 \end{bmatrix}; \qquad \rho_{(123)} = \begin{bmatrix} -1 & -1 \\ 1 & 0 \end{bmatrix}.$$

This is in fact the standard representation of S_3*, which has dimension* 2*.*

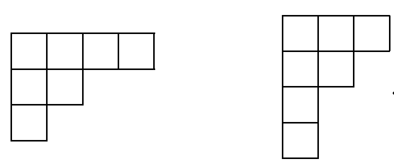
This construction shows that all representations of S_n can be constructed over $\text{GL}_n(\mathbb{Q})$, *i.e.*, the matrices can be all expressed with rational coefficients.

Definition 2.32. *We shall use* V_λ *to indicate the representation which is derived from the partition* λ*.*

Definition 2.33. *If* λ *is a partition of* n*, then its conjugate, or opposite, partition* λ'*, is defined by interchanging the rows and columns of its Young diagram. This can be done without referring to the Young diagram by defining*

$$(\lambda')_i = \left| \{\lambda_j \mid \lambda_j \geq i\} \right|.$$

Example 2.34. *The effect is to reflect the diagram about the* -45° *line as in the following example:*



Definition 2.35. *Consider a Young diagram corresponding to a partition* λ*. The hooklength of a box in* λ *is the number of boxes in* λ *directly to the right or directly below, including the box itself.*

Example 2.36. *The Young diagram below has each box labeled with the hooklengths.*

6	4	2	1
3	1		
1			

Theorem 2.37. *If λ is a partition of n, then the dimension of V_λ is given by*

$$\dim V_\lambda = \frac{n!}{\prod(\textit{Hooklengths of } \lambda)},$$

that is $n!$ divided by the product of the hooklengths [7, 4.12].

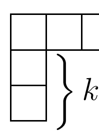
Definition 2.38. *The alternating representation of S_n is given by $\alpha \mapsto \operatorname{sgn}(\alpha)$ and corresponds to the partition $\lambda = (1, 1, \ldots, 1)$.*

Theorem 2.39. *For λ' the conjugate partition of λ, U' the alternating representation, and V_λ the representation corresponding to the partition λ, then*

$$V_{\lambda'} = V_\lambda \otimes U'.$$

Proof. This is found as Exercise 4.4c of [7], but we have not included this proof since it is similar to the proof of Theorem 2.50. ■

Theorem 2.40. *The k-th exterior power $\bigwedge^k V$ of the standard representation V of S_n corresponds to a hook*

$\lambda =$ 

with $n - k$ elements in the row and $k + 1$ elements in the column.

Proof. This is Exercise 4.6 of [7], with a hint in Exercise 4.43. ■

2.2 Irreducible Representations of $\operatorname{Cox}(B_n)$

We have discussed how the irreducible representations of S_n can be determined from Young diagrams. We shall now describe a similar process by which all the irreducible representations of $\operatorname{Cox}(B_n)$ can be obtained. This is the main result of [1]. We first introduce some notation for $\operatorname{Cox}(B_n)$, then continue with the algorithm in a manner similar to the construction for S_n.

Definition 2.41. *We define a positive transposition to be an element of* $\operatorname{Cox}(B_n)$ *of the form* $(i\ j)(-i\ -j)$*. In terms of* $\operatorname{Aut}(F_n)$*, it is the element defined by*

$$x_i \mapsto x_j; \qquad x_i^{-1} \mapsto x_j^{-1};$$
$$x_j \mapsto x_i; \qquad x_j^{-1} \mapsto x_i^{-1}.$$

A negative transposition in $\operatorname{Cox}(B_n)$ *will be an element of the form* $(i\ -i)$*. We shall also call negative transpositions sign changes, and often denote them* $\varepsilon_i = (i\ -i)$*. When we speak of permutations, we mean those elements of* $\operatorname{Cox}(B_n)$ *which can be written as products of positive transpositions.*

We shall have occasion to talk about S_n and $\operatorname{Cox}(B_n)$ as subgroups of $\operatorname{Aut}(F_n)$. When we refer to $S_n < \operatorname{Aut}(F_n)$ as a natural subgroup, we mean the subgroup of $\operatorname{Aut}(F_n)$ isomorphic to S_n consisting of permutations. That is to say the subgroup generated by $(1\ 2)(-1\ -2)$ and $(1\ 2\ \ldots n)(-1\ -2\ \ldots -n)$ or in the notation of $\operatorname{Aut}(F_n)$, by P and Q. Similarly, when we talk of $\operatorname{Cox}(B_n)$ we mean the subgroup of $\operatorname{Aut}(F_n)$ generated by P, Q, and $\sigma = \varepsilon_1$.

Definition 2.42. *Let* $\pi : \operatorname{Cox}(B_n) \to S_n$ *be the projection defined by* $\pi\big((i\ -i)\big) = 1$*, and* $\pi\big((i\ j)(-i\ -j)\big) = (i\ j)$*.*

Definition 2.43. *We can now define* $\det : \operatorname{Cox}(B_n) \to \mathbb{C}$ *by* $\det(\text{transposition}) = -1$*. Thus* $\det(\alpha) = \operatorname{sgn}(\pi(\alpha))$ *if* α *is a permutation, and* $\det\big((i\ -i)\big) = -1$*.*

The reason for the name det is that this map is equal to the determinant of the standard representation. We shall define the standard representation of $\operatorname{Cox}(B_n)$ in section 2.3.

Definition 2.44. *Let* $x = (\lambda, \mu)$ *be a pair of partitions such that* $|\lambda| + |\mu| = n$*. We shall call this a pair of partitions of* n*. We allow either of* λ*,* μ *to be empty.*

To construct the analog of a Young tableau for a pair of partitions $x = (\lambda, \mu)$, we start with a pair of Young diagrams, one for each of λ, μ. We then number the boxes with members of the set $\{\pm 1, \pm 2, \dots, \pm n\}$ with distinct magnitudes in each box. In other words we number the boxes as before, but allow entries to be either positive or negative.

We define the row group $\mathrm{Row}(x)$ to be those elements of $\mathrm{Cox}(B_n)$ which fix the rows of x as sets up to sign. Additionally the elements of $\mathrm{Row}(x)$ may not change the sign of an element of λ, but are allowed to change the sign of any entry in μ, . Similarly the column subgroup $\mathrm{Col}(x)$ is defined to be those permutations which fix the columns of x, and may change the sign of any entry in λ, but not of μ.

$$\mathrm{Row}(x) = \{\alpha \in S_n \mid \alpha \text{ fixes rows of } x \text{ up to sign and fixes signs of } \lambda\} \ ;$$
$$\mathrm{Col}(x) = \{\beta \in S_n \mid \beta \text{ fixes columns of } x \text{ up to sign and fixes signs of } \mu\} \ .$$

Now define $a, b, c \in \mathbb{C}\,\mathrm{Cox}(B_n)$ in the complex group algebra to be

$$\begin{aligned} a &= \sum_{\alpha \in \mathrm{Row}(x)} \alpha; \\ b &= \sum_{\beta \in \mathrm{Col}(x)} \det(\beta)\, \beta; \\ c &= aba. \end{aligned} \tag{6}$$

Notice that $c = aba$, not ab as before. Then $\mathbb{C}\,\mathrm{Cox}(B_n)\, c$ is an irreducible representation of $\mathrm{Cox}(B_n)$ [1, 2.6] with the action defined by

$$\alpha \cdot (\gamma c) = \alpha \gamma c \quad \text{ for } \gamma \in \mathbb{C}\,\mathrm{Cox}(B_n),\ \alpha \in \mathrm{Cox}(B_n).$$

Example 2.45. *As an example, consider the pair of partitions $x = \big((1), (2)\big)$. With the associated pair of Young tableaux, $\big(\boxed{1}, \boxed{2}\boxed{3}\big)$. We have*

$$\mathrm{Row}(x) = \big\langle (2\ 3)(-2\ -3)\,, (2\ -2)\,, (3\ -3)\,\big\rangle$$
$$\mathrm{Col}(x) = \big\langle (1\ -1)\,\big\rangle\,.$$

Then we calculate a, b, and c:

$$
\begin{aligned}
a &= (1) + (2\ {-2}) + (3\ {-3}) + (2\ {-2})\,(3\ {-3}) + (2\ 3)(-2\ {-3}) \\
&\quad + (2\ {-2})\,(2\ 3)(-2\ {-3}) + (3\ {-3})\,(2\ 3)(-2\ {-3}) \\
&\quad + (2\ {-2})\,(3\ {-3})\,(2\ 3)(-2\ {-3})\,; \\
b &= (1) - (1\ {-1})\,; \\
c &= 2\Big((1) - (1\ {-1}) + (2\ 3)(-2\ {-3}) - (1\ {-1})\,(2\ 3)(-2\ {-3}) + (2\ {-2}) \\
&\quad - (1\ {-1})\,(2\ {-2}) + (2\ {-2})\,(2\ 3)(-2\ {-3}) - (1\ {-1})\,(2\ {-2})\,(2\ 3)(-2\ {-3}) \\
&\quad + (3\ {-3}) - (1\ {-1})\,(3\ {-3}) + (3\ {-3})\,(2\ 3)(-2\ {-3}) - (1\ {-1})\,(3\ {-3})\,(2\ 3)(-2\ {-3}) \\
&\quad + (2\ {-2})\,(3\ {-3}) - (1\ {-1})\,(2\ {-2})\,(3\ {-3}) + (2\ {-2})\,(3\ {-3})\,(2\ 3)(-2\ {-3}) \\
&\quad - (1\ {-1})\,(2\ {-2})\,(3\ {-3})\,(2\ 3)(-2\ {-3})\Big).
\end{aligned}
$$

There are two other elements which, with c, form the basis for this representation. They are

$$
\begin{aligned}
c' &= (1\ 2)(-1\ {-2})\,c \\
&= 2\cdot\Big((1\ 2)(-1\ {-2}) + (3\ {-3})\,(1\ 2)(-1\ {-2}) \\
&\qquad - (2\ {-2})\,(1\ 2)(-1\ {-2}) - (2\ {-2})\,(3\ {-3})\,(1\ 2)(-1\ {-2}) + (1\ 3\ 2)(-1\ {-3}\ {-2}) \\
&\qquad + (3\ {-3})\,(1\ 3\ 2)(-1\ {-3}\ {-2}) - (2\ {-2})\,(1\ 3\ 2)(-1\ {-3}\ {-2}) \\
&\qquad - (2\ {-2})\,(3\ {-3})\,(1\ 3\ 2)(-1\ {-3}\ {-2}) + (1\ {-1})\,(1\ 2)(-1\ {-2}) \\
&\qquad + (1\ {-1})\,(3\ {-3})\,(1\ 2)(-1\ {-2}) - (1\ {-1})\,(2\ {-2})\,(1\ 2)(-1\ {-2}) \\
&\qquad - (1\ {-1})\,(2\ {-2})\,(3\ {-3})\,(1\ 2)(-1\ {-2}) + (1\ {-1})\,(1\ 3\ 2)(-1\ {-3}\ {-2}) \\
&\qquad + (1\ {-1})\,(3\ {-3})\,(1\ 3\ 2)(-1\ {-3}\ {-2}) \\
&\qquad - (1\ {-1})\,(2\ {-2})\,(3\ {-3})\,(1\ 3\ 2)(-1\ {-3}\ {-2})\Big),
\end{aligned}
$$

and

$$
\begin{aligned}
c'' &= (1\ 3)(-1\ -3)\, c \\
&= 2 \cdot \big((1\ 2\ 3)(-1\ -2\ -3) - (3\ -3)\,(1\ 2\ 3)(-1\ -2\ -3) \\
&\qquad + (2\ -2)\,(1\ 2\ 3)(-1\ -2\ -3) - (2\ -2)\,(3\ -3)\,(1\ 2\ 3)(-1\ -2\ -3) \\
&\qquad + (1\ 3)(-1\ -3) - (3\ -3)\,(1\ 3)(-1\ -3) + (2\ -2)\,(1\ 3)(-1\ -3) \\
&\qquad - (2\ -2)\,(3\ -3)\,(1\ 3)(-1\ -3) + (1\ -1)\,(1\ 2\ 3)(-1\ -2\ -3) \\
&\qquad - (1\ -1)\,(3\ -3)\,(1\ 2\ 3)(-1\ -2\ -3) + (1\ -1)\,(2\ -2)\,(1\ 2\ 3)(-1\ -2\ -3) \\
&\qquad - (1\ -1)\,(2\ -2)\,(3\ -3)\,(1\ 2\ 3)(-1\ -2\ -3) + (1\ -1)\,(1\ 3)(-1\ -3) \\
&\qquad - (1\ -1)\,(3\ -3)\,(1\ 3)(-1\ -3) + (1\ -1)\,(2\ -2)\,(1\ 3)(-1\ -3) \\
&\qquad - (1\ -1)\,(2\ -2)\,(3\ -3)\,(1\ 3)(-1\ -3) \big).
\end{aligned}
$$

Then for the generators P, Q, σ of $\operatorname{Cox}(B_3)$ *we have:*

$$
\rho_{(1\ 2)(-1\ -2)} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \rho_{(1\ 2\ 3)(-1\ -2\ -3)} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \quad \rho_{(1\ -1)} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}
$$

This example is the standard representation of $\operatorname{Cox}(B_3)$, *which we shall discuss in the next section.*

Definition 2.46. *The conjugate to a pair of partitions $x = (\lambda, \mu)$ is defined to be*

$$x' = (\mu', \lambda')$$

where λ', and μ' are the conjugate partitions of λ, μ respectively.

Theorem 2.47. *If (λ, μ) is a pair of partitions of n, define $V_{\lambda\mu}$ to be the representation of* $\operatorname{Cox}(B_n)$ *arising from (λ, μ). then the dimension of $V_{\lambda\mu}$ is given by*

$$\dim V_{\lambda\mu} = \frac{n!}{\prod(\mathit{Hooklengths\ of\ } \lambda) \prod(\mathit{Hooklengths\ of\ } \mu)}$$

that is $n!$ divided by the product of the hooklengths of both λ and μ [1, Eqn 3.5].

Lemma 2.48. *With λ, a, b as described above we have*

$$\mathbb{C}\,\mathrm{Cox}\,(B_n)\,bab \cong \mathbb{C}\,\mathrm{Cox}\,(B_n)\,aba$$

Proof. Consider the maps

$$\mathbb{C}\,\mathrm{Cox}\,(B_n)\,aba \xrightarrow{\cdot b} \mathbb{C}\,\mathrm{Cox}\,(B_n)\,bab$$

$$\mathbb{C}\,\mathrm{Cox}\,(B_n)\,bab \xrightarrow{\cdot a} \mathbb{C}\,\mathrm{Cox}\,(B_n)\,aba$$

given by right multiplication of a and b. We shall prove that both of the composites functions are multiplication by non-zero scalars. Some scalar multiple of a, and aba are idempotent, *i.e.*, $a^2 = n_a a$ and $(aba)^2 = n_{aba}aba$ for some $n_a, n_{aba} \in \mathbb{C} \setminus \{0\}$. From this we see

$$\begin{aligned}\mathbb{C}\,\mathrm{Cox}\,(B_n)\,ababa &= \mathbb{C}\,\mathrm{Cox}\,(B_n)\,\frac{1}{n_a}abaaba\\ &= \mathbb{C}\,\mathrm{Cox}\,(B_n)\,\frac{n_{aba}}{n_a}aba. \quad \blacksquare\end{aligned}$$

Definition 2.49. *The alternating representation of* $\mathrm{Cox}\,(B_n)$*, which is defined by* $\alpha \mapsto \det(\alpha)$*, and corresponds to the pair of partitions* $\lambda = (1, 1, \ldots, 1)$*,* $\mu = \emptyset$*.*

Theorem 2.50. *For* $x' = (\mu', \lambda')$ *conjugate to the pair of partitions* $x = (\lambda, \mu)$*, and* U' *the alternating representation of* $\mathrm{Cox}\,(B_n)$*, we have*

$$V_{\mu'\lambda'} = V_{\lambda\mu} \otimes U'.$$

Proof. We first note that for all $\alpha \in \mathrm{Row}(x)$, $\alpha \in \mathrm{Col}(x')$, and for $\beta \in \mathrm{Col}(x)$, $\beta \in \mathrm{Row}(x')$. Therefore, if a', b' correspond to x', and all sums are over $\alpha, \alpha' \in \mathrm{Row}(x)$ and $\beta, \beta' \in \mathrm{Col}(x)$ as necessary, we have

$$\begin{aligned}Wa'b'a' &= \sum \beta\alpha\beta'\ \det\alpha\\ &= \sum \beta\alpha\beta'\ \det\alpha\ (\det\beta\det\beta')^2\\ &= \sum \beta\alpha\beta'\ (\det\beta\det\beta')\ \det(\beta\alpha\beta');\end{aligned}$$

and

$$
\begin{aligned}
(Waba) \otimes U' &\cong (Wbab) \otimes U' \\
&= \sum \beta\alpha\beta' \left(\det\beta \det\beta'\right) \det(\beta\alpha\beta'). \qquad \blacksquare
\end{aligned}
$$

2.3 The Standard Representation of $\mathrm{Cox}\,(B_n)$

We now study the standard representation of $\mathrm{Cox}\,(B_n)$ in detail.

Definition 2.51. *The standard representation V of $\mathrm{Cox}\,(B_n)$ is a representation of degree n defined by the permutation representation of S_n, and the image of a negative transposition given by*

$$
(i\ -i) \mapsto \begin{bmatrix} 1 & 0 & \dots & & 0 \\ 0 & \ddots & & & \vdots \\ \vdots & & -1 & & \\ & & & \ddots & 0 \\ 0 & \dots & & 0 & 1 \end{bmatrix},
$$

being the identity matrix with the i-th diagonal replaced with -1. The standard representation is simply the action of $\mathrm{Cox}\,(B_n)$ on the standard basis of $\mathbb{C}^n$, and can be thought of as the group of symmetries of the n-cube $[-1,1]^n$ [2].

Theorem 2.52. *Let $x = (\lambda, \mu)$ be a pair of partitions of n of the form $\lambda = (1, 1, \dots, 1)$ (k terms), $\mu = (n-k)$. If V denotes the standard representation of $\mathrm{Cox}\,(B_n)$, then the representation derived from x is $\bigwedge^k V$.*

Proof. For x as described, the associated Young diagrams look like

$$
\lambda = \left.\begin{array}{|c|}\hline \ \\ \hline \ \\ \hline \ \\ \hline \end{array}\right\} k \text{ summands} \qquad \mu = \underbrace{\begin{array}{|c|c|c|c|}\hline \ & \ & \ & \ \\ \hline \end{array}}_{n-k \text{ summands}}
$$

which is to say that λ is a column, and μ is a row. Either may be empty ($k = 0, n$). When choosing a tableau we choose to number λ with the numbers $1, 2, \dots, k$. Clearly, $\mathrm{Row}(x) \cong \mathrm{Cox}\,(B_{n-k})$ and $\mathrm{Col}(x) \cong \mathrm{Cox}\,(B_k)$.

For convenience we let $W = \mathbb{C}\,\mathrm{Cox}\,(B_n)\,c$ with c as defined in (6) and define a set of "vectors" V_λ indexed by Young tableaux for λ, which we will show to be a basis for W. We will denote these by a k-tuple of values since λ is always a column. For example $V_{\begin{array}{|c|}\hline 1\\\hline 2\\\hline 3\\\hline\end{array}}$ will be written $V_{[1,2,3]}$.

To each element $\alpha \in \mathrm{Cox}\,(B_n)$ assign a Young tableau

$$\nu = \begin{array}{|c|}\hline \nu_1\\\hline \nu_2\\\hline \\\hline \nu_k\\\hline\end{array}$$

of shape λ, such that $\nu_i = \alpha(i)$, and define $V_\nu = V_{[\nu_1,\nu_2,\ldots,\nu_k]}$ to be $\alpha_\nu c$. Clearly the set of all V_ν span W. Any two α giving the same V_ν will differ (on the right) by a permutation in $\mathrm{Row}(x)$. We now observe that for $\alpha \in \mathrm{Row}(x)$, $\alpha c = c$ so they are in fact equal. Thus for each tableau ν we can choose a representative $\alpha_\nu \in \mathrm{Cox}\,(B_n)$.

Given a positive or negative transposition τ, the action of τ on V_ν is

$$\begin{aligned}\tau \cdot V_{[\nu_1,\nu_2,\ldots,\nu_k]} &= \tau \cdot \alpha_\nu c\\ &= (\tau^{-1}\alpha_\nu\tau)c\\ &= V_{[\tau(\nu_1),\tau(\nu_2),\ldots,\tau(\nu_k)]}\end{aligned}$$

We now prove that the V_ν satisfy the alternating property of the exterior powers. It suffices to show the equivalences

$$\begin{aligned}V_{[2,1,3,\ldots,k]} &= (1\ 2)(-1\ -2)\,c \\ &= -c \\ &= -V_{[1,2,\ldots,k]};\end{aligned} \qquad \begin{aligned}V_{[-1,2,\ldots,k]} &= (1\ -1)\,c\\ &= -c\\ &= -V_{[1,2,\ldots,k]}.\end{aligned}$$

Thus we need only concern ourselves with those $V_{[\nu_1,\nu_2,\ldots,\nu_k]}$ which have $0 < \nu_1 < \nu_2 < \cdots < \nu_k$ since all others are multiples of these. Also the V_λ are alternating under the action of $\mathrm{Cox}\,(B_n)$, and there is a natural set bijection

$$v_{\nu_1} \wedge v_{\nu_2} \wedge \cdots \wedge v_{\nu_k} \leftrightarrow V_{[\nu_1,\nu_2,\ldots,\nu_k]}$$

from the basis of $\bigwedge^k V$ to the $\{V_\lambda\}$. Since the V_λ are a basis for the representation corresponding to $V_{\lambda\mu}$ they are in fact equivalent representations. ■

2.4 Frobenius' Formula

We now return our attention to the symmetric group S_n to discuss Frobenius' Formula, and after which we derive a similar result for $\text{Cox}(B_n)$. This will allow us to understand how representations of the subgroups $\text{Cox}(B_n)$, S_n, and Σ_n interact. The proof of Frobenius' Formula, reproduced here, is the subject of section 4.3 of [7]. A few important results concerning symmetric polynomials have been moved to Appendix A.

Definition 2.53. *Define a generalized partition of n to be any k-tuple of positive integers which sum to n. In other words we do not require that they be weakly decreasing.*

Definition 2.54. *With this definition, we define a subgroup S_λ of S_n, often called a Young subgroup, for each generalized partition $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_k)$. The subgroup S_λ is the image of the injection*

$$S_{\lambda_1} \times S_{\lambda_2} \times \cdots \times S_{\lambda_k} \hookrightarrow S_n.$$

Definition 2.55. *We shall be have occasion to discuss coefficients of certain monomials in a polynomial, and therefore desire some notation. We shall denote the coefficient of the monomial $X^\lambda = x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_k^{\lambda_k}$ in a polynomial P by $[P]_\lambda$.*

Definition 2.56. *We now let $U_\lambda = \text{Ind}_{S_\lambda}^{S_n} 1$ be the representation induced by the trivial representation of the Young subgroup S_λ, and define $\psi_\lambda = \chi_{U_\lambda}$ to be the character of U_λ.*

Lemma 2.57. *If W is the trivial representation of H a subgroup of G then, for a conjugacy class C of G,*

$$\chi_{\operatorname{Ind} W}(C) = \frac{[G:H]}{|C|} \cdot |C \cap H| \,. \tag{7}$$

Proof. This is part b of Exercise 3.19 of [7]. If C is a conjugacy class of G, and $g \in C$ then by Theorem 2.16

$$\chi_{\operatorname{Ind} W}(g) = \sum_{g\sigma=\sigma} \chi_W(s^{-1}gs)$$

if $C \cap H$ decomposes into conjugacy classes $C_1, C_2, \ldots, C_k$ of H, then

$$\begin{aligned}
\chi_{\operatorname{Ind} W}(g) &= \sum_{i=1}^{k} \frac{|G|}{|H|}\frac{|C_i|}{|C|}\chi_W(s^{-1}gs) \\
&= [G:H]\sum_{i=1}^{k}\frac{|C_i|}{|C|}\,1 \qquad \text{Since } W \text{ is trivial} \\
&= [G:H]\frac{|C \cap H|}{|C|} \qquad \blacksquare
\end{aligned}$$

The representation V_λ contains the representation U_λ since right multiplication by b gives a surjection from $U_\lambda = Wa \twoheadrightarrow Wab = V_\lambda$.

Definition 2.58. *Given an n-tuple $i = (i_1, i_2, \ldots, i_n)$ such that $\sum_{j=1}^{n} j \cdot i_j = n$, we denote by C_i the conjugacy class consisting of i_j j-cycles.*

The character of U_λ, an induced representation, can be easily computed. The number of elements in C_i is seen to be

$$|C_i| = \frac{n!}{1^{i_1} i_1! 2^{i_2} i_2! \cdots n^{i_n} i_n!} \tag{8}$$

which we combine with (7) to give

$$\begin{aligned}
\psi_\lambda(C_i) &= \frac{1}{|C_i|}\,[S_n : S_\lambda] \cdot |C_i \cap S_\lambda| \\
&= \frac{1^{i_1} i_1! 2^{i_2} i_2! \cdots n^{i_n} i_n!}{n!} \cdot \frac{n!}{\lambda_1! \lambda_2! \cdots \lambda_k!} \cdot |C_i \cap S_\lambda|
\end{aligned}$$

To count $|C_i \cap S_\lambda|$, we write the p-th component of an element of S_λ as a product of $r_{p,1}$ 1-cycles, $r_{p,2}$ 2-cycles, *etc.*. Then $|C_i \cap S_\lambda|$ becomes

$$|C_i \cap S_\lambda| = \sum \prod_{p=1}^{k} \frac{\lambda_p!}{1^{r_{p,1}} r_{p,1}! 2^{r_{p,2}} r_{p,2}! \cdots k^{r_{p,k}} r_{p,k}!}$$

with the sum over all collections $\{r_{p,q} \mid 1 \le p \le n\}$ satisfying the equations

$$i_q = r_{1,q} + r_{2,q} + \cdots + r_{k,q},$$

$$\lambda_p = r_{p,1} + r_{p,2} + \cdots + r_{p,n}.$$

Simplifying we find that

$$\psi_\lambda(C_i) = \sum \prod_{q=1}^{n} \frac{i_q!}{r_{1,q}! r_{2,q}! \cdots r_{k,q}!}$$

with the sum over the same set of integers $\{r_{p,q}\}$. This sum is precisely the coefficient of the monomial $X^\lambda = x_1^{\lambda_1} x_2^{\lambda_2} \ldots x_k^{\lambda_k}$ in the polynomial

$$P^{(i)} = (x_1 + x_2 + \cdots + x_k)^{i_1} (x_1^2 + x_2^2 + \cdots + x_k^2)^{i_2} \cdots (x_1^n + x_2^n + \cdots + x_k^n)^{i_n}.$$

Thus we have the formula

$$\psi_\lambda(C_i) = \left[P^{(i)}\right]_\lambda .$$

To prove Frobenius' formula, we will compare these coefficients with the coefficients

$$\omega_\lambda(i) = \left[\Delta \cdot P^{(i)}\right]_\ell ,$$

where $\ell = (\lambda_1 + k - 1, \lambda_2 + k - 2, \ldots, \lambda_k)$ and $\Delta(x) = \prod_{i<j} (x_i - x_j)$ is the usual discriminant. Frobenius' formula, which we now state, is simply the assertion that $\chi_\lambda(C_i) = \omega_\lambda(i)$.

Theorem 2.59 (Frobenius' Formula)**.** *If λ is a partition of n and χ_λ the character of V_λ then*

$$\chi_\lambda(C_i) = \left[\Delta(x) \cdot \prod_j P_j(x)^{i_j}\right]_{(l_1, l_2, \ldots, l_k)}$$

where $P_j(x) = x_1^j, x_2^j, \ldots, x_k^j$, with Δ and ℓ as before [7, 4.10].

For a general symmetric polynomial P, there is a relation [7, p. 56].

$$[P]_\lambda = \sum_\mu K_{\mu\lambda} \left[\Delta \cdot P\right]_{(\mu_1+k-1,\mu_2+k-2,\ldots,\mu_k)}, \tag{9}$$

where the $K_{\mu\lambda}$ are universal constants known as Kostka numbers. They can be defined as the number of ways to fill a Young Diagram for μ with λ_1 1's, λ_2 2's *etc.* such that rows are weakly increasing and columns are strictly increasing. In particular $K_{\lambda\lambda} = 1$ and $K_{\mu\lambda} = 0$ for $\mu < \lambda$ under the lexicographical ordering defined in 2.28.

With these relations and $P^{(i)}$ substituted into (9) we find that

$$\psi_\lambda(C_i) = \sum_\mu K_{\mu\lambda}\omega_\mu(i) = \omega_\lambda(i) + \sum_{\mu>\lambda} K_{\mu\lambda}\omega_\mu(i). \tag{10}$$

The result of Theorem A.1 can be written in light of (8) as

$$\frac{1}{n!}\sum_i |C_i|\,\omega_\lambda(i)\omega_\mu(i) = \delta_{\lambda\mu}.$$

In other words the $\omega_\lambda(i)$ satisfy the same orthogonality relationships as the characters χ_λ.

Theorem 2.60. *If $\chi_\lambda = \chi_{V_\lambda}$ is the character of V_λ, then for any conjugacy class C_i, the relationship*

$$\chi_\lambda(C_i) = \omega_\lambda(i)$$

describes the character on that conjugacy class [7, 4.37].

Proof. Since U_λ contains V_λ,

$$\psi_\lambda = \sum_\mu n_{\lambda\mu}\chi_\mu, \quad n_{\lambda\mu} \in \mathbb{Z}, \tag{11}$$

with $n_{\lambda\lambda} \geq 1$, and all $n_{\lambda\mu} \geq 0$. This equation, together with (10) allows us to deduce several things. First

$$\omega_\lambda = \sum_\mu m_{\lambda\mu}\chi_\mu, \quad m_{\lambda\mu} \in \mathbb{Z},$$

and since the $\omega_\lambda(i)$ are orthonormal,

$$1 = (\omega_\lambda, \omega_\lambda) = \sum_\mu m_{\lambda\mu}^2$$

implies $\omega_\lambda = \pm\chi$ for some irreducible character.

We now fix λ and assume inductively that $\chi_\mu = \omega_\mu$ for all $\mu > \lambda$, then by (10)

$$\psi_\lambda = \omega_\lambda + \sum_{\mu>\lambda} K_{\mu\lambda}\chi_\mu.$$

Together with the linear independence of characters and (11), the only possibility is that $\psi_\lambda = \omega_\lambda$. ■

We shall now extend the definition of ψ_λ to any k-tuple $a = (a_1, a_2, \dots, a_k)$ of integers which sum to n. If one of the a_i is negative we define $\psi_a = 0$, otherwise we define $\psi_a = \psi_\lambda$ where λ is a reordering of a in weakly decreasing order. In the latter case, ψ_a is the character induced from the trivial representation by the inclusion $S_{a_1} \times S_{a_2} \times \cdots \times S_{a_k} \hookrightarrow S_n$. It can be shown that

$$\chi_\lambda = \sum_{\tau\in S_k} \operatorname{sgn}(\tau)\, \psi_{(\lambda_1+\tau(1)-1,\lambda_2+\tau(2)-2,\dots,\lambda_k+\tau(k)-k)}.$$

If we view ψ_a as a formal product $\psi_a = \psi_{a_1} \cdot \psi_{a_2} \cdots \psi_{a_k}$ which we can write

$$\chi_\lambda = |\psi_{\lambda_i+j-i}| = \begin{vmatrix} \psi_{\lambda_1} & \psi_{\lambda_1+1} & \cdots & \psi_{\lambda_1+k-1} \\ \psi_{\lambda_2-1} & \psi_{\lambda_2} & & \vdots \\ \vdots & & \ddots & \\ \psi_{\lambda_k-k+1} & \cdots & & \psi_{\lambda_k} \end{vmatrix}.$$

This formal product is in fact the character version of an "outer product" on representations. We define this product in the following way.

Definition 2.61. *Given non-negative integers n_i and representations V_i of S_{n_i} we shall denote by $V_1 \circ V_2 \circ \cdots \circ V_k$ the representation of S_n induced from the external* *tensor product representation* $V_1 \otimes V_2 \otimes \ldots \otimes V_k$ *of* $S_{n_1} \times S_{n_2} \times \cdots \times S_{n_k}$ *by inclusion of* $S_{n_1} \times S_{n_2} \times \cdots \times S_{n_k}$ *into* S_n. *This product is clearly associative and commutative.*

To easily decompose the outer product into irreducible factors, we need only consider the case of two factors V_1, V_2, both irreducible. If we let λ be a partition of ℓ, μ a partition of m, and allow ν to range over all partitions of $n = \ell + m$, then

$$V_\lambda \circ V_\mu = \sum_\nu N_{\lambda\mu\nu} V_\nu.$$

The coefficients $N_{\lambda\mu\nu}$ are given by the Littlewood–Richardson rule *i.e.*, the number of ways ν can be obtained from a strict μ-expansion of λ. See Theorem A.2 for more information.

Definition 2.62. *We shall use the notation* ${}^y\phi = y^{-1}\phi y$ *to indicate conjugation. For* ϕ *the character of a representation, we shall use* $\phi^G = \mathrm{Ind}^G \phi$ *to indicate the character of the representation induced to* G.

Theorem 2.63. *Let* ζ *be an irreducible character of* $(\mathbb{Z}_2)^n \cong N \triangleleft \mathrm{Cox}\,(B_n)$, *and* ψ *an irreducible character of the centralizer* $C = C_{S_n}(\zeta)$ *of* ζ. *Define a map* $\phi : NC \to \mathbb{C}$ *by* $\phi(nc) = \zeta(n)\psi(c)$. *Then the following hold [11, 1.1]*

1. ϕ *is an irreducible character of* NC, *and we write* $\phi = \zeta\psi$;
2. ϕ *induced to* $\mathrm{Cox}\,(B_n)$ *is an irreducible character* $\phi^{\mathrm{Cox}(B_n)}$ *of* $\mathrm{Cox}\,(B_n)$;
3. *if* $\phi_1 = \zeta_1\psi_1$ *and* $\phi_2 = \zeta_2\psi_2$, *and* ${\phi_2}^{\mathrm{Cox}(B_n)} = {\phi_2}^{\mathrm{Cox}(B_n)}$, *then* $\zeta_1 = \alpha^{-1}\zeta_2\alpha$ *and* $\psi_1 = \alpha^{-1}\psi_2\alpha$ *for some* $\alpha \in S_n$;
4. *Every irreducible character of* $\mathrm{Cox}\,(B_n)$ *can be obtained in this way.*

Proof. Since N is an abelian 2-group, we may assume that ζ is positive on the first p elements of N, and negative on the last $q = n - p$ elements. Then $C \cong S_p \times S_q$

1. is clear since $N \cap C = 1$;

2. Clearly $\phi(1) > 0$, and by Mackey's formula (Theorem 2.64) we see that

$$\left\langle \phi^{\mathrm{Cox}(B_n)}, \phi^{\mathrm{Cox}(B_n)} \right\rangle = \sum_{y \in \Delta} \left\langle \phi\big|_M, {}^y\phi\big|_M \right\rangle,$$

where $M = NC \cap {}^y(NC)$ and Δ are the usual double cosets for Mackey's formula. Since $N \lhd \mathrm{Cox}\,(B_n)$, $M \geq N$, so if $\left\langle \phi\big|_M, {}^y\phi\big|_M \right\rangle \neq 0$ then $\left\langle \phi\big|_N, {}^y\phi\big|_N \right\rangle \neq 0$, which implies that $\zeta = {}^y\zeta$, and so $y \in C_{\mathrm{Cox}(B_n)}(\zeta) = NC$. Therefore only the first summand ($y = 1$) in Mackey's formula is non-zero, and since ϕ is irreducible, $\phi^{\mathrm{Cox}(B_n)}$ is also.

3. Suppose $\phi_1^{\mathrm{Cox}(B_n)} = \phi_2^{\mathrm{Cox}(B_n)}$, and let $C_i = C_{S_n}(\zeta_i)$ for $i = 1, 2$. By the definition of induced character

$$\phi_i^{\mathrm{Cox}(B_n)}(\bar{n}) = |C_i|^{-1}\,\psi_i(1)\zeta_1^{\mathrm{Cox}(B_n)}(\bar{n})$$

for all $\bar{n} \in N$. Ergo $\zeta_1^{\mathrm{Cox}(B_n)}\big|_N = \zeta_2^{\mathrm{Cox}(B_n)}\big|_N$. If we suppose that $\zeta_1 \neq {}^\alpha\zeta_2$ for all $\alpha \in S_n$, then $\zeta_1 \neq {}^\beta\zeta_2$ for all $\beta \in \mathrm{Cox}\,(B_n)$ since $\mathrm{Cox}\,(B_n) = NS_n$. The fact that $N \lhd \mathrm{Cox}\,(B_n)$ implies

$$\left\langle {\zeta_1}^{\mathrm{Cox}(B_n)}, {\zeta_2}^{\mathrm{Cox}(B_n)} \right\rangle = \sum_{y \in \Delta} \langle \zeta_1, {}^y\zeta_2 \rangle \big|_N = 0.$$

Thus $\left\langle {\zeta_1}^{\mathrm{Cox}(B_n)}, {\zeta_1}^{\mathrm{Cox}(B_n)} \right\rangle = \left\langle {\zeta_1}^{\mathrm{Cox}(B_n)}\big|_N, \zeta_1 \right\rangle$, and by Frobenius' Reciprocity formula $\left\langle {\zeta_1}^{\mathrm{Cox}(B_n)}, {\zeta_2}^{\mathrm{Cox}(B_n)} \right\rangle = 0$, a clear contradiction. Hence there exists $\beta \in S_n$ such that $\zeta_1 = {}^\beta\zeta_2$. It is sufficient to assume that $\beta = 1$, and then show $\psi_1 = \psi_2$.

We may now write unambiguously ζ and C instead of ζ_i and C_i. Suppose that $\psi_1 \neq \psi_2$, and that $\psi_1, \psi_2, \ldots, \psi_k$ are the irreducible characters of C. Let

$\phi_i = \zeta\psi_i$. By Frobenius' reciprocity $\left\langle \zeta^{NC}, \phi_i \right\rangle = \psi_i(1)$ which we shall call a_i, so

$$\zeta^{NC} = a_1\phi_1 + a_2\phi_2 + \cdots + a_k\phi_k + \theta,$$

where θ is some character of NC such that $\langle \theta, \phi_i \rangle = 0$ for all i. But

$$|C| + \theta(1) = a_1^2 + a_2^2 + \cdots + a_k^2 + \theta(1) = \zeta^{NC}(1) = |C|,$$

so $\theta = 0$. By the transitivity of induction of characters it follows that

$$\zeta^{\mathrm{Cox}(B_n)} = \sum_{i=1}^{k} a_i {\phi_i}^{\mathrm{Cox}(B_n)} \tag{12}$$

We now proceed to compute $\left\langle \zeta^{\mathrm{Cox}(B_n)}, \zeta^{\mathrm{Cox}(B_n)} \right\rangle$. Let $t = [\mathrm{Cox}\,(B_n) : NC]$, and set $\mathrm{Cox}\,(B_n)\,/NC = \{\beta_1 NC, \beta_2 NC, \ldots, \beta_t NC\}$. Then

$$\zeta^{\mathrm{Cox}(B_n)}\big|_N = [NC : N] \sum_{i=1}^{t} {}^{\beta_i}\zeta.$$

Hence $\left\langle \zeta^{\mathrm{Cox}(B_n)}, \zeta^{\mathrm{Cox}(B_n)} \right\rangle = \left\langle \zeta^{\mathrm{Cox}(B_n)}\big|_N, \zeta \right\rangle = [NC : N] = |C|$. The contradiction now follows from (12) because

$$\begin{aligned}
|C| &= \left\langle \zeta^{\mathrm{Cox}(B_n)}, \zeta^{\mathrm{Cox}(B_n)} \right\rangle \\
&= \left\langle \sum_{i=1}^{k} a_i {\phi_i}^{\mathrm{Cox}(B_n)}, \sum_{i=1}^{k} a_i {\phi_i}^{\mathrm{Cox}(B_n)} \right\rangle \\
&\geq (a_1 + a_2)^2 + a_3^2 + \cdots + a_k^2 \\
&> a_1^2 + a_2^2 + \cdots + a_k^2 \\
&= |C|.
\end{aligned}$$

Thus $\phi_1 = \phi_2$ thereby completing the proof.

4. By part 2, $\phi^{\mathrm{Cox}(B_n)}$ determines an irreducible character ζ of $C \cong S_p \times S_q$. Thus $\phi^{\mathrm{Cox}(B_n)}$ determines a unique pair of partitions λ, μ of p and q respectively.

Since Cox (B_n) has precisely as many conjugacy classes as there are pairs of partitions which sum to n we see that we get all possible characters in this way. ■

Theorem 2.64 (Mackey's Formula)**.** *Let $H, K < G$, and let Δ be a set of double coset representatives such that*

$$G = \bigcup_{y \in \Delta} HyK$$

is a disjoint union. Then for a character ϕ of K, the formula

$$\phi^G\big|_H = \sum_{y \in \Delta} \left(\phi^y\big|_{({}^yK) \cap H} \right)^H$$

gives the restriction to H of the induced character ϕ^G [8, 6.4.1].

Proof. The subgroup H acts by right multiplication on the set $\{Ky\}$ of right cosets of K. Fix $y \in \Delta$ and let $h_{y,i}, h_{y,i}, \ldots, h_{y,i}$ be cosets representatives so that $Ky = \bigcup_{i=0}^{d_y} Kyh_{y,i}$ is a disjoint union. Then the set $\{y \mid y \in \Delta, 0 \leq i \leq d_y\}$ is a complete set of right coset representatives of K in G.

Suppose $z \in H$, then $z \in {}^yK$ if and only if $z \in ({}^yK) \cap H$, and so $\left({}^y\phi\big|_{({}^yK) \cap H} \right)(z) = {}^y\phi(z)$ for all y leading to

$$\begin{aligned} \phi^G(z) &= \sum_{y \in \Delta, i} \phi(yh_{y,i} z h_{y,i}^{-1} y^{-1}) \\ &= \sum_{y \in \Delta, i} \phi(h_{y,i} z h_{y,i}^{-1}) \\ &= \sum_{y \in \Delta, i} \left({}^y\phi\big|_{({}^yK) \cap H} \right)(h_{y,i} z h_{y,i}^{-1}) \\ &= \sum_{y \in \Delta} \left({}^y\phi\big|_{({}^yK) \cap H} \right)^H (z) \quad \blacksquare \end{aligned}$$

If we choose $H = NS_p \times S_q$ and $K = S_n$ then $HK = NS_n =$ Cox (B_n), so it follows that for an irreducible character $\phi = (\zeta\psi)^{\mathrm{Cox}(B_n)}$ of Cox (B_n), we have $\phi\big|_{S_n} = \psi^{S_n}$. Therefore, we obtain the following theorem from A.2.

Theorem 2.65. *Given a pair of partitions λ, μ of n, and ν a partition of n, if $V_{\lambda,\mu}$, V_ν are the corresponding irreducible representations of* $\mathrm{Cox}(B_n)$, *and S_n, then ν appears in* $\mathrm{Res}_{S_n}^{\mathrm{Cox}(B_n)} V_{\lambda\mu}$ *exactly $N_{\lambda\mu\nu}$ times, where $N_{\lambda\mu\nu}$ is the number of strict μ-expansions of λ.*

This result, together with Frobenius' formula, provides an easy condition to check which can determine that a given representation of $\mathrm{Cox}(B_n)$ cannot be extended to $\mathrm{Aut}(F_n)$. We will now describe this condition.

Definition 2.66. *A multiset is a set-like object in which order is unimportant, but multiplicity is important. Thus* $\{1,2,3\}$ *is the same as* $\{2,1,3\}$, *but not* $\{1,1,2,3\}$ *as multisets.*

Let $V = \bigoplus_{i=0}^{N} V_{\lambda_i,\mu_i}$ be a representation of $\mathrm{Cox}(B_n)$ which decomposes into a direct sum of irreducible representations indexed by pairs of partitions (λ_i, μ_i) of n. Define

$$W_i = \{\nu \mid \nu \text{ is a strict } \mu_i\text{-expansion of } \lambda_i\}$$

to be the multiset of all strict μ-expansions of λ (or all strict λ-expansions of μ). We let $W = \bigcup_{i=1}^{N} W_i$ be the multiset union of all the W_i. This represents the decomposition of V into irreducibles when restricted to S_n.

We now seek a representation of Σ_n which agrees with V on S_n. To do so we consider the partitions of $n+1$, and search for a multiset W' of partitions which restricts to W. The restriction of a partition of $n+1$ to partitions of n is accomplished by removing a box in turn from every row for which it is possible to remove a box 2.59.

If such a multiset exists, then the representation of Σ_n corresponding to this multiset is a representation V' which agrees with V on S_n. Thus V can be extended to a representation of $\langle \mathrm{Cox}\,(B_n)\,,\Sigma_n\rangle$ and possibly to $\mathrm{Aut}\,(F_n)$. If no such multiset exists, then such an extension is impossible and we need not perform further calculations.

The formulation above is entirely in terms of partitions, and so this condition is purely combinatorial.

In theory explicit matrices for the representations V, and V' could be used to construct explicit matrices for representation of $\langle \mathrm{Cox}\,(B_n)\,,\Sigma_n\rangle$, but in practice the situation is not straightforward. The representations V and V' can be easily made to agree on P (or Q) using the Jordan Canonical Form, but the author is unaware of simple methods to ensure agreement on both P and Q simultaneously. A similar problem is encountered in Section 5.1 when we consider the case $n = 2$.

3 Elementary Results for $\mathrm{Aut}(F_n)$

In this section we give some general results concerning the representation theory of $\mathrm{Aut}(F_n)$. Many of these results do not hold for $n = 2$, which case we will treat specially in section 5.1.

3.1 1-Dimensional Representations of $\mathrm{Aut}(F_n)$

We first consider the 1-dimensional representations of $\mathrm{Cox}(B_n)$. Since such a representation must be abelian, we consider the abelianization of $\mathrm{Cox}(B_n)$. From (P.3) we see that $Q^{n-1}P^{n-1} = 1$. Along with (P.2) we find that $P^{n-1} = Q$. Since $\sigma^2 = P^2 = 1$ in every instance there are only four possibilities for 1-dimensional representations of $\mathrm{Cox}(B_n)$. (This can also be easily seen from the hook length formula, since there are only 4 possible ways to have a single row or column.)

In keeping somewhat with the notation of [11] we shall call these the trivial 1, the long sign ξ, the short sign η, and the alternating det (which Mayer called ε) representations.

Table 3.1: 1-dimensional representations of $\mathrm{Cox}(B_n)$

n odd

	1	ξ	η	det
P	1	1	-1	-1
Q	1	1	1	1
σ	1	-1	1	-1
U	1			1

n even

	1	ξ	η	det
P	1	1	-1	-1
Q	1	1	-1	-1
σ	1	-1	1	-1
U	1			1

When $n \neq 2$ equation (U.8) simplified in an abelian context gives $U = 1$, and equation (U.10) implies that $U = P\sigma$. Thus the long and short sign representations cannot be extended from $\mathrm{Cox}(B_n)$ to $\mathrm{Aut}(F_n)$. When $n = 2$, there are in fact four 1-dimensional representations of $\mathrm{Aut}(F_n)$.

Theorem 3.1. *There are four 1-dimensional representations of $\mathrm{Cox}(B_n)$ of which only two (the trivial and alternating) can be extended to $\mathrm{Aut}(F_n)$ for $n \geq 3$.*

3.2 General Results for $\mathrm{Aut}(F_n)$

Many of the following results are found in [3], though most of the proofs are different owing to differences in development.

Definition 3.2. *The standard representation of $\mathrm{Aut}(F_n)$ is defined by the standard representation of $\mathrm{Cox}(B_n)$ and the image of U*

$$U \mapsto \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}.$$

We will denote this representation V. Note that V is irreducible since $V\big|_{\mathrm{Cox}(B_n)}$ is irreducible.

Theorem 3.3. *If the image of U under a homomorphism $\phi : \mathrm{Aut}(F_n) \to G$ from $\mathrm{Aut}(F_n)$ with $n \geq 3$ is central, then ϕ has image of cardinality at most 2. (This is a slight generalization of Lemma 1 of [3].)*

Proof. If U is central then (U.8) implies $U = 1$. Adding the stipulation that $U = 1$ to equation (U.10) gives us the relation

$$\begin{aligned} 1 &= U^{-1}PUP\sigma U\sigma P\sigma \\ &= PP\sigma\sigma P\sigma \\ &= P\sigma \end{aligned}$$

and indeed $P = \sigma$ since both are of order 2. From equation (P.7) we see that

$$\begin{aligned} 1 &= QP\sigma(QP)^{-1}\sigma \\ &= Q(QP)^{-1}P \\ &= QPQ^{-1}P, \end{aligned}$$

which is to say that $\sigma = P \rightleftarrows Q$, using $\rightleftarrows$ as before to signify "commutes with". Then equation (P.3) gives

$$\begin{aligned} 1 &= (QP)^{n-1} \\ &= Q^{n-1}P^{n-1}. \end{aligned}$$

Thus $Q = P^{n-1}$ since Q has order n, and $\phi(\mathrm{Aut}\,(F_n))$ is contained in $\phi(\langle P \rangle) \cong \mathbb{Z}_2$. In fact the only surjection onto $\mathbb{Z}_2$ is the map determined by sending $P \mapsto -1$ and $U \mapsto 1$ which is the composition of the determinant and the natural map from $\mathrm{Aut}\,(F_n)$ to $\mathrm{GL}_n(\mathbb{Z})$. ■

Theorem 3.4. *Let $n \geq 3$ and let ϕ be a homomorphism from* $\mathrm{Aut}\,(F_n)$ *to an arbitrary group G. If ϕ is not injective on S_n, then ϕ factors through* det *[3, Prop 1].*

Proof. Let $K = \ker \phi|_{S_n}$. If K is not trivial or equal to S_n, then K must be either A_n the alternating group, or in the case $n = 4$, isomorphic to the Klein 4-group $\mathbb{Z}_2 \times \mathbb{Z}_2$, since these are the only normal subgroups of S_n. In either case the element

$$Q^2PQ^{-2}P = \begin{cases} (1\ 3\ 2)\,(-1\ -3\ -2) & n = 3 \\ (1\ 2)(-1\ -2)\,(3\ 4)(-3\ -4) & n > 3 \end{cases}$$

is in K. This is equivalent to adding the relation that $P \rightleftarrows Q^2$. Combined with equation (U.1) it follows immediately that $P \rightleftarrows U$. Equation (U.10) tell us that

$$U = PUP\sigma U\sigma P\sigma$$
$$U = U\sigma U\sigma P\sigma$$
$$1 = \sigma U\sigma P\sigma$$
$$1 = U\sigma P.$$

We see that $P \rightleftarrows \sigma \rightleftarrows U$, and from (P.7) that $\sigma \rightleftarrows Q$, so that σ is central. Consider now equation (U.6) from which we get the following equivalent commutator relations:

$$U \rightleftarrows PQ^{-1}\sigma U\sigma QP$$
$$U \rightleftarrows PQ^{-1}UQP$$
$$U \rightleftarrows PQ^{-1}\sigma PQP \quad \text{replacing } U \text{ with } \sigma P$$
$$U \rightleftarrows PQ^{-1}\sigma PQP$$
$$U \rightleftarrows Q^{-1}\sigma PQ \quad \text{Since } U \rightleftarrows P$$
$$U \rightleftarrows Q^{-1}PQ. \quad \text{Since } \sigma \text{ is central.}$$

We combine this with (U.2) to see that $U \rightleftarrows QP$, and therefore with Q. Thus U is central and Theorem 3.3 applies. ■

Corollary 3.5. *The image of* $\mathrm{Aut}\,(F_n)$ *($n \geq 3$) under a homomorphism ϕ is trivial if and only if $\phi(P) = 1$ [3, Prop 2].*

Proof. This follows from noticing that the image of det is generated by $\det P$. ■

Corollary 3.6. *The group* $\mathrm{Aut}\,(F_n)$ *is the normal closure of the positive transposition P which has order 2. Hence any quotients of* $\mathrm{Aut}\,(F_n)$ *are also the normal closure of an element of order 2 [3, Coro 1].*

Theorem 3.7. *There is a subgroup* $\Sigma = \Sigma_n$ *of* $\operatorname{Aut}(F_n)$ *isomorphic to* S_{n+1} *which intersects* $\operatorname{Cox}(B_n) < \operatorname{Aut}(F_n)$ *in the natural copy of* S_n *[3].*

Proof. We define an automorphism a_n below and then show that $\langle S_n, a_n \rangle \cong S_{n+1}$.

$$
\begin{aligned}
a_n :\ x_1 &\mapsto x_1 x_n^{-1} \\
x_2 &\mapsto x_2 x_n^{-1} \\
x_3 &\mapsto x_3 x_n^{-1} \\
&\vdots \\
x_{n-1} &\mapsto x_{n-1} x_n^{-1} \\
x_n &\mapsto x_n^{-1}
\end{aligned}
$$

We denote by $a_i = (i, i+1)$ for $1 \leq i \leq n-1$. Note that $a_n^2 = 1$ and a_n commutes with all a_i except a_{n-1}. In the latter case $a_{n-1}a_n a_{n-1} = a_n a_{n-1} a_n$. Thus

$$\Sigma = \left\langle a_1, a_2, \ldots, a_n \,|\, a_i^2 = 1;\ a_i a_j = a_j a_i \text{ if } |i-j| > 1;\ (a_i a_j)^3 = 1 \text{ if } |i-j| = 1 \right\rangle.$$

Which is precisely the presentation of S_{n+1} [4, pg.219]. Note that a_n is given by the formula $a_n = Q(QUP)^{n-1}\sigma Q^{-1}$. ■

Theorem 3.8. *The group* $\mathrm{GL}_{n-1}(\mathbb{Z})$ *does not contain a copy of* S_{n+1}*.*

Proof. See for example, Exercise 4.14 p. 50 of [7] ■

Corollary 3.9. *For* $n \geq 3$ *any representation of* $\operatorname{Aut}(F_n)$ *of degree less than* n *must be trivial, isomorphic to* $\mathbb{Z}_2$*, or have non-integer entries in* U*.*

Proof. Any integer representation of $\operatorname{Aut}(F_n)$ of degree less than n maps into $\mathrm{GL}_{n-1}(\mathbb{Z})$, which does not contain a copy of S_{n+1}, and therefore its image must factor through the determinant map. ■

Definition 3.10. *It will be convenient to denote by $U_{i,j}$ the automorphism $U_{i,j} : x_i \mapsto x_i x_j$ that sends x_i to $x_i x_j$, and sends the rest of the generators to themselves.*

Theorem 3.11. *Let ρ be a representation of* $\mathrm{Aut}\,(F_n)$ *for $n \geq 3$. If σ is represented centrally then ρ is finite.*

Proof. Milnor [13] gives a presentation for $\mathrm{SL}_n(\mathbb{Z})$ for $n \geq 3$. There are $n(n-1)$ generators, $e_{i,j}$ (with $i \neq j$) with the relations

$$\left(e_{12}e_{21}^{-1}e_{12}\right)^4 = 1; \tag{SL.1}$$

$$[e_{ij}, e_{kl}] = 1 \quad \text{if } j \neq k,\ i \neq l; \tag{SL.2}$$

$$[e_{ij}, e_{jk}] = e_{ik} \quad \text{if } i, j, k \text{ are distinct.} \tag{SL.3}$$

From (U.9) we see immediately that $U^2 = 1$ since σ is central. We now verify that the relationship (SL.2) is satisfied by each $U_{i,j}$ in $\mathrm{Aut}\,(F_n)$. Let $i \neq l$ and $j \neq k$. $[U_{i,j}, U_{k,l}]$ is then given by

$$\begin{array}{lllll} x_i \xrightarrow{U_{i,j}} & x_i x_j \xrightarrow{U_{k,l}} & x_i x_j \xrightarrow{U_{i,j}^{-1}} & x_i \xrightarrow{U_{k,l}^{-1}} & x_i \\ x_k \xrightarrow{U_{i,j}} & x_k \xrightarrow{U_{k,l}} & x_k x_l \xrightarrow{U_{i,j}^{-1}} & x_k x_l \xrightarrow{U_{k,l}^{-1}} & x_k. \end{array}$$

Likewise, the relationship (SL.3) is seen to be satisfied in $\mathrm{Aut}\,(F_n)$ by the $U_{i,j}$. Let i, j, k be distinct, then $[U_{i,j}, U_{j,k}]$ is

$$\begin{array}{lllll} x_i \xrightarrow{U_{i,j}} & x_i x_j \xrightarrow{U_{j,k}} & x_i x_j x_k \xrightarrow{U_{i,j}^{-1}} & x_i x_j^{-1} x_j x_k \xrightarrow{U_{j,k}^{-1}} & x_i x_k \\ x_j \xrightarrow{U_{i,j}} & x_j \xrightarrow{U_{j,k}} & x_j x_k \xrightarrow{U_{i,j}^{-1}} & x_j x_k \xrightarrow{U_{j,k}^{-1}} & x_j. \end{array}$$

We now note that

$$U_{1,2} = U;$$

$$U_{2,1} = P U_{1,2} P.$$

We will use these together with $U^2 = 1$ to show that the $U_{i,j}$ satisfy (SL.1):

$$\begin{aligned}\left(U_{1,2}U_{2,1}^{-1}U_{1,2}\right)^4 &= \left(U(PUP)^{-1}U\right)^4 \\ &= (UPUPU)^4 \\ &= (UPUPUUPUPU)^2 \\ &= 1^2 = 1.\end{aligned}$$

The $U_{i,j}$ matrices also satisfy the relationship $U_{i,j}^2 = 1$ for all i, j since they are all conjugate to $U_{1,2} = U$. This relationship, along with those previously shown, prove that the elements $U_{i,j}$ generate $\mathrm{SL}_n(\mathbb{Z}_2)$.

Thus the group $\mathrm{Aut}\,(F_n)\,/\langle\sigma\rangle$ contains $\mathrm{SL}_n(\mathbb{Z}_2)$ as a subgroup

$$\mathrm{SL}_n(\mathbb{Z}_2) \cong \langle\!\langle U \rangle\!\rangle \subset \mathrm{Aut}\,(F_n)\,/\langle \sigma = 1\rangle.$$

Since P, and Q, have finite order, it suffices to show that $\mathrm{SL}_n(\mathbb{Z})/e_{12}^2$ is finite. This is a result of Mennicke [12] for $n \geq 3$. ■

Lemma 3.12. *Adding the relation $\sigma \rightleftarrows Q$ or $\sigma \rightleftarrows P$ is equivalent to stipulating that σ be central for $n \geq 3$.*

Proof. Equation (P.7) gives immediately the equivalence of $\sigma \rightleftarrows Q$ and $\sigma \rightleftarrows P$. Then equation (U.3) shows that $U \rightleftarrows \sigma$. ■

Example 3.13. *Let ρ be a representation of* $\mathrm{Aut}\,(F_3)$. *The additional stipulation that σ be represented centrally under ρ implies that ρ has order less than or equal to* 336 *for $n = 3$ as calculated by GAP. For $n = 4$, the size is* 40320, *for $n = 5$ the size is* 19998720.

Theorem 3.14. *Let $\Sigma <$ $\mathrm{Aut}\,(F_n)$ be as in Theorem 3.7. If ϕ :* $\mathrm{Aut}\,(F_n) \to G$ *is a homomorphism such that $\phi|_\Sigma$ has non-trivial kernel, then ϕ factors through* $\det : \mathrm{Aut}\,(F_n) \to \mathbb{Z}_2$ *[3, Theorem B].*

Proof. Let K be the kernel of $\phi|_\Sigma$. If $K = A_{n+1}$ then $K \cap S_n = A_n$, so the result follows from Theorem 3.4. When $n = 3$ there is also the possibility namely $K \cong \mathbb{Z}_2 \times \mathbb{Z}_2$ the Klein 4 group. This group consists of the elements

$$\{1, (1\ 2)(3\ 4)\,, (1\ 3)(2\ 4)\,, (1\ 4)(2\ 3)\}\,.$$

Recall that in our case the element $(3\ 4)$ is actually $a_3 = Q(QUP)^2\sigma Q^{-1}$, so $PQ(QUP)^2\sigma Q^{-1} \in K$. Call this element α, then α has order 2, and $\alpha(U_{2,3})\alpha = U_{1,3}^{-1}$. From this we see that $\phi(U_{2,3}) = \phi(U_{1,3}^{-1})$. Considering the commutator

$$\begin{aligned} U_{2,3} &= [U_{2,1}, U_{1,3}] \\ &= (U_{1,2}U_{1,3}U_{1,2}^{-1})U_{1,3}^{-1} \end{aligned}$$

we see that $\phi(U_{1,2}U_{1,3}U_{1,2}^{-1}) = 1$, and in fact $\phi(U_{1,3}) = 1$. We then apply Theorem 3.3 to arrive at the result. ■

Theorem 3.15. *Consider a semi-direct product $N \rtimes S_n$ with N abelian and suppose one of the following conditions is true:*

1. *$n \geq 5$;*
2. *$n = 4$ and N does not contain an S_n-invariant subgroup isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$;*
3. *$n = 3$ and N does not contain an S_n-invariant subgroup isomorphic to $\mathbb{Z}_3$.*

If $\phi : N \rtimes S_n \to G$ is a homomorphism with non-trivial kernel K, then either $K \cap S_n$ or $K \cap N$ is non-trivial [3, Lemma 4].

Proof. If $K \cap S_n$ and $K \cap N$ are both trivial, then $\phi(N) \cong N$ and $\phi(S_n) \cong S_n$ must intersect. This intersection must be normal in $\phi(S_n)$ since $N \lhd N \rtimes S_n$. If $n \geq 5$ or $n \leq 2$ then there are no such normal subgroups. In the cases $n = 3$ and $n = 4$, the only possible subgroups are those described. The intersection, however, must be of the form $\phi(N_0) \cong N_0$, and all possible subgroups have been precluded by hypothesis. ■

Theorem 3.16. *Suppose that a homomorphism $\phi : \mathrm{Aut}\,(F_n) \to G$ is not injective on $\mathrm{Cox}\,(B_n) \cong \mathbb{Z}_2^n \rtimes S_n$, then the image of ϕ is either finite or isomorphic to $\mathrm{PGL}_n(\mathbb{Z})$ [3, Prop C].*

Proof. If ϕ is not injective on S_n then $\phi(\mathrm{Aut}\,(F_n))$ is finite by Theorem 3.4. Now suppose that ϕ is injective on S_n. The only proper non-trivial S_n-invariant subgroups of $N \cong \mathbb{Z}_2^n$ are $\langle z \rangle$ for $z = \varepsilon_1\varepsilon_2 \ldots \varepsilon_n$ and $H = \langle \varepsilon_i\varepsilon_j \rangle$ with $i < j$. In the case of $n = 4$, neither of these is isomorphic to $\mathbb{Z}_2 \times \mathbb{Z}_2$, so we can apply the result of Theorem 3.15 to show that $K = \ker \phi|_N$ is non-trivial. Since K is normal in $\mathrm{Cox}\,(B_n)$, it must be one of the two subgroups listed above.

If $K = N$ or $K = H$ then $(P\sigma)^2 = \varepsilon_1\varepsilon_2$ is in K. This is equivalent to adding the relation $P \rightleftarrows \sigma$. We can now apply Theorem 3.12 to see that the image of ϕ is finite.

Let us denote as before by $U_{i,j}$ the element of $\mathrm{Aut}\,(F_n)$ which sends x_i to x_ix_j, and leaves the rest unchanged. Similarly we shall use $U^{i,j}$ to denote the automorphism which sends x_i to x_jx_i.

If $K = \langle z \rangle$ then $\phi(z) = 1$, and then the identity $zU_{i,j}z = U^{i,j}$ implies that $\phi(U_{i,j}) = \phi(U^{i,j})$ for all i, j. Adding this relation to the presentation of $\mathrm{Aut}\,(F_n)$ gives the presentation of $\mathrm{GL}_n(\mathbb{Z})$. Thus ϕ factors through $\mathrm{GL}_n(\mathbb{Z})$. This is the same as acting on the abelianization of F_n. Since $\phi(z)$ is in the center of $\mathrm{GL}_n(\mathbb{Z})$, ϕ actually factors through $\mathrm{PGL}_n(\mathbb{Z})$. Since all non-trivial normal subgroups of $\mathrm{PGL}_n(\mathbb{Z})$ have finite index [12] the image of ϕ is either finite of isomorphic to $\mathrm{PGL}_n(\mathbb{Z})$. ■

If A is a permutation matrix, then $A^{-1} = A^T$, that is, the inverse and the transpose are the same. In particular $\left(A^{-1}\right)^T = A$. If A is a matrix in the standard representation of $\operatorname{Cox}(B_n)$, then it is either a permutation matrix or a diagonal matrix with entries of ± 1, so the $A^{-1} = A^T$ for all A in the standard representation of $\operatorname{Cox}(B_n)$.

Applying the contragredient (inverse transpose) operator to all the defining relations of $\operatorname{Aut}(F_n)$ preserves the order in which the equations are written, since both inverse and transpose reverse the order. If the matrices P, Q, σ satisfy $(A^{-1})^T = A$, then $\left(U^{-1}\right)^T$ satisfies precisely the same relationships as U.

Theorem 3.17. *The standard representation V of* $\operatorname{Cox}(B_n)$ *extends to two different representations, the standard representation of* $\operatorname{Aut}(F_n)$*, and the inverse transpose of the standard representation:*

$$U \xrightarrow{V} \begin{bmatrix} 1 & 1 & \dots & 0 \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix} \qquad U \xrightarrow{\left(V^{-1}\right)^T} \begin{bmatrix} 1 & 0 & \dots & 0 \\ -1 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \end{bmatrix}$$

Also all exterior powers of the standard representation of $\operatorname{Cox}(B_n)$ *will have inverse transpose representations when extended to* $\operatorname{Aut}(F_n)$*.*

4 Commutators

We shall use as our definition for commutators $[x, y] = xyx^{-1}y^{-1}$ and $G' = [G, G]$.

Definition 4.1. *Suppose N is a subgroup of G, then N is said to be characteristic in G if N is invariant under* $\mathrm{Aut}(G)$*. If N is characteristic in G, then N must necessarily be normal in G.*

Theorem 4.2. *For any N characteristic in G, the projection $\pi : \mathrm{Aut}(G) \to \mathrm{Aut}\,(G/N)$ is a homomorphism of groups, though it may not be onto.*

Lemma 4.3. *Let $N = \big[F_n, [F_n, F_n]\big]$. Then N is characteristic in F_n.*

Proof. By definition, N is normally generated by all elements of the form

$$\big[a, [b, c]\big] = a(bcb^{-1}c^{-1})\, a^{-1}(cbc^{-1}b^{-1}).$$

Then for $\alpha \in \mathrm{Aut}(G)$ acting on the right we have

$$\begin{aligned}(\big[a, [b, c]\big])\alpha &= \big(a(bcb^{-1}c^{-1})\, a^{-1}(cbc^{-1}b^{-1})\big)\alpha \\ &= (a)\alpha\big((b)\alpha(c)\alpha(b)^{-1}\alpha(c)^{-1}\alpha\big)\,(a)^{-1}\alpha\big((c)\alpha(b)\alpha(c)^{-1}\alpha(b)^{-1}\alpha\big) \\ &= \big[(a)\alpha, [(b)\alpha, (c)\alpha]\big] \in N. \quad \blacksquare\end{aligned}$$

Note that the above proof is valid for $[F_n, F_n]$ and other similar subgroups as well.

We now cite some useful facts about commutators from [10]. Let $a, b, c \in F_n$ in the following equations.

$$[a, b]cN = c[a, b]N \quad \text{in } F_n/N. \tag{C.1}$$

$$[b, a]N = [a, b]^{-1}N \quad \text{in } F_n/N. \tag{C.2}$$

In general we have

$$
\begin{aligned}
[a,b][a,c]\big[[c,a],b\big] &= (aba^{-1}b^{-1})(aca^{-1}c^{-1})(cac^{-1}a^{-1}baca^{-1}c^{-1}b^{-1}) \\
&= abca^{-1}c^{-1}b^{-1} \\
&= [a,bc].
\end{aligned}
$$

Since $[[c,a],b] \in N$, in F_n/N we find

$$[a,bc]N = [a,b][a,c]N. \tag{C.3}$$

Similarly it can be shown that

$$[ab,c]N = [a,c][b,c]N, \tag{C.4}$$

from which it follows easily that in F_n/N

$$[a,b^2]N = [a,b]^2N; \tag{C.5}$$

$$[a,b][a,b^{-1}]N = [a,bb^{-1}]N = N. \tag{C.6}$$

Thus in $\operatorname{Aut}(F_n)/N$

$$[a,b]^{-1}N = [a,b^{-1}]N \tag{C.7}$$

$$[a^r,b^s]N = [a,b]^{r\cdot s}N. \tag{C.8}$$

This last fact is proved by induction. Now we consider

$$
\begin{aligned}
x_i^\varepsilon x_j^r N &= x_j^r x_i^\varepsilon x_i^{-\varepsilon} x_j^{-r} x_i^\varepsilon x_j^r N \\
&= x_j^r x_i^\varepsilon [x_i,x_j]^{\varepsilon r} N \quad \text{ in } F_n/N.
\end{aligned} \tag{C.9}
$$

Lemma 4.4. *Elements of F_n/N can be written uniquely as*

$$x_1^{m_1}x_2^{m_2}\dots x_n^{m_n}[x_1,x_2]^{k_{1,2}}[x_1,x_3]^{k_{1,3}}\cdots[x_1,x_n]^{k_{1,n}}[x_2,x_3]^{k_{2,3}}\cdots[x_{n-1},x_n]^{k_{n-1,n}}N \tag{13}$$

Proof. We shall proceed by induction on the length of a word w. If the length of w is 1 then $w = x_i^{\pm 1}$ which clearly has the desired form. Now suppose the result is true for words of length $k < m$ with $m \geq 2$ and let w be a word of length m. Then w can be written $w = x_i^{\varepsilon_i} w'$ where $\varepsilon_i = \pm 1$ and w' is a word of length $m - 1$. From the induction hypothesis $w'N = w''N$ with w'' in the form of (13).

Using (C.9) we have the following in F_n/N:

$$\begin{aligned} x_i^{\varepsilon} x_1^{a_1} x_2^{a_2} \cdots x_{i-1}^{a_{i-1}} &= x_1^{a_1} x_2^{a_2} \cdots x_{i-1}^{a_{i-1}} x_i^{\varepsilon} [x_i, x_1]^{\varepsilon a_1} [x_i, x_2]^{\varepsilon a_2} \cdots [x_i, x_{i-1}]^{\varepsilon a_{i-1}} \\ &= x_1^{a_1} x_2^{a_2} \cdots x_{i-1}^{a_{i-1}} x_i^{\varepsilon} [x_1, x_i]^{-\varepsilon a_1} [x_2, x_i]^{-\varepsilon a_2} \cdots [x_{i-1}, x_i]^{-\varepsilon a_{i-1}}. \end{aligned}$$

since $[a, b]$ commutes with x_i^r. Thus for a word of the form $x_i^{\varepsilon} w''$

$$\begin{aligned} x_i^{\varepsilon} w'' &= x_i^{\varepsilon} x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n} [x_1, x_2]^{k_{1,2}} \cdots [x_{n-1}, x_n]^{k_{n-1,n}} \\ &= x_1^{a_1} \cdots x_{i-1}^{a_{i-1}} x_i^{\varepsilon + a_i} x_{i+1}^{a_{i+1}} \cdots x_n^{a_n} [x_1, x_2]^{k_{1,2}} \cdots [x_1, x_i]^{k_{1,i} - \varepsilon a_1} \cdots [x_1, x_n]^{k_{1,n}} \\ &\quad [x_2, x_3]^{k_{2,3}} \cdots [x_2, x_i]^{k_{2,i} - \varepsilon a_2} \cdots [x_{n-1}, x_n]^{k_{n-1,n}} \end{aligned}$$

as required. This completes the proof of Lemma 4.4 ■

Theorem 4.5. *Let* $\alpha \in \operatorname{Aut}(F_n)$, *and write*

$$(x_1)\alpha/N = x_1^{a_{11}} x_2^{a_{12}} \cdots [x_{n-1}, x_n]^{a_{1m}}$$

$$(x_2)\alpha/N = x_1^{a_{21}} x_2^{a_{22}} \cdots [x_{n-1}, x_n]^{a_{2m}}$$

$$\vdots$$

$$([x_{n-1}, x_n])\alpha/N = x_1^{a_{m1}} x_2^{a_{m2}} \cdots [x_{n-1}, x_n]^{a_{mm}}$$

where $m = \dfrac{n(n+1)}{2}$. *Then the map*

$$\alpha \mapsto R(\alpha) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mm} \end{bmatrix}$$

is a representation of Aut (F_n). *This representation is the standard representation* V *direct sum the second exterior power of the standard representation:* $R = V \oplus \bigwedge^2 V$.

Proof. To prove that it is a representation, it suffices to show that $R(\alpha\beta) = R(\alpha)R(\beta)$. We first note a few things about the map which will simplify our calculations.

Consider x_i, a generator of F_n. The images of x_i under P, Q, σ, and U, are easily seen to not contain any commutators. Likewise the image of a commutator $[x_i, x_j]$ will consist entirely of commutators. This also follows from the fact that $[F_n, F_n]$ is characteristic. Thus the generators P, Q, σ, U of Aut (F_n) are block diagonal and so $R(\alpha)$ has the form

$$R(\alpha) = \begin{bmatrix} M_1(\alpha) & 0 \\ 0 & M_2(\alpha) \end{bmatrix}$$

for any α in Aut (F_n). Here $M_1(\alpha)$ has degree n and $M_2(\alpha)$ has degree $\binom{n}{2}$.

We now turn our attention to the first block $M_1(\alpha)$ of the matrix.

Suppose that α, β are automorphisms, then the i, j-th element of the matrix $R(\alpha\beta)$ will be the exponent of the x_j in the image $(x_i)\alpha\beta$ of x_i under the composition $\alpha\beta$.

$$\begin{aligned}(x_i)\alpha\beta/N &= (x_1^{a_{i1}} x_2^{a_{i2}} \cdots x_n^{a_{in}})\beta \\ &= ((x_1)\beta)^{a_{i1}}((x_2)\beta)^{a_{i2}} \cdots ((x_n)\beta)^{a_{in}} \\ &= \left(x_1^{b_{11}} x_2^{b_{12}} \cdots x_n^{b_{1n}}\right)^{a_{i1}} \left(x_1^{b_{21}} x_2^{b_{22}} \cdots x_n^{b_{2n}}\right)^{a_{i2}} \cdots \left(x_1^{b_{n1}} x_2^{b_{n2}} \cdots x_n^{b_{nn}}\right)^{a_{in}}\end{aligned}$$

Since x_i, x_j commute up to a commutator, we see that the j-th power must be

$$R(\alpha\beta)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

which is precisely the definition of matrix multiplication. It is not difficult to see that this is the standard representation, since it is the action of Aut (F_n) on the abelianization of F_n.

As we consider the block $M_2(\alpha)$, we shall use quadruple subscripts instead of double on the exponents. For example

$$[x_i, x_j]\alpha = [x_1, x_2]^{a_{ij12}} \cdots [x_{n-1}, x_n]^{a_{i,j,n-1,n}}.$$

We first consider the action of β on an arbitrary commutator

$$([x_i, x_j])\,\beta = [x_1^{b_{i1}} x_2^{b_{i2}} \cdots x_n^{b_{in}},\ x_1^{b_{j1}} x_2^{b_{j2}} \cdots x_n^{b_{jn}}]$$

which we can split up according to (C.3) and (C.4). We find that it has an exponent of $b_{ik}b_{j\ell} - b_{i\ell}b_{jk}$ on the $[x_j, x_k]$ term which in our notation is $b_{ijk\ell}$. This is by definition the $ijk\ell$-th entry of the matrix $R(\beta)$. Consider then

$$([x_i, x_j])\,\alpha\beta = ([x_1, x_2]^{a_{ij12}} [x_1, x_3]^{a_{ij13}} \cdots [x_{n-1}, x_n]^{a_{i,j,n-1,n}})\,\beta.$$

Ignoring all but the $k\ell$-th term we have

$$= [x_k, x_\ell]^{a_{ij12}b_{12k\ell}} [x_k, x_\ell]^{a_{ij13}b_{13k\ell}} \cdots [x_k, x_\ell]^{a_{i,j,n-1,n}b_{n-1,n,k,\ell}}$$

$$= [x_k, x_\ell]^{\sum_{c<d} a_{ijcd}b_{cdk\ell}}$$

which, taking into account the quadruple indices is the formula for matrix multiplication.

There is an natural map from the commutator $[x_i, x_j]$ to the basis element $e_i \wedge e_j$ of the exterior power of the standard representation $\bigwedge^2 \mathbb{C}^n$. It is clear that the action of $\mathrm{Aut}\,(F_n)$ on both of these objects respects this map, and from this it follows that $M_2(\alpha)$ is in fact the exterior power of the standard representation $\bigwedge^2 V$. ■

This type of construction holds for other subgroups in the descending central series for $\mathrm{Aut}\,(F_n)$. The result is to add another exterior power each time.

5 Results for specific n

We shall now consider not only the special case of $n = 2$, but also perform calculations for small n.

5.1 Results for $\mathrm{Aut}(F_2)$

Many of the theorems in section 3 do not hold for $n = 2$. This difficulty has led to some surprising consequences. For example it was known in 1992 that for $n \geq 3$, $\mathrm{Aut}(F_n)$ is not linear, *i.e.,* it has no injective matrix representation [6]. It wasn't until 2000, however, that it was shown that $\mathrm{Aut}(F_2)$ is linear, which was done by showing that the braid group B_4 is linear (see [5] and [9]).

The representation theory of $\mathrm{Aut}(F_2)$ is discussed in [16], where P, σ, and an involution $\tau = P\sigma PU$ defined by

$$\tau : x_1 \mapsto x_1 x_2$$
$$x_2 \mapsto x_2^{-1}$$

are used as generators. They calculate all irreducible representations of degree less than or equal to 5, up to what they call weak equivalence. Two representations ρ_1 and ρ_2 are weakly equivalence if ρ_1 is equivalent to either $\chi \otimes \rho_2$, for some 1-dimensional representation χ, or its dual.

The first thing which we note in our treatment of $\mathrm{Aut}(F_2)$, is that for Σ_2 as described in Theorem 3.4,

$$\mathrm{Aut}(F_2) = \langle \mathrm{Cox}(B_2), \Sigma_2 \rangle = \langle P, \sigma, a_2 \rangle .$$

Table 5.1: Irreducible Representations of Cox (B_2)

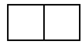
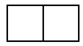
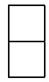
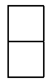
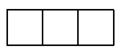
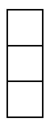
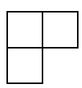
	λ	μ	P	σ
ρ_1	$\emptyset$		$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$
ρ_2		$\emptyset$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}-1\end{bmatrix}$
ρ_3	$\emptyset$		$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$
ρ_4		$\emptyset$	$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}-1\end{bmatrix}$
ρ_5			$\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$

Table 5.2: Irreducible Representations of S_3

λ	P	a_2
	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$
	$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}-1\end{bmatrix}$
	$\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$	$\frac{1}{2}\cdot\begin{bmatrix}1 & 1\\ 3 & -1\end{bmatrix}$

This follows from the fact that $U = a_2 P \sigma P$ which is easily verified:

$$x_1 \xrightarrow{a_2} x_1 x_2^{-1} \xrightarrow{P} x_2 x_1^{-1} \xrightarrow{\sigma} x_2 x_1 \xrightarrow{P} x_1 x_2$$

$$x_2 \xrightarrow{a_2} x_2^{-1} \xrightarrow{P} x_1^{-1} \xrightarrow{\sigma} x_1 \xrightarrow{P} x_2$$

This seems an enviable position, since it implies that we need only concern ourselves with finding representations of Cox (B_2), and S_3 which agree on P. This condition is easy to check using the Jordan Canonical Form of the matrices for P.

The irreducible representations of interest are given in tables 5.1 and 5.2. They have been conjugated so that the P matrices are in Jordan Canonical Form rather than the form in which they were originally calculated.

Ultimately, this information alone is insufficient to calculate all of the representations of $\mathrm{Aut}(F_2)$. This is because we are able to conjugate either representation separately by an invertible matrix, so long as P is not changed. This means that everywhere a appears we must substitute $S^{-1}aS$ for an arbitrary invertible matrix S which will be made of two blocks corresponding to the blocks of ± 1 in P.

Since we are only categorize representations up to conjugation, we are able to conjugate the entire representation by any matrix which does not change P or σ. Thus we are really only interested in invertible matrices S which do not change P, but do change σ. The author is unaware of any methods of easily determining all such matrices. If an algorithm for this were known, finding all representations of $\mathrm{Aut}(F_2)$ would then be easy.

We turn our attention to those equations which are of interest for $n = 2$ assuming that matrices are given for P, σ, and a_3. Any equations involving only elements of $\mathrm{Cox}(B_2)$, or Σ_2 will be automatically satisfied. Equations (U.1-U.3,U.7-U.8) do not apply, the relation $P = Q$ causes equation (U.4) to be trivial, and forces equation (U.6) equal to (U.5). Since $U = a_2 P\sigma P$ equation (U.9) simplifies to

$$1 = P\sigma P(aP\sigma P)P\sigma P(aP\sigma P)$$

and everything cancels out. With the same substitution equation (U.10) becomes

$$U = aP\sigma P = P(aP\sigma P)P\sigma(aP\sigma P)\sigma P\sigma$$

$$aP\sigma P = PaPaP\sigma P\sigma P\sigma$$

$$aP\sigma P = PaPa\sigma P \quad \text{Since } (P\sigma)^4 = 1$$

$$aP = PaPa$$

$$1 = (aP)^3$$

which will be automatically satisfied since it involves only elements of Σ_2 (and is true in Σ_2). Thus the only equation left is equation (U.5) which takes the form

$$\begin{aligned}
U &\rightleftarrows \sigma U \sigma \\
(aP\sigma P) &\rightleftarrows \sigma(aP\sigma P)\sigma \\
1 &= (aP\sigma P)(\sigma aP\sigma P\sigma)(aP\sigma P)^{-1}(\sigma aP\sigma P\sigma)^{-1} \\
&= (aP\sigma P)(\sigma aP\sigma P\sigma)(P\sigma Pa)(\sigma P\sigma Pa\sigma) \\
&= (aP\sigma P)\sigma a(P\sigma P\sigma P\sigma P)a(\sigma P\sigma Pa\sigma) \\
&= (aP\sigma P)(\sigma a\sigma a)(\sigma P\sigma Pa\sigma) \\
&= (P\sigma P)(\sigma a\sigma a)(P\sigma P)(\sigma a\sigma a) \\
(a\sigma)^2 &= (P\sigma P)(\sigma a)^2(P\sigma P)
\end{aligned}$$

This is therefore the only equation which needs to be checked to determine if a given pair of representations (and invertible matrix S) are compatible for determining a representation of $\mathrm{Aut}\,(F_2)$.

We include the irreducible representations of $\mathrm{Aut}\,(F_2)$ of dimension 3 or less to show how different the case of $n = 2$ is. For dimension 3 we include the representations only up to weak equivalence. For representations of higher dimension see [16].

We use x to denote a free variable *i.e.*, it can take on any value. Note that ρ_5 has non-constant trace. This is something which we have not found for $n \geq 3$.

Table 5.3: Irreducible Representations of $\mathrm{Aut}\,(F_2)$

$\mathrm{Cox}\,(B_2)$	U
ρ_1	$\begin{bmatrix}1\end{bmatrix}$
ρ_2	$\begin{bmatrix}1\end{bmatrix}$
ρ_3	$\begin{bmatrix}-1\end{bmatrix}$
ρ_4	$\begin{bmatrix}-1\end{bmatrix}$
ρ_5	$\frac{1}{4}\begin{bmatrix}x & -12\\ 2 & 3x^{-1}\end{bmatrix}$
$\rho_{1,3}$	$\frac{1}{2}\begin{bmatrix}-1 & 3\\ 1 & 1\end{bmatrix}$
$\rho_{1,4}$	$\frac{1}{2}\begin{bmatrix}-1 & 1\\ -3 & -1\end{bmatrix}$
$\rho_{2,3}$	$\frac{1}{2}\begin{bmatrix}1 & 1\\ -3 & 1\end{bmatrix}$
$\rho_{2,4}$	$\frac{1}{2}\begin{bmatrix}1 & 1\\ 3 & -1\end{bmatrix}$
$\rho_{1,1,3}$	$\frac{1}{2}\begin{bmatrix}-x-1 & x & 2\\ -2x-3 & x+2 & 2\\ x+3 & -x & 1\end{bmatrix}$
$\rho_{1,1,3}$	$\frac{1}{2}\begin{bmatrix}2 & 0 & 0\\ -3x & -1 & 3\\ 2x & 1 & 1\end{bmatrix}$
$\rho_{1,5}$	$\frac{1}{2}\begin{bmatrix}2 & 0 & 0\\ x & 3 & -1\\ x & 1 & 1\end{bmatrix}$
$\rho_{1,5}$	$\frac{1}{2}\begin{bmatrix}2 & x & x\\ 0 & 1 & -1\\ 0 & 1 & 3\end{bmatrix}$
$\rho_{1,5}$	$\frac{1}{2}\begin{bmatrix}0 & 2 & -2\\ 1 & -1 & -1\\ -1 & -1 & -1\end{bmatrix}$

Continued on next page...

Continued ...

Cox (B_2)	U
$\rho_{1,5}$	$\frac{1}{2}\begin{bmatrix} 0 & 2 & 2 \\ 1 & -1 & 1 \\ -1 & -1 & 1 \end{bmatrix}$

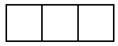
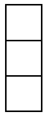
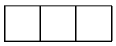
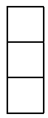
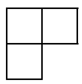
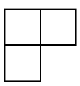
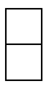
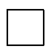
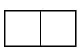
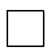
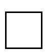
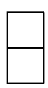
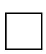
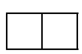
5.2 Results for Aut (F_3)

We now include the irreducible representations of Cox (B_3) in table 5.5. Notice that the 1 and 2-dimensional representations of Cox (B_3) all have σ central. Thus by Theorem 3.11 any representation which does not have a 3-dimensional component when restricted to Cox (B_3) must have finite image.

Table 5.4: Character Table for Cox (B_3)

$\lvert C_i \rvert$	1	3	6	6	3	6	8	8	6	1
χ_1	1	1	1	1	1	1	1	1	1	1
χ_2	1	−1	−1	1	1	1	1	−1	−1	−1
χ_3	1	−1	1	−1	1	−1	1	−1	1	−1
χ_4	1	1	−1	−1	1	−1	1	1	−1	1
χ_5	2	2	0	0	2	0	−1	−1	0	2
χ_6	2	−2	0	0	2	0	−1	1	0	−2
χ_7	3	−1	−1	1	−1	−1	0	0	1	3
χ_8	3	−1	1	−1	−1	1	0	0	−1	3
χ_9	3	1	−1	−1	−1	1	0	0	1	−3
χ_{10}	3	1	1	1	−1	−1	0	0	−1	−3

Table 5.5: Irreducible Representations of $\mathrm{Cox}(B_3)$

	λ	μ	P	Q	σ
χ_1	$\emptyset$		$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$
χ_2		$\emptyset$	$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}-1\end{bmatrix}$
χ_3		$\emptyset$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}-1\end{bmatrix}$
χ_4	$\emptyset$		$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$
χ_5	$\emptyset$		$\begin{bmatrix}1 & 0\\ -1 & -1\end{bmatrix}$	$\begin{bmatrix}-1 & -1\\ 1 & 0\end{bmatrix}$	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
χ_6		$\emptyset$	$\begin{bmatrix}1 & 0\\ -1 & -1\end{bmatrix}$	$\begin{bmatrix}-1 & -1\\ 1 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}$
χ_7			$\begin{bmatrix}-1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & -1\\ -1 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}$
χ_8			$\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & -1\end{bmatrix}$
χ_9			$\begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & -1\end{bmatrix}$	$\begin{bmatrix}0 & -1 & 0\\ 0 & 0 & -1\\ 1 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$
χ_{10}			$\begin{bmatrix}0 & 1 & 0\\ 1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}$	$\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 1 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$

We calculated all representations of dimension less than or equal to 4, as well as those which have no more than 2 irreducible components when restricted to $\mathrm{Cox}(B_3)$. The results are summarized in table 5.6 where $r = \dfrac{7 \pm \sqrt{-7}}{2}$ and x is a free variable.

Table 5.6: Irreducible Representations of $\mathrm{Aut}(F_3)$

$\mathrm{Cox}(B_3)$	U
ρ_1	$\begin{bmatrix}1\end{bmatrix}$
ρ_2	$\begin{bmatrix}1\end{bmatrix}$
ρ_7	$\begin{bmatrix}1&0&0\\0&1&0\\0&1&1\end{bmatrix}$ $\begin{bmatrix}1&0&0\\0&1&-1\\0&0&1\end{bmatrix}$
ρ_{10}	$\begin{bmatrix}1&1&0\\0&1&0\\0&0&1\end{bmatrix}$ $\begin{bmatrix}1&0&0\\-1&1&0\\0&0&1\end{bmatrix}$
$\rho_{3,6}$	$\frac{1}{3}\begin{bmatrix}-2&2r-6&r\\-\frac{r+3}{2}&-r+2&-2r+7\\1&2r-6&r-3\end{bmatrix}$
$\rho_{4,5}$	$\frac{1}{3}\begin{bmatrix}-2&r-2&\frac{r}{2}\\\frac{-3r+1}{2}&-r+2&-2r+3\\2&2r+4&r-3\end{bmatrix}$
$\rho_{2,7}$	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\x&0&1&1\\0&0&0&1\end{bmatrix}$ $\begin{bmatrix}1&0&x&0\\0&1&0&0\\0&0&1&0\\0&0&-1&1\end{bmatrix}$
$\rho_{1,10}$	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\x&-1&1&0\\0&0&0&1\end{bmatrix}$ $\begin{bmatrix}1&0&x&0\\0&1&1&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$

Recall that the maximum cardinality of the image if σ is central is 336. The representation $\rho_{3,6}$ has image of order 336 isomorphic to

$$\rho_{3,6} \cong \langle (1\ 7\ 3\ 5\ 4\ 2\ 6)\,, (1\ 6\ 7)(2\ 4\ 3)(8\ 9)\rangle$$

according to the GAP library of small groups. The representation $\rho_{4,5}$ has order

$168 = 336/2$ and is isomorphic to

$$\rho_{4,5} \cong \langle (3\ 4)(5\ 6)\,, (1\ 2\ 3)(4\ 5\ 7) \rangle\,.$$

We now include some conjectures applicable for all $n \geq 3$ since we have the most information about $n = 3$.

Conjecture 5.1. *The only irreducible representations found to extend from* $\mathrm{Cox}\,(B_n)$ *to* $\mathrm{Aut}\,(F_n)$ *are the exterior powers of the standard representation* V *for* $n = 3, 4, 5$. *It is conjectured that these are the only irreducible representations of* $\mathrm{Cox}\,(B_n)$ *which extend to* $\mathrm{Aut}\,(F_n)$ *for all* $n \geq 3$.

Conjecture 5.2. *When there are possibilities as to how to rearrange the restricted representations so that the different representations of* Σ_n *match up with different representations of* $\mathrm{Cox}\,(B_n)$, *then there will be more then there will be free variables in the extended representation.*

For example consider the case of $\rho_1 \oplus \rho_{10}$. *The restriction to* S_n *gives the following decomposition*

$$\ydiagram{2,1} \oplus \ydiagram{3} \oplus \ydiagram{3} = \left(\ydiagram{1}, \ydiagram{2}\right) \oplus \left(\emptyset, \ydiagram{3}\right)\Big|_{S_n}$$
$$= \ydiagram{3,1} \oplus \ydiagram{4}\Big|_{S_n}.$$

Here the restriction of $\ydiagram{4}$ *could correspond to the restriction of* $\left(\emptyset, \ydiagram{3}\right)$, *or it could correspond to part of the restriction of* $\left(\ydiagram{1}, \ydiagram{2}\right)$. *This accounts for the possibility of having a non-zero free variable or not*

Conjecture 5.3. *If a representation* $\rho = \bigoplus_i V_{\lambda_i,\mu_i}$ *of* $\mathrm{Cox}\,(B_n)$ *extends to* $\mathrm{Aut}\,(F_n)$, *then the representations of* $\mathrm{Cox}\,(B_n)$ *obtained from either conjugating each partition but not swapping positions, or from swapping positions but not conjugating, will not extend to partitions of* $\mathrm{Aut}\,(F_n)$ *unless swapping and not conjugating is the same as swapping and conjugating.*

5.3 Results for $\text{Aut}(F_4)$

We calculated the representations of $\text{Aut}(F_4)$ of dimension less than or equal to 5, as well as those which are irreducible on $\text{Cox}(B_4)$. Those representations which are irreducible on $\text{Aut}(F_4)$ are included in table 5.7.

Table 5.7: Irreducible Representations of $\text{Aut}(F_4)$

$\text{Cox}(B_4)$	U
ρ_1	$\begin{bmatrix}1\end{bmatrix}$
ρ_2	$\begin{bmatrix}1\end{bmatrix}$
ρ_{11}	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&1\\0&0&0&1\end{bmatrix}$ $\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&-1&1\end{bmatrix}$
ρ_{14}	$\begin{bmatrix}1&1&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$ $\begin{bmatrix}1&0&0&0\\-1&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$
$\rho_{1,14}$	$\begin{bmatrix}1&0&x&0&0\\0&1&1&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$
$\rho_{2,11}$	$\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\x&0&0&1&1\\0&0&0&0&1\end{bmatrix}$
ρ_{18}	$\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&1\\0&0&0&1&0&0\\0&0&0&1&1&0\\0&0&0&0&0&1\end{bmatrix}$ $\begin{bmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&1&-1&0\\0&0&0&0&1&0\\0&0&-1&0&0&1\end{bmatrix}$

Table 5.8: Character Table for Cox (B_4)

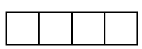
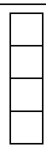
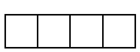
$\lvert C_i \rvert$	1	4	12	12	6	24	32	32	24	4	12	24	12	32	48	48	32	12	12	1
χ_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
χ_2	1	−1	−1	1	1	1	1	−1	−1	−1	1	−1	−1	−1	−1	1	1	1	1	1
χ_3	1	−1	1	−1	1	−1	1	−1	1	−1	1	−1	1	−1	1	−1	1	1	−1	1
χ_4	1	1	−1	−1	1	−1	1	1	−1	1	1	1	−1	1	−1	−1	1	1	−1	1
χ_5	2	−2	0	0	2	0	−1	1	0	−2	2	−2	0	1	0	0	−1	2	0	2
χ_6	2	2	0	0	2	0	−1	−1	0	2	2	2	0	−1	0	0	−1	2	0	2
χ_7	3	−3	−1	1	3	1	0	0	−1	−3	−1	1	−1	0	1	−1	0	−1	1	3
χ_8	3	−3	1	−1	3	−1	0	0	1	−3	−1	1	1	0	−1	1	0	−1	−1	3
χ_9	3	3	−1	−1	3	−1	0	0	−1	3	−1	−1	−1	0	1	1	0	−1	−1	3
χ_{10}	3	3	1	1	3	1	0	0	1	3	−1	−1	1	0	−1	−1	0	−1	1	3
χ_{11}	4	−2	−2	2	0	0	1	−1	0	2	0	0	2	1	0	0	−1	0	−2	−4
χ_{12}	4	−2	2	−2	0	0	1	−1	0	2	0	0	−2	1	0	0	−1	0	2	−4
χ_{13}	4	2	−2	−2	0	0	1	1	0	−2	0	0	2	−1	0	0	−1	0	2	−4
χ_{14}	4	2	2	2	0	0	1	1	0	−2	0	0	−2	−1	0	0	−1	0	−2	−4

Continued on next page...

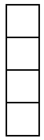
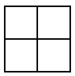
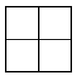
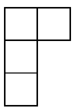
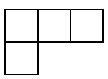
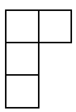
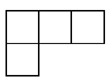
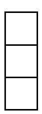
Continued ...

$\lvert C_i \rvert$	1	4	12	12	6	24	32	32	24	4	12	24	12	32	48	48	32	12	12	1
χ_{15}	6	0	2	0	−2	0	0	0	−2	0	2	0	2	0	0	0	0	−2	0	6
χ_{16}	6	0	−2	0	−2	0	0	0	2	0	2	0	−2	0	0	0	0	−2	0	6
χ_{17}	6	0	0	−2	−2	2	0	0	0	0	−2	0	0	0	0	0	0	2	−2	6
χ_{18}	6	0	0	2	−2	−2	0	0	0	0	−2	0	0	0	0	0	0	2	2	6
χ_{19}	8	−4	0	0	0	0	−1	1	0	4	0	0	0	−1	0	0	1	0	0	−8
χ_{20}	8	4	0	0	0	0	−1	−1	0	−4	0	0	0	1	0	0	1	0	0	−8

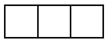
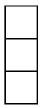
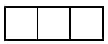
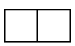
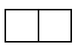
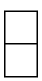
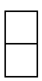
Table 5.9: Irreducible Representations of Cox (B_4)

	λ	μ	P	Q	σ
χ_1	$\emptyset$		$[1]$	$[1]$	$[1]$
χ_2		$\emptyset$	$[-1]$	$[-1]$	$[-1]$
χ_3		$\emptyset$	$[1]$	$[1]$	$[-1]$

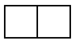
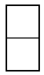
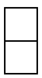
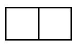
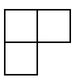
Continued on next page...

	λ	μ	P	Q	σ
χ_4	$\emptyset$		$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$
χ_5		$\emptyset$	$\begin{bmatrix}1 & 0\\ -1 & -1\end{bmatrix}$	$\begin{bmatrix}-1 & -1\\ 0 & 1\end{bmatrix}$	$\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}$
χ_6	$\emptyset$		$\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0\\ -1 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$
χ_7		$\emptyset$	$\begin{bmatrix}1 & 0 & 0\\ -1 & -1 & 0\\ 1 & 0 & -1\end{bmatrix}$	$\begin{bmatrix}1 & 1 & 0\\ -1 & 0 & 1\\ 1 & 0 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & -1\end{bmatrix}$
χ_8		$\emptyset$	$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -1 & -1 & -1\end{bmatrix}$	$\begin{bmatrix}-1 & -1 & -1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 0\\ 0 & -1 & 0\\ 0 & 0 & -1\end{bmatrix}$
χ_9	$\emptyset$		$\begin{bmatrix}1 & -1 & 1\\ 0 & -1 & 0\\ 0 & 0 & -1\end{bmatrix}$	$\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 1 & -1 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$
χ_{10}	$\emptyset$		$\begin{bmatrix}1 & 0 & -1\\ 0 & 1 & -1\\ 0 & 0 & -1\end{bmatrix}$	$\begin{bmatrix}0 & 0 & -1\\ 1 & 0 & -1\\ 0 & 1 & -1\end{bmatrix}$	$\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}$
χ_{11}			$\begin{bmatrix}-1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\end{bmatrix}$	$\begin{bmatrix}-1 & 0 & 0 & 0\\ 0 & -1 & 0 & 0\\ 0 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\end{bmatrix}$

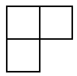
Continued on next page...

	λ	μ	P	Q	σ
χ_{12}			$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}$	$\begin{bmatrix}-1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&1\end{bmatrix}$
χ_{13}			$\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&-1&0\\0&0&0&-1\end{bmatrix}$	$\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}$	$\begin{bmatrix}-1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$
χ_{14}			$\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$	$\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}$	$\begin{bmatrix}-1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$
χ_{15}			$\begin{bmatrix}1&0&0&0&0&0\\0&0&1&0&0&0\\0&1&0&0&0&0\\0&0&0&0&1&0\\0&0&0&1&0&0\\0&0&0&0&0&1\end{bmatrix}$	$\begin{bmatrix}0&0&1&0&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\\1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&0&1&0&0\end{bmatrix}$	$\begin{bmatrix}-1&0&0&0&0&0\\0&-1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&-1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}$
χ_{16}			$\begin{bmatrix}-1&0&0&0&0&0\\0&0&1&0&0&0\\0&1&0&0&0&0\\0&0&0&0&1&0\\0&0&0&1&0&0\\0&0&0&0&0&-1\end{bmatrix}$	$\begin{bmatrix}0&0&-1&0&0&0\\0&0&0&0&-1&0\\0&0&0&0&0&-1\\-1&0&0&0&0&0\\0&-1&0&0&0&0\\0&0&0&-1&0&0\end{bmatrix}$	$\begin{bmatrix}-1&0&0&0&0&0\\0&-1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&-1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{bmatrix}$

Continued on next page...

	λ	μ	P	Q	σ
χ_{17}			$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
χ_{18}			$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
χ_{19}			$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Continued on next page...

	λ	μ	P	Q	σ
χ_{20}			$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

5.4 Results for $\mathrm{Aut}(F_5)$

For $n = 5$ we tested only those representations which are irreducible on $\mathrm{Cox}(B_5)$. Those which are able to be extended to a representation of $\mathrm{Aut}(F_5)$ are included as table 5.10. Note that they are the exterior powers of the standard representation as conjectured.

Table 5.10: Irreducible Representations of $\mathrm{Aut}(F_5)$

$\mathrm{Cox}(B_5)$	U
ρ_1	$\begin{bmatrix}1\end{bmatrix}$
ρ_2	$\begin{bmatrix}1\end{bmatrix}$
ρ_9	$\begin{bmatrix}1&0&0&0&0\\0&1&-1&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$ $\begin{bmatrix}1&0&0&0&0\\0&1&0&0&0\\0&1&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$
ρ_{12}	$\begin{bmatrix}1&0&0&0&0\\-1&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$ $\begin{bmatrix}1&1&0&0&0\\0&1&0&0&0\\0&0&1&0&0\\0&0&0&1&0\\0&0&0&0&1\end{bmatrix}$
ρ_{23}	$\begin{bmatrix}1&0&0&0&0&0&0&0&0&0\\0&1&-1&0&0&0&0&0&0&0\\0&0&1&0&0&0&0&0&0&0\\0&0&0&1&0&0&0&0&0&0\\0&0&0&0&1&0&0&0&0&0\\0&0&0&0&0&1&0&0&0&0\\0&0&0&0&-1&0&1&0&0&0\\0&0&0&0&0&0&0&1&0&0\\0&0&0&0&0&0&0&0&1&0\\0&0&0&0&0&0&0&0&-1&1\end{bmatrix}$

Continued on next page...

Continued ...

Cox (B_5)	U
ρ_{23}	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
ρ_{26}	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
ρ_{26}	$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Table 5.11: Character Table for Cox (B_5)

$\|C_i\|$	1	5	10	10	5	1	20	20	60	60	60	60	20	20	80	80	160	160	80	80	60	120	60	60	120	60	240	240	240	240	160	160	160	160	384	384
χ_1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
χ_2	1	−1	1	−1	1	−1	−1	1	1	−1	−1	1	1	−1	1	−1	−1	1	1	−1	1	−1	1	−1	1	−1	−1	1	1	−1	−1	1	1	−1	1	−1
χ_3	1	−1	1	−1	1	−1	1	−1	−1	1	1	−1	−1	1	1	−1	−1	1	1	−1	1	−1	1	−1	1	−1	1	−1	−1	1	1	−1	−1	1	1	−1
χ_4	1	1	1	1	1	1	−1	−1	−1	−1	−1	−1	−1	−1	1	1	1	1	1	1	1	1	1	1	1	1	−1	−1	−1	−1	−1	−1	−1	−1	1	1
χ_5	4	−4	4	−4	4	−4	−2	2	2	−2	−2	2	2	−2	1	−1	−1	1	1	−1	0	0	0	0	0	0	0	0	0	0	1	−1	−1	1	−1	1
χ_6	4	4	4	4	4	4	−2	−2	−2	−2	−2	−2	−2	−2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	−1	−1
χ_7	4	−4	4	−4	4	−4	2	−2	−2	2	2	−2	−2	2	1	−1	−1	1	1	−1	0	0	0	0	0	0	0	0	0	0	−1	1	1	−1	−1	1
χ_8	4	4	4	4	4	4	2	2	2	2	2	2	2	2	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	−1	−1	−1	−1	−1	−1
χ_9	5	−3	1	1	−3	5	−3	3	1	−1	1	−1	−3	3	2	−2	0	0	−2	2	1	−1	1	1	−1	1	−1	1	−1	1	0	0	0	0	0	0
χ_{10}	5	3	1	−1	−3	−5	−3	−3	−1	−1	1	1	3	3	2	2	0	0	−2	−2	1	1	1	−1	−1	−1	−1	−1	1	1	0	0	0	0	0	0
χ_{11}	5	−3	1	1	−3	5	3	−3	−1	1	−1	1	3	−3	2	−2	0	0	−2	2	1	−1	1	1	−1	1	1	−1	1	−1	0	0	0	0	0	0
χ_{12}	5	3	1	−1	−3	−5	3	3	1	1	−1	−1	−3	−3	2	2	0	0	−2	−2	1	1	1	−1	−1	−1	1	1	−1	−1	0	0	0	0	0	0
χ_{13}	5	−5	5	−5	5	−5	1	−1	−1	1	1	−1	−1	1	−1	1	1	−1	−1	1	1	−1	1	−1	1	−1	−1	1	1	−1	1	−1	−1	1	0	0
χ_{14}	5	5	5	5	5	5	1	1	1	1	1	1	1	1	−1	−1	−1	−1	−1	−1	1	1	1	1	1	1	−1	−1	−1	−1	1	1	1	1	0	0
χ_{15}	5	−5	5	−5	5	−5	−1	1	1	−1	−1	1	1	−1	−1	1	1	−1	−1	1	1	−1	1	−1	1	−1	1	−1	−1	1	−1	1	1	−1	0	0
χ_{16}	5	5	5	5	5	5	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	−1	1	1	1	1	1	1	1	1	1	1	−1	−1	−1	−1	0	0
χ_{17}	6	−6	6	−6	6	−6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−2	2	−2	2	−2	2	0	0	0	0	0	0	0	0	1	−1
χ_{18}	6	6	6	6	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−2	−2	−2	−2	−2	−2	0	0	0	0	0	0	0	0	1	1
χ_{19}	10	2	−2	−2	2	10	4	2	0	−2	0	−2	4	2	1	1	−1	−1	1	1	2	0	−2	2	0	−2	0	0	0	0	1	1	−1	−1	0	0
χ_{20}	10	−2	−2	2	2	−10	4	−2	0	−2	0	2	−4	2	1	−1	1	−1	1	−1	2	0	−2	−2	0	2	0	0	0	0	1	−1	1	−1	0	0
χ_{21}	10	2	−2	−2	2	10	−4	−2	0	2	0	2	−4	−2	1	1	−1	−1	1	1	2	0	−2	2	0	−2	0	0	0	0	−1	−1	1	1	0	0
χ_{22}	10	−2	−2	2	2	−10	−4	2	0	2	0	−2	4	−2	1	−1	1	−1	1	−1	2	0	−2	−2	0	2	0	0	0	0	−1	1	−1	1	0	0
χ_{23}	10	2	−2	−2	2	10	2	4	−2	0	−2	0	2	4	1	1	−1	−1	1	1	−2	0	2	−2	0	2	0	0	0	0	−1	−1	1	1	0	0
χ_{24}	10	−2	−2	2	2	−10	2	−4	2	0	−2	0	−2	4	1	−1	1	−1	1	−1	−2	0	2	2	0	−2	0	0	0	0	−1	1	−1	1	0	0
χ_{25}	10	2	−2	−2	2	10	−2	−4	2	0	2	0	−2	−4	1	1	−1	−1	1	1	−2	0	2	−2	0	2	0	0	0	0	1	1	−1	−1	0	0
χ_{26}	10	−2	−2	2	2	−10	−2	4	−2	0	2	0	2	−4	1	−1	1	−1	1	−1	−2	0	2	2	0	−2	0	0	0	0	1	−1	1	−1	0	0
χ_{27}	10	−6	2	2	−6	10	0	0	0	0	0	0	0	0	−2	2	0	0	2	−2	2	−2	2	2	−2	2	0	0	0	0	0	0	0	0	0	0
χ_{28}	10	6	2	−2	−6	−10	0	0	0	0	0	0	0	0	−2	−2	0	0	2	2	2	2	2	−2	−2	−2	0	0	0	0	0	0	0	0	0	0
χ_{29}	15	−9	3	3	−9	15	3	−3	−1	1	−1	1	3	−3	0	0	0	0	0	0	−1	1	−1	−1	1	−1	−1	1	−1	1	0	0	0	0	0	0
χ_{30}	15	9	3	−3	−9	−15	3	3	1	1	−1	−1	−3	−3	0	0	0	0	0	0	−1	−1	−1	1	1	1	−1	−1	1	1	0	0	0	0	0	0

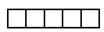
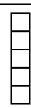
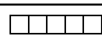
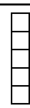
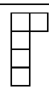
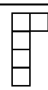
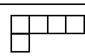
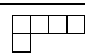
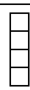
Continued on next page...

Continued . . .

$\lvert C_i \rvert$	1	5	10	10	5	1	20	20	60	60	60	60	20	20	80	80	160	160
χ_{31}	15	−9	3	3	−9	15	−3	3	1	−1	1	−1	−3	3	0	0	0	0
χ_{32}	15	9	3	−3	−9	−15	−3	−3	−1	−1	1	1	3	3	0	0	0	0
χ_{33}	20	4	−4	−4	4	20	−2	2	−2	2	−2	2	−2	2	−1	−1	1	1
χ_{34}	20	−4	−4	4	4	−20	−2	−2	2	2	−2	−2	2	2	−1	1	−1	1
χ_{35}	20	4	−4	−4	4	20	2	−2	2	−2	2	−2	2	−2	−1	−1	1	1
χ_{36}	20	−4	−4	4	4	−20	2	2	−2	−2	2	2	−2	−2	−1	1	−1	1

$\lvert C_i \rvert$	80	80	60	120	60	60	120	60	240	240	240	240	160	160	160	160	384	384
χ_{31}	0	0	−1	1	−1	−1	1	−1	1	−1	1	−1	0	0	0	0	0	0
χ_{32}	0	0	−1	−1	−1	1	1	1	1	1	−1	−1	0	0	0	0	0	0
χ_{33}	−1	−1	0	0	0	0	0	0	0	0	0	0	1	1	−1	−1	0	0
χ_{34}	−1	1	0	0	0	0	0	0	0	0	0	0	1	−1	1	−1	0	0
χ_{35}	−1	−1	0	0	0	0	0	0	0	0	0	0	−1	−1	1	1	0	0
χ_{36}	−1	1	0	0	0	0	0	0	0	0	0	0	−1	1	−1	1	0	0

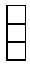
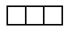
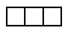
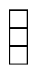
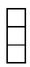
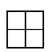
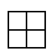
Table 5.12: Irreducible Representations of Cox (B_5)

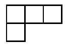
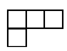
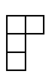
	λ	μ	P	Q	σ
χ_1	$\emptyset$		$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$
χ_2		$\emptyset$	$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}-1\end{bmatrix}$
χ_3		$\emptyset$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}-1\end{bmatrix}$
χ_4	$\emptyset$		$\begin{bmatrix}-1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$	$\begin{bmatrix}1\end{bmatrix}$
χ_5		$\emptyset$	$\begin{bmatrix}1&0&0&0\\-1&-1&0&0\\1&0&-1&0\\-1&0&0&-1\end{bmatrix}$	$\begin{bmatrix}-1&-1&0&0\\1&0&-1&0\\-1&0&0&-1\\1&0&0&0\end{bmatrix}$	$\begin{bmatrix}-1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{bmatrix}$
χ_6	$\emptyset$		$\begin{bmatrix}1&-1&1&-1\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{bmatrix}$	$\begin{bmatrix}0&-1&0&0\\0&0&-1&0\\0&0&0&-1\\1&-1&1&-1\end{bmatrix}$	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$
χ_7		$\emptyset$	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\-1&-1&-1&-1\end{bmatrix}$	$\begin{bmatrix}-1&-1&-1&-1\\1&0&0&0\\0&1&0&0\\0&0&1&0\end{bmatrix}$	$\begin{bmatrix}-1&0&0&0\\0&-1&0&0\\0&0&-1&0\\0&0&0&-1\end{bmatrix}$
χ_8	$\emptyset$		$\begin{bmatrix}1&0&0&-1\\0&1&0&-1\\0&0&1&-1\\0&0&0&-1\end{bmatrix}$	$\begin{bmatrix}0&0&0&-1\\1&0&0&-1\\0&1&0&-1\\0&0&1&-1\end{bmatrix}$	$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$
χ_9			$\begin{bmatrix}-1&0&0&0&0\\0&-1&0&0&0\\0&0&-1&0&0\\0&0&0&0&1\\0&0&0&1&0\end{bmatrix}$	$\begin{bmatrix}0&0&0&0&1\\-1&0&0&0&0\\0&-1&0&0&0\\0&0&-1&0&0\\0&0&0&-1&0\end{bmatrix}$	$\begin{bmatrix}-1&0&0&0&0\\0&-1&0&0&0\\0&0&-1&0&0\\0&0&0&-1&0\\0&0&0&0&1\end{bmatrix}$

Continued on next page. . .

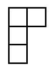
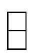
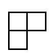
	λ	μ	P	Q	σ
χ_{10}	(1)	$(1,1,1,1)$	$\begin{bmatrix} 0&1&0&0&0\\ 1&0&0&0&0\\ 0&0&-1&0&0\\ 0&0&0&-1&0\\ 0&0&0&0&-1 \end{bmatrix}$	$\begin{bmatrix} 0&-1&0&0&0\\ 0&0&-1&0&0\\ 0&0&0&-1&0\\ 0&0&0&0&-1\\ 1&0&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1 \end{bmatrix}$
χ_{11}	(4)	(1)	$\begin{bmatrix} 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&0&1\\ 0&0&0&1&0 \end{bmatrix}$	$\begin{bmatrix} 0&0&0&0&1\\ 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0\\ 0&-1&0&0&0\\ 0&0&-1&0&0\\ 0&0&0&-1&0\\ 0&0&0&0&1 \end{bmatrix}$
χ_{12}	(1)	(4)	$\begin{bmatrix} 0&1&0&0&0\\ 1&0&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1\\ 1&0&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1 \end{bmatrix}$
χ_{13}	$(3,2)$	$\emptyset$	$\begin{bmatrix} 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ -1&-1&0&-1&0\\ -1&0&-1&0&-1 \end{bmatrix}$	$\begin{bmatrix} -1&-1&0&-1&0\\ -1&0&-1&0&-1\\ 1&0&0&0&0\\ 1&0&0&1&1\\ 0&1&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0\\ 0&-1&0&0&0\\ 0&0&-1&0&0\\ 0&0&0&-1&0\\ 0&0&0&0&-1 \end{bmatrix}$
χ_{14}	$\emptyset$	$(3,2)$	$\begin{bmatrix} 0&1&0&0&0\\ 1&0&0&0&0\\ 0&0&0&0&1\\ 1&-1&-1&1&1\\ 0&0&1&0&0 \end{bmatrix}$	$\begin{bmatrix} 0&0&-1&0&0\\ 0&0&-1&0&1\\ 0&0&0&-1&0\\ 0&1&0&-1&0\\ 1&-1&-1&0&1 \end{bmatrix}$	$\begin{bmatrix} 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1 \end{bmatrix}$
χ_{15}	$(2,2,1)$	$\emptyset$	$\begin{bmatrix} 1&0&0&0&0\\ -1&-1&0&0&0\\ 0&0&1&0&0\\ 0&0&-1&-1&0\\ 1&0&1&0&-1 \end{bmatrix}$	$\begin{bmatrix} 1&1&1&1&0\\ -1&-1&-1&0&1\\ -1&-1&0&0&0\\ 0&1&0&0&0\\ 0&0&0&1&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0\\ 0&-1&0&0&0\\ 0&0&-1&0&0\\ 0&0&0&-1&0\\ 0&0&0&0&-1 \end{bmatrix}$
χ_{16}	$\emptyset$	$(2,2,1)$	$\begin{bmatrix} 0&1&1&0&-1\\ 1&0&-1&-1&0\\ 0&0&1&1&-1\\ 0&0&0&-1&0\\ 0&0&0&0&-1 \end{bmatrix}$	$\begin{bmatrix} 0&0&0&0&1\\ 0&0&-1&-1&1\\ 0&0&0&1&0\\ 1&-1&-1&-1&1\\ 0&-1&-1&0&1 \end{bmatrix}$	$\begin{bmatrix} 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&1&0&0\\ 0&0&0&1&0\\ 0&0&0&0&1 \end{bmatrix}$
χ_{17}	$(3,1,1)$	$\emptyset$	$\begin{bmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ -1&-1&0&-1&0&0\\ 1&0&-1&0&-1&0\\ 0&1&1&0&0&-1 \end{bmatrix}$	$\begin{bmatrix} 1&1&0&1&0&0\\ -1&0&1&0&1&0\\ 1&0&0&0&0&0\\ 0&-1&-1&0&0&1\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0\\ 0&-1&0&0&0&0\\ 0&0&-1&0&0&0\\ 0&0&0&-1&0&0\\ 0&0&0&0&-1&0\\ 0&0&0&0&0&-1 \end{bmatrix}$
χ_{18}	$\emptyset$	$(3,1,1)$	$\begin{bmatrix} 1&0&-1&0&1&0\\ 0&1&-1&0&0&1\\ 0&0&-1&0&0&0\\ 0&0&0&1&-1&1\\ 0&0&0&0&-1&0\\ 0&0&0&0&0&-1 \end{bmatrix}$	$\begin{bmatrix} 0&0&1&0&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1\\ 1&0&-1&0&1&0\\ 0&1&-1&0&0&1\\ 0&0&0&1&-1&1 \end{bmatrix}$	$\begin{bmatrix} 1&0&0&0&0&0\\ 0&1&0&0&0&0\\ 0&0&1&0&0&0\\ 0&0&0&1&0&0\\ 0&0&0&0&1&0\\ 0&0&0&0&0&1 \end{bmatrix}$
χ_{19}	(2)	(3)	$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{20}	(3)	(2)	$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&-1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{21}	$(1,1)$	$(1,1,1)$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&-1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&-1&0\\ 0&0&0&0&0&0&0&0&0&-1 \end{bmatrix}$	$\begin{bmatrix} 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ -1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$

Continued on next page...

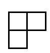
	λ	μ	P	Q	σ
χ_{22}			$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&-1 \end{bmatrix}$	$\begin{bmatrix} 0&0&0&0&-1&0&0&0&0&0\\ 0&0&0&0&0&0&-1&0&0&0\\ 0&0&0&0&0&0&0&0&-1&0\\ 1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&-1\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&-1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{23}			$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ -1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{24}			$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&-1 \end{bmatrix}$	$\begin{bmatrix} 0&0&0&0&-1&0&0&0&0&0\\ 0&0&0&0&0&0&-1&0&0&0\\ 0&0&0&0&0&0&0&0&-1&0\\ 1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&-1\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&-1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{25}			$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&-1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&-1&0\\ 0&0&0&0&0&0&0&0&0&-1 \end{bmatrix}$	$\begin{bmatrix} 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{26}			$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$	$\begin{bmatrix} 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&-1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{27}			$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0\\ -1&-1&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&-1&-1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&-1&-1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$	$\begin{bmatrix} 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1\\ -1&-1&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&-1&-1&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&-1&-1&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&-1&-1&0&0\\ 0&0&0&0&0&0&0&1&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0\\ 0&0&-1&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&-1&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&-1&0&0&0\\ 0&0&0&0&0&0&0&-1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{28}			$\begin{bmatrix} 0&1&0&0&0&0&0&0&0&0\\ 1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&1&0 \end{bmatrix}$	$\begin{bmatrix} 0&-1&0&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0\\ 0&-1&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&-1&0&0&0\\ 0&0&0&-1&0&1&0&0&0&0\\ 0&0&0&0&0&0&-1&1&0&0\\ 0&0&0&0&0&0&0&0&-1&0\\ 0&0&0&0&0&0&0&0&-1&1\\ 1&0&0&0&0&0&0&0&0&0\\ 0&0&1&0&0&0&0&0&0&0 \end{bmatrix}$	$\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0\\ 0&1&0&0&0&0&0&0&0&0\\ 0&0&-1&0&0&0&0&0&0&0\\ 0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$

	λ	μ	$PQ\sigma$
χ_{29}			$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ -1&-1&0&-1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&0&-1&-1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-1&-1&-1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \end{bmatrix}$ $\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ -1&-1&0&-1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&0&-1&-1&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-1&-1&-1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&-1&-1&-1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \end{bmatrix}$ $\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{30}			$\begin{bmatrix} 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&-1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&-1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&-1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&-1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1 \end{bmatrix}$ $\begin{bmatrix} 0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0 \\ 0&1&0&0&0&0&-1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&-1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0 \\ 0&0&0&0&1&0&0&0&0&-1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&-1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&-1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1 \\ 0&0&0&0&0&0&0&0&0&0&1&-1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&-1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix}$ $\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{31}			$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ -1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&-1&-1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&-1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-1&-1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&-1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \end{bmatrix}$ $\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\ 1&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ -1&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&-1&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-1&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&-1&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \end{bmatrix}$

Continued on next page...

	λ	μ	$PQ\sigma$
			$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
χ_{32}			$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$ $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
χ_{33}			$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$ $\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ $\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

Continued on next page...

	λ	μ	$PQ\sigma$
χ_{34}			$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ -1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&-1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1 \end{bmatrix}$ $\begin{bmatrix} 0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0 \\ -1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0 \\ 0&0&-1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&-1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&-1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&-1&0&-1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&-1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \end{bmatrix}$ $\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$
χ_{35}			$\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1 \end{bmatrix}$ $\begin{bmatrix} 0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&-1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&-1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&-1&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&-1 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \end{bmatrix}$ $\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$

Continued on next page...

	λ	μ	$PQ\sigma$
χ_{36}			$\begin{bmatrix} 1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$ $\begin{bmatrix} 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\ 1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&1&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&0&-1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&-1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&-1&0&0&0&0 \end{bmatrix}$ $\begin{bmatrix} -1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&-1&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}$

A Polynomial Results

Theorem A.1. *For partitions λ, μ of n, we have*

$$\sum_i \frac{1}{1^{i_1} i_1! 2^{i_2} i_2! \cdots n^{i_n} i_n!} \omega_\lambda(i) \omega_\mu(i) = \delta_{\lambda\mu} = \begin{cases} 1 & if \lambda = \mu \\ 0 & otherwise \end{cases}$$

Theorem A.2. *This is formula (A.8) in [7]. To multiply two Schur polynomials the formula*

$$S_\lambda \cdot S_\mu = \sum N_{\lambda\mu\nu} S_\nu$$

with the sum over all partitions of $\ell + m$ can be used. The Littlewood–Richardson rule says the $N_{\lambda\mu\nu}$ are given by the number of ways that the Young diagram of λ can be expanded to the Young diagram of ν by a strict μ-expansion.

If $\mu = (\mu_1, \mu_2, \ldots, \mu_k)$ then a μ-expansion of λ is obtained by adding μ_1 boxes to λ, such that no two are in the same column. Label these boxes with a 1. Similarly add μ_i boxes in order for all i. To determine if an expansion is called strict, list the integer labels from right to left, starting on the top row and working down. If each integer $1 \leq p \leq k-1$ occurs at least as many times as the next integer $p+1$ in the first t elements of this list for all positive integers t, then the expansion is called strict. For example, the expansion $\begin{array}{|c|c|c|c|} \hline \ & \ & 1 & 1 \\ \hline \ & 2 \\ \cline{1-2} \end{array}$ *is strict, but* $\begin{array}{|c|c|c|c|} \hline \ & \ & 1 & 2 \\ \hline \ & 1 \\ \cline{1-2} \end{array}$ *is not.*

B GAP Code

In this section we include the GAP source code used to perform the calculations in this paper.

```
# Procedures useful in converting partitions of n into explicit
# matrices for representations of S_n, and pairs of partitions
# into explicit matrices for representations of W_n, the
# hyperoctahedral group (aka Coxeter Group of type B_n, Cube Group,
# etc.)

# The Singular package is much faster than the built in Groebner
# Basis!
LoadPackage("singular");
GBASIS:= SINGULARGBASIS;;

# The directory to store all the calculations
outdir := "/Users/gvol/thesis/calculations/";

# There should be a better way (built in), but I can't seem to find
# it.  Why can't Display() be made to return a string?
Pretty := function(M)
    local i, str;
    str := Concatenation("[ ", String(M[1]));
    for i in [2..Length(M)] do
        str := Concatenation(str, ",\n  ", String(M[i]));
    od;
    return  Concatenation(str, " ]");
end;

# Return generators of the CubeGroup on n elements, starting with
# elements offset+1.
CubeGroupGens:= function(n,offset)
    # Generators:
    # 1 ϵ1
    # i (i i+1)
    local s, i;
    s:= List([1..n], x->[1..100+offset+n]);
    s[1]{[offset+1, 100+offset+1]}:= [100+offset+1, offset+1];
    for i in [2+offset..n+offset] do
        s[i-offset]{[i-1, i, 100+i-1, 100+i]}:= [i, i-1, 100+i, 100+i-1];
    od;
    return List(s, PermList);
end;

# Return generators of the SymmetricGroup on n elements, starting with
# elements offset+1
SymmetricGroupGens:= function(n,offset)
    local s, i;
    s:= List([1..n], x->[1..100+offset+n]);
    for i in [2+offset..n+offset] do
        s[i-offset]{[i-1, i, 100+i-1, 100+i]}:= [i, i-1, 100+i, 100+i-1];
```

```
    od;
    return List(s, PermList);
end;

# Returns the Rox group for a pair of partitions.
# Call as CoxRowGroup(lambda, []) for the Symmetric Group
CoxRowGroup := function(lambda, mu)
    local i,tot, SCur, gens;
    gens:=[];    tot:=0;

    for i in lambda do
        Append(gens,SymmetricGroupGens(i,tot));
        tot:= tot + i;
    od;
    for i in mu do
        Append(gens,CubeGroupGens(i,tot));
        tot := tot + i;
    od;
    return Group(gens,()); # In case gens is empty
end;

# Returns the Column group for a pair of partitions.
# Call as CoxColGroup([], lambda) for the Symmetric Group
CoxColGroup := function(lambda, mu)
    local i, j, tot, SCur, gens;
    gens:=[];    tot := 0;

    if Length(lambda)>0 then
        for i in [1 .. lambda[1]] do
            Add(gens, (i,100+i) );
        od;
        tot := lambda[1];
        for i in [2 .. Length(lambda)] do
            for j in [1 .. lambda[i] ] do
                Add(gens,
                    (     tot+j-lambda[i-1],     tot+j)
                    (100+tot+j-lambda[i-1],100+tot+j) );
            od;
            tot := tot + lambda[i];
        od;
    fi;

    if Length(mu)>0 then
        tot := tot + mu[1];
        for i in [2 .. Length(mu)] do
            for j in [1 .. mu[i] ] do
                Add(gens,
                    (     tot+j-mu[i-1],     tot+j)
                    (100+tot+j-mu[i-1],100+tot+j) );
            od;
            tot := tot + mu[i];
        od;
    fi;
    return Group(gens,());
```

```
end;

# Returns the direct sum of two matrices.  It's a shame this isn't
# built in.
DirectSumOfMatrices := function(A,B)
    if A=[] then return B; fi;
    if B=[] then return A; fi;
    return
      Concatenation(
              List([1..Length(A)],
                   i->Concatenation(
                            A[i],
                            List([1..Length(B[1])], i->0))),
              List([1..Length(B)],
                   i->Concatenation(
                            List([1..Length(A[1])], i->0),
                            B[i])));
end;

# Determinant map of interest in Cox(B_n)
Det := function(perm)
    # Right now this is arbitrarily limited to n=15 hoping that it
    # will improve the speed.
    # This works faster than the homomorphism version which I tried.
    if perm=() then
        return 1;
    else
        return
          SignPerm(perm) *
          SignPerm(PermList(List(ListPerm(perm){[1..15]},
                  i->Minimum(AbsoluteValue(i-100),i))));
    fi;
end;

# Returns a list of the Hook Lengths of a partition (in no particular order).
HookLengths := function(lambda)
    # This is used it to calculate the dimension of a partition.
    # It fairly fast and somewhat interesting, so I'll keep it even
    # though it's not strictly necessary.
    local HL, mu, i, newPart;

    if Length(lambda) = 0 then return []; fi;
    if Length(lambda) = 1 then return [1..lambda[1]]; fi;
    HL := HookLengths(lambda{[ 2..Length(lambda) ]});
    mu := AssociatedPartition(lambda);
    for i in [1 .. lambda[1]] do
        Add(HL, lambda[1]-i+mu[i]);
    od;
    return HL;
end;

# Returns the dimension of a pair of partitions
# Call as DimPart(lambda, []) for S_n
DimPart := function(lambda,mu)
```

```
    return
      Factorial(Sum(lambda)+Sum(mu)) /
      ( Product(HookLengths(lambda)) * Product(HookLengths(mu)) );
end;

# Returns a unique string for every partition of n
SymPartitionToIdentifier := function(lambda)
    return JoinStringsWithSeparator(
                    Concatenation(["S",String(Sum(lambda)),"_"],
                            List(lambda, String)) ,"");
end;

# Returns a unique string for every pair of partitions of n
CoxPartitionToIdentifier :=  function(lambda,mu)
    return JoinStringsWithSeparator(
                    Concatenation(["W",String(Sum(lambda)+Sum(mu)),"_"],
                            List(lambda, String),
                            ["_"],List(mu, String)) ,"");
end;

# Take the Young symmetrizer c, a group G, the group ring CG, and a
# dimension.  Return a basis for CGc.
cToBasis := function(c,G,CG,dim)
    local basis, p, bob, alpha, b, coef, i, j;

    basis := [c];
    # Stop once we have enough basis elements
    for i in [1..dim] do
        for alpha in GeneratorsOfGroup(G) do
            bob := alpha * basis[i];
            for b in basis do
                coef := CoefficientsAndMagmaElements(bob);
                p := Position(coef, (CoefficientsAndMagmaElements(b)[1] ));
                if p <> fail then
                    bob := bob - coef[p+1] * b;
                fi;
                if bob=Zero(CG) then break; fi;
            od;

            if bob<>Zero(CG) then
                bob := bob / CoefficientsAndMagmaElements(bob)[2];
                # Reduce other basis elements in terms of the new one
                for j in [1..Length(basis)] do
                    coef := CoefficientsAndMagmaElements(basis[j]);
                    p := Position(coef,
                             CoefficientsAndMagmaElements(bob)[1] );
                    if p<>fail then
                        basis[j] := basis[j] - coef[p+1] * bob;
                    fi;
                od;
                Add(basis, bob);
                if Length(basis)=dim then return basis; fi;
            fi;
        od;
```

```
    od;
    return basis;
end;

# Returns a matrix corresponding to the action of Perm on a basis.
# Now I find out that there may be the same thing built in.
MatrixFromActionOnBasis := function(basis,Perm)
    local bP, PCol, Pmat, p, b;

    Pmat := [];    PCol:=[];
    for b in basis do
        bP := Perm * b;
        PCol:= List(basis,
                    b -> Position(CoefficientsAndMagmaElements(bP),
                            CoefficientsAndMagmaElements(b)[1]));
        for p in [1..Length(PCol)] do
            if PCol[p]=fail then
                PCol[p]:=0;
            else
                PCol[p]:=CoefficientsAndMagmaElements(bP)[PCol[p]+1];
            fi;
        od;
        Add(Pmat,PCol);
    od;
    return TransposedMat(Pmat);
end;

# Explicitly calculate the matrices corresponding to P, Q, S for
# a pair of partitions lambda, mu.
CalculateCoxPartitionToMatrix := function(lambda, mu)
    local rowGp, colGp, alpha, a, b, c, Wn, CWn, emb, bob, basis,
          coef, PCol, Pmat, bP, Perm, Q, i, j, U,
          Urow, gamma, d, p, filename, P;
    rowGp := CoxRowGroup(lambda,mu);
    colGp := CoxColGroup(lambda,mu);

    Wn := Group(CubeGroupGens( Sum(lambda) + Sum(mu), 0));
    CWn:= GroupRing(Rationals, Wn);
    emb:= Embedding(Wn,CWn);

    a := Zero(CWn);
    c := Zero(CWn);
    for alpha in rowGp do
        a := a + alpha^emb;
    od;

    for alpha in colGp do
        c := c + Det(alpha)*alpha^emb;
    od;

    # Calculate whichever is more efficient
    if Length(CoefficientsAndMagmaElements(a)) <
       Length(CoefficientsAndMagmaElements(c)) then
        c:=a*c*a;
```

```
    else
        c:=c*a*c;
    fi;

    coef := CoefficientsAndMagmaElements(c);
    # Normalize
    c := c/coef[2];
    basis := cToBasis(c,Wn,CWn,DimPart(lambda,mu));
    # Here I use the fact that Wn.1 is (1, -1) and all the rest are
    # (i, i+1) in order
    Q:=Wn.1;
    for i in GeneratorsOfGroup(Wn) do
        Q:= i*Q;
    od;
    if Length(GeneratorsOfGroup(Wn)) > 1 then
        P := Wn.2;
    else
        P := ();
    fi;
    return rec(lambda:= lambda,
               mu:= mu,
               n:= Sum(lambda)+Sum(mu),
               P:= StructuralCopy(MatrixFromActionOnBasis(basis,P)),
               Q:= StructuralCopy(MatrixFromActionOnBasis(basis,Q)),
               S:= StructuralCopy(MatrixFromActionOnBasis(basis,Wn.1)));
end;

# Return a list of [character number, character] where character
# number is the number of the character in the character table, and
# character is the character of the representation on the conjugacy
# classes.
CoxCharacterOfRep := function(B)
    local x, elem, Wn, char, MatGroup, phi, MatBasis, P, Q, i, n,
          permlist, LL, SconjC, conjC;

    Wn := CubeGroupGens(B.n,0);
    Q:=Wn[1];
    for i in Wn do
        Q:= i*Q;
    od;
    if Length(Wn) > 1 then
        P := Wn[2];
    else
        P := ();
    fi;
    MatGroup := Group([B.P, B.Q, B.S]);
    # Make sure that the group has the right generators for the homo.
    # We keep the original Wn to ensure the conjugacy classes are in
    # the same order as those from the character table
    phi := GroupHomomorphismByImages(Group([P, Q, Wn[1] ]), MatGroup,
                    [P, Q, Wn[1] ], GeneratorsOfGroup(MatGroup));
    if phi = fail then
        Print("There is something wrong!!\nCould not make a homomorphism.",0);
        return fail;
```

```
    fi;
    Wn:= Group(Wn);      char := [];

    # Ensure that characters are always in the same order
    conjC := ShallowCopy(ConjugacyClasses(Wn));
    SconjC := ShallowCopy(conjC);
    Sort(SconjC);
    elem := List(SconjC, x->Representative(x));
    for x in elem do
        Add(char, Trace(Image(phi,x)) );
    od;
    LL := List(Irr(Wn),ShallowCopy);
    permlist := List(conjC, i->Position(SconjC ,i) );
    for x in LL do
        x{permlist} := x{[1..Length(LL)]};
    od;
    return [Position(LL, char), char];
end;

# Explicitly calculate the matrices corresponding to P, Q, Q_{n-1}
# for a partition lambda.  Useful for calculating the representation
# of both S_n, and S_{n+1}
CalculateSymPartitionToMatrix := function(lambda)
    local rowGp, colGp, alpha, a, b, c, Wn, CWn, emb, bob, basis,
          coef, coef2, p, PCol, Pmat, bP, Perm, Q, i, Q1, P, n;

    rowGp := CoxRowGroup(lambda,[]);
    colGp := CoxColGroup([],lambda);
    Wn := Group(SymmetricGroupGens( Sum(lambda), 0));
    CWn:= GroupRing(Rationals, Wn);
    emb:= Embedding(Wn,CWn);

    a := Zero(CWn);
    c := Zero(CWn);
    for alpha in rowGp do
        a := a + alpha^emb;
    od;
    for alpha in colGp do
        c := c + Det(alpha)*alpha^emb;
    od;
    c:=a*c;

    # Normalize
    coef := CoefficientsAndMagmaElements(c);
    c := c/coef[2];
    basis := cToBasis(c,Wn,CWn,DimPart(lambda,[]));

    Q := Wn.1;
    for i in GeneratorsOfGroup(Wn) do
        Q := i*Q;
    od;
    if Length(GeneratorsOfGroup(Wn)) > 1 then
        P := Wn.2;
        Q1:= Wn.(Length(GeneratorsOfGroup(Wn))) *Q;
```

```
    else
        P := ();
        Q1:= ();
    fi;
    return rec(lambda:= lambda,
               n:= Sum(lambda),
               P:= StructuralCopy(MatrixFromActionOnBasis(basis,P)),
               Q:= StructuralCopy(MatrixFromActionOnBasis(basis,Q)),
               Q1:= StructuralCopy(MatrixFromActionOnBasis(basis,Q1)));
end;

# Return a list of [character number, character] where character
# number is the number of the character in the character table, and
# character is the character of the representation on the conjugacy
# classes.
SymCharacterOfRep := function(B)
    local x, elem, char, MatGroup, phi, MatBasis, P, Q, i, n, Sn,
          permlist, LL, SconjC, conjC;

    Sn := SymmetricGroupGens(B.n, 0);
    Q := Sn[1];
    for i in Sn do
        Q := i*Q;
    od;
    if Length(Sn) > 1 then
        P := Sn[2];
    else
        P := ();
    fi;
    MatGroup := Group([B.P, B.Q]);
    # Make sure that the group has the right generators for the homo.
    # We keep the original Wn to ensure the conjugacy classes are in
    # the same order as those from the character table
    phi := GroupHomomorphismByImages(Group([P, Q]), MatGroup,
                  [P, Q], GeneratorsOfGroup(MatGroup));
    if phi = fail then
        Print("There is something wrong!!\nCould not make a homomorphism.",0);
        return fail;
    fi;

    Sn := Group(Sn);
    char := [];
    # To ensure that characters are always in the same order
    conjC := ShallowCopy(ConjugacyClasses(Sn));
    SconjC := ShallowCopy(conjC);
    Sort(SconjC);

    elem := List(SconjC, x->Representative(x));
    for x in elem do
        Add(char, Trace(Image(phi,x)) );
    od;
    LL := List(Irr(Sn),ShallowCopy);
    permlist := List(conjC, i->Position(SconjC ,i) );

```

```
    for x in LL do
        x{permlist} := x{[1..Length(LL)]};
    od;
    return [Position(LL, char), char];
end;


# Return the character of a representation of S_{n+1} restricted to S_n
SymCharacterOfRestrictedRep := function(B)
    local permlist, SconjC, conjC, char, CharTab1, CharTab, Sn1, Sn;

    if( B.n = 1) then
        return [];
    fi;

    Sn  := Group(SymmetricGroupGens(B.n, 0));
    Sn1 := CharacterTable(Subgroup(Sn,SymmetricGroupGens(B.n-1,0)));
    CharTab  := Irr(Sn);
    CharTab1 := Irr(Sn1);
    char := ShallowCopy(RestrictedClassFunction(CharTab[B.charno], Sn1));

    # To ensure that characters are always in the same order
    conjC := ShallowCopy(ConjugacyClasses(Sn1));
    SconjC := ShallowCopy(conjC);
    Sort(SconjC);
    permlist := List(conjC, i->Position(SconjC ,i) );
    char{permlist} := char{[1..Length(char)]};

    return char;
end;

# Forward declaration to avoid a syntax warning.
GetMatForAllSymPartsOf:=0;

# Return tables of the partitions of n in order of their character
# number
SymCharacterLookupTable := function(n)
    local mu, lambda, CharSn, CharWn, Sn1, Wn, SymReps, CoxReps, rep,
          i, LookUpList, SymLookUpList, CoxLookUpList, Sn;

    SymReps := GetMatForAllSymPartsOf(n);
    Sn := Group(SymmetricGroupGens(n,0));
    CharSn := List(Irr(Sn),ShallowCopy);
    SymLookUpList:=[];
    for i in [1..Length(CharSn)] do
        for rep in SymReps do
            if rep.char = CharSn[i] then
                lambda := rep.lambda;
                break;
            fi;
        od;
        Add(SymLookUpList,lambda);
    od;
    return SymLookUpList;
```

```
end;

# Decompose a representation of S_{n+1} to it's irreducible components
# in S_n
SymComponentsOfRestrictedRep := function(B)
    local i, c, chiList, SymLookUp, chis, chi,CharTab1, char, CharTab,
          Sn1, Sn;

    if(B.n = 1) then
        return [];
    fi;

    Sn   := Group(SymmetricGroupGens(B.n, 0));
    Sn1  := CharacterTable(Subgroup(Sn,SymmetricGroupGens(B.n-1,0)));
    CharTab   := Irr(Sn);
    CharTab1  := Irr(Sn1);

    chi  := RestrictedClassFunction(CharTab[B.charno], Sn1);
    chis  := ConstituentsOfCharacter( chi );
    SymLookUp  := SymCharacterLookupTable(B.n - 1);

    chiList  :=[];
    for c in chis do
        for i in [1..ScalarProduct(c, chi)] do
            Add(chiList,SymLookUp[Position(CharTab1,c)]);
        od;
    od;

    return chiList;
end;

# Return the character of a representation of W_n when restricted to
# S_n
CoxCharacterOfRestrictedRep := function(B)
    local permlist, SconjC, conjC, char, chis, chi, CharTab1, CharTab,
          Sn, Wn;

    if(B.n = 1) then
        return [];
    fi;

    Wn   := Group(CubeGroupGens(B.n, 0));
    Sn  := CharacterTable(Subgroup(Wn, SymmetricGroupGens(B.n,0)));
    CharTab   := Irr(Wn);
    CharTab1  := Irr(Sn);

    chi  := RestrictedClassFunction(CharTab[B.charno], Sn);
    chis  := ConstituentsOfCharacter( chi );
    char  := ShallowCopy(RestrictedClassFunction(CharTab[B.charno], Sn));

    # To ensure that characters are always in the same order
    conjC  := ShallowCopy(ConjugacyClasses(Sn));
    SconjC  := ShallowCopy(conjC);
    Sort(SconjC);
```

```
    permlist := List(conjC, i->Position(SconjC ,i) );
    char{permlist} := char{[1..Length(char)]};

    return char;
end;

# Decompose a representation of W_n to it's irreducible components
# in S_n
CoxComponentsOfRestrictedRep := function(B)
    local i, c, chiList, SymLookUp, chis, chi,CharTab1, char,
          CharTab, Sn1, Sn, Wn, CoxLookUp;

    if(B.n = 1) then
        return [];
    fi;

    Wn  := Group(CubeGroupGens(B.n, 0));
    Sn := CharacterTable(Subgroup(Wn,SymmetricGroupGens(B.n,0)));
    CharTab  := Irr(Wn);
    CharTab1 := Irr(Sn);
    chi := RestrictedClassFunction(CharTab[B.charno], Sn);
    chis := ConstituentsOfCharacter( chi );
    SymLookUp := SymCharacterLookupTable(B.n);
    chiList :=[];
    for c in chis do
        for i in [1..ScalarProduct(c, chi)] do
            Add(chiList,SymLookUp[Position(CharTab1,c)]);
        od;
    od;
    return chiList;
end;

# Usually this is the function that you want to call.  It returns a
# list of the matrices corresponding to P, Q, S, for a pair of
# partitions.  It reads them from disk if possible, or calculates them
# if necessary, in which case it stores them to disk in a human and
# machine readable format.
CoxPartitionToMatrix := function(lambda, mu)
    local bob, filename, B, S, Q, P, StartTime, out, str, chars,
          rchars, char, charno, dimension, n, rcomp, rchar, time;

    filename := CoxPartitionToIdentifier(lambda, mu);
    bob := Concatenation(outdir, filename);
    if IsReadableFile( bob ) then
        Read(bob);
        return EvalString(filename);
    else
        str := "";
        out := OutputTextString(str,true);
        AppendTo(out, filename,"␣:=␣rec(\n#␣\n");

        StartTime := Runtime();
        B := CalculateCoxPartitionToMatrix(lambda,mu);
        AppendTo(out, "#␣Time␣taken:\n");
```

```
        AppendTo(out, "     time:= ",Runtime()- StartTime,",\n# \n");

        AppendTo(out, "# Pair of Partitions\n");
        AppendTo(out, "   lambda:= ",lambda,",\n");
        AppendTo(out, "       mu:= ",     mu,",\n");
        AppendTo(out, "# of\n");
        AppendTo(out, "        n:= ", Sum(lambda)+Sum(mu),", \n");
        AppendTo(out, "# with\n");
        AppendTo(out, "dimension:= ", DimPart(lambda,mu) ,",\n");

        chars := CoxCharacterOfRep(B);
        B.charno := chars[1];
        B.char := chars[2];
        AppendTo(out, "  charno := ", B.charno, ",\n");
        AppendTo(out, "    char := ", B.char, ",\n");

        rchar := CoxCharacterOfRestrictedRep(B);
        AppendTo(out, "   rchar := ", rchar, ",\n");
        rcomp := CoxComponentsOfRestrictedRep(B);
        AppendTo(out, "   rcomp := ", rcomp, ",\n");

        AppendTo(out, "# \n# Matrices:\n# \n");
        AppendTo(out, "P :=\n");
        AppendTo(out, Pretty(B.P),",\n");
        AppendTo(out, "Q :=\n");
        AppendTo(out, Pretty(B.Q),",\n");
        AppendTo(out, "S :=\n");
        AppendTo(out, Pretty(B.S),"\n);\n");

        CloseStream(out);
        PrintTo(Concatenation([outdir,filename]), str);
        return B;
    fi;
end;

# Usually this is the function that you want to call.
# It returns a list of the matrices corresponding to P, Q,
# for a partition.  It reads them from disk,  or calculates
# them if necessary, in which case it stores them to disk in a
# human and machine readable format.
SymPartitionToMatrix := function(lambda)
    local bob, filename, B, out, Q, P, StartTime, str, Q1, char,
          alpha, rcharno, rchar, charno, dimension, n, rcomp, time,
          chars;

    filename := SymPartitionToIdentifier(lambda);
    bob := Concatenation(outdir, filename);
    if IsReadableFile( bob ) then
        Read(bob);
        return EvalString(filename);
    else
        str := "";
        out := OutputTextString(str,true);
        AppendTo(out, filename," := rec(\n# \n");
```

```
        StartTime := Runtime();
        B := CalculateSymPartitionToMatrix(lambda);
        AppendTo(out, "# Time taken:\n");
        AppendTo(out, "     time:= ",Runtime()- StartTime,",\n# \n");

        AppendTo(out, "# Partition\n");
        AppendTo(out, "   lambda:= ",lambda,",\n");
        AppendTo(out, "# of\n");
        AppendTo(out, "        n:= ", Sum(lambda),", \n");

        AppendTo(out, "# with\n");
        AppendTo(out, "dimension:= ", DimPart(lambda,[]) ,",\n");

        chars := SymCharacterOfRep(B);
        B.charno := chars[1];
        B.char := chars[2];
        AppendTo(out, "  charno := ", B.charno, ",\n");
        AppendTo(out, "    char := ", B.char, ",\n");

        rchar := SymCharacterOfRestrictedRep(B);
        AppendTo(out, "   rchar := ", rchar, ",\n");
        rcomp := SymComponentsOfRestrictedRep(B);
        AppendTo(out, "   rcomp := ", rcomp, ",\n");

        AppendTo(out, "# \n# Matrices:\n# \n");
        AppendTo(out, "P :=\n");
        AppendTo(out, Pretty(B.P),",\n");
        AppendTo(out, "Q :=\n");
        AppendTo(out, Pretty(B.Q),",\n");
        AppendTo(out, "Q1 :=\n");
        AppendTo(out, Pretty(B.Q1),",\n");
        AppendTo(out, "alpha :=\n");   # a_n in the paper
        AppendTo(out, Pretty(B.Q*B.Q1^1),"\n);\n");

        CloseStream(out);
        PrintTo(Concatenation([outdir,filename]), str);

        return B;
    fi;
end;

# Returns all pairs of partitions of n
PairsOfPartitions := function(n)
    local mu, lambda, i, AllReps;
    AllReps :=[];
    for i in [0..n] do
        for lambda in Partitions(i) do
            for mu in Partitions(n-i) do
                Add(AllReps, [lambda,mu]);
            od;
        od;
    od;
    return AllReps;
```

```
end;

# Force calculation of all representations of W_n for a given n.
GetMatForAllCoxPartsOf := function(n)
    local i, lambda, mu, AllReps;
    AllReps:=[];
    for i in [0..n] do
        for lambda in Partitions(i) do
            for mu in Partitions(n-i) do
                Add(AllReps, CoxPartitionToMatrix(lambda,mu));
            od;
        od;
    od;
    return AllReps;
end;

# Force calculation of all representations of S_n for a given n.
GetMatForAllSymPartsOf := function(n)
    local lambda, AllReps;
    AllReps:=[];
    for lambda in Partitions(n) do
        Add(AllReps, SymPartitionToMatrix(lambda));
    od;
    return AllReps;
end;

# Returns a list of columns of W_n to compare to a list of columns of
# S_n based on conjugacy classes.
ColsToKeep := function(n)
    local colsWn, colsSn, x, y, Sn1, Wn;

    Wn := Group(CubeGroupGens(n,0));
    Sn1 := Group(SymmetricGroupGens(n+1,0));
    colsWn:=[];
    colsSn:=[];
    for x in ConjugacyClasses(Sn1) do
        for y in ConjugacyClasses(Wn) do
            if IsConjugate(Sn1, Representative(x), Representative(y)) then
                Add(colsWn,Position(ConjugacyClasses(Wn),y));
                Add(colsSn,Position(ConjugacyClasses(Sn1),x));
            fi;
        od;
    od;
    return [colsWn,colsSn];
end;

# Flattens, removes 0, and then returns a Reduced Groebner Basis
ToGrobBasis := function( relations )
    relations := Set(Flat(relations));
    if Length(relations)=0 then return []; fi;

    if relations[1] = relations[1]-relations[1] then
        relations := relations{[2..Length(relations)]};
    fi;
```

```
    if Length(relations)=0 then return []; fi;
    return GroebnerBasis(relations,MonomialGrlexOrdering());
end;

# Reduces the polynomials in U by relations
ReduceMatModIdeal := function(U,relations)
    local i, j;
    for i in [1..Length(U)] do
        for j in [1..Length(U)] do
            if IsPolynomial(U[i][j]) then
               U[i][j] := PolynomialReducedRemainder( U[i][j], relations,
                                    MonomialGrlexOrdering() );
            fi;
        od;
    od;
    return U;
end;

# Return an n*n matrix with unique indeterminates in each position
IndeterminateMatrix := function(n)
    local i, j, Urow, U;
    U:=[];
    for i in [1..n] do
        Urow:=[];
        for j in [1..n] do
            Add(Urow,Indeterminate(Rationals,
                    JoinStringsWithSeparator( ["U",String(i),String(j)],
                            "_")));
        od;
        Add(U,Urow);
    od;
    return StructuralCopy(U);
end;

# Tries to extend a representation of W_n to a representation of
# Aut(F_n) by calculating possible U matrices.  Must be given an
# initial U matrix.
DoCalculateU := function(reps, U)
    local charCox, Q, P, relations, Qi, S, rep, b,
          hardrels, s, indet, B, sols, roots, n, r, root, UU;

    n := reps.n;
    relations := [];
    P := reps.P;
    Q := reps.Q;
    Qi := Inverse(Q);
    S := reps.S;

    # From experience it seems that the first few are able to
    # introduce several zeros into U, so we reduce by them to simplify
    # later calculations.
    #1
    if n<>2 then
```

```
        if n <> 3 then
            Append(relations, U * Qi^2*P*Q^2 - Qi^2*P*Q^2 * U);
            relations := ToGrobBasis(relations);
            U := ReduceMatModIdeal(U,relations);
        fi;
        #2
        Append(relations, U * Q*P*Qi*P*Q - Q*P*Qi*P*Q * U);
        relations := ToGrobBasis(relations);
        U := ReduceMatModIdeal(U,relations);
        #3
        Append(relations, U * Qi^2*S*Q^2 - Qi^2*S*Q^2 * U);
        relations := ToGrobBasis(relations);
        U := ReduceMatModIdeal(U,relations);
    fi;
    #4
    if n <> 3 then
        Append(relations, U * Qi^2*U*Q^2 - Qi^2*U*Q^2 * U);
        relations := ToGrobBasis(relations);
        U := ReduceMatModIdeal(U,relations);
    fi;
    #5
    Append(relations, U * S*U*S - S*U*S * U);
    #6
    Append(relations, U * P*Qi*S*U*S*Q*P - P*Qi*S*U*S*Q*P * U);
    #7
    if n <> 2 then
        Append(relations,U * P*Qi*P*Q*P*U*P*Qi*P*Q*P -
               P*Qi*P*Q*P*U*P*Qi*P*Q*P * U);
    fi;
    #8
    if n <> 2 then
        Append(relations, (P*Qi*U*Q)^2 * U*Qi - U*Qi*U);
    fi;
    #9
    Append(relations, (P*S*P*U)^2 - U^0 );
    #10
    if n <> 1 then
        Append(relations, P*U*P*S*U*S*P*S - U);
        relations := ToGrobBasis(relations);
        U := ReduceMatModIdeal(U,relations);
    fi;

    hardrels :=[]; roots :=[];
    for b in relations do
        if String(b) = "1" then return false; fi;
        if IsUnivariatePolynomial(b) then
            indet:= IndeterminateOfUnivariateRationalFunction(b);
            s := String(indet);
            if Length(RootsOfUPol(b))>1 then
                s := SplitString(s, "_" );
                Add(roots, [indet, RootsOfUPol(b)]);
                Add(hardrels, b);
            fi;
        else
```

```
            Add(hardrels, b);
        fi;
    od;
    return [U, hardrels];
end;

# Sort representations lexicographically
repLT := function(a,b)
    if a.dimension < b.dimension then
        return true;
    elif a.dimension = b.dimension then
        if a.lambda < b.lambda then
            return true;
        elif a.lambda = b.lambda and a.mu < b.mu then
            return true;
        fi;
    fi;
    return false;
end;

# Make a Representation record by taking the direct sum of a list of
# representations
DirectSumOfReps := function(reps)
    local U, Qi, dimTotal, S, Q, P, n, rep, lt, charTotal, parts,
          char, dimension, rcharTotal, rchar, Q1;
    Sort(reps, repLT);

    n := reps[1].n;
    P := []; Q := []; S := []; Q1 := [];
    dimTotal := 0;
    charTotal := 0*reps[1].char;
    rcharTotal := 0*reps[1].rchar;
    parts := [];
    for rep in reps do
        if rep.n <> n then
            return fail;
        fi;
        dimTotal := dimTotal + rep.dimension;
        charTotal := charTotal + rep.char;
        rcharTotal := rcharTotal + rep.rchar;

        if IsBound(rep.mu) then
            Add(parts, [rep.lambda, rep.mu]);
        else
            Add(parts, rep.lambda);
        fi;
        P := DirectSumOfMatrices( P, rep.P);
        Q := DirectSumOfMatrices( Q, rep.Q);
        if IsBound(rep.S) then
            S := DirectSumOfMatrices( S, rep.S);
        fi;
        if IsBound(rep.Q1) then
            Q1 := DirectSumOfMatrices( Q1, rep.Q1);
        fi;
```

```
    od;
    if Q1 = [] then
        return rec( n := n,
                    parts:=parts,
                    dimension := dimTotal,
                    char := charTotal,
                    rchar:= rcharTotal,
                    P := P,
                    Q := Q,
                    S := S,
                );
    else
        return rec( n := n,
                    parts:=parts,
                    dimension := dimTotal,
                    char := charTotal,
                    rchar:= rcharTotal,
                    P := P,
                    Q := Q,
                    Q1 := Q1,
                );
    fi;
end;

# Tries to calculate a possible U matrix for the direct sum of a list
# of representations of W_n
CalculateU := function(reps)
    local UU, sols, S, Q, P, s, indet, roots, hardrels, relations, U,
          B, BB, allrels, b, r, dimension, n;

    reps := DirectSumOfReps(reps);
    B := DoCalculateU(reps, IndeterminateMatrix( reps.dimension ));
    if B = false then return false; fi;
    U := B[1];  relations := B[2];
    hardrels :=[]; roots :=[];
    allrels := [];

    # Recalculate to try and simplify the relations returned
    B := DoCalculateU(reps, U);
    U := B[1];  relations := B[2];

    for b in relations do
        if String(b) = "1" then return false; fi;
        if IsUnivariatePolynomial(b) then
            indet:= IndeterminateOfUnivariateRationalFunction(b);
            s := String(indet);
            if Length(RootsOfUPol(b))>1 then
                s := SplitString(s, "_" );
                Add(roots, [indet, RootsOfUPol(b)]);
                Add(hardrels, b);
            fi;
        else
            Add(hardrels, b);
        fi;
```

```
    od;
    if Length(roots) <> 1 then
        return rec( n := reps.n,
                    dimension := reps.dimension,
                    P := reps.P,
                    Q := reps.Q,
                    S := reps.S,
                    U := [U],
                    relations := [hardrels],
                    );
    else
        sols :=[];
        roots := roots[1];
        for r in roots[2] do
            UU := StructuralCopy(U);
            UU := ReduceMatModIdeal(UU,
                        Concatenation(hardrels,[roots[1]-r]));
            BB := DoCalculateU(reps, UU);
            Add(sols,BB[1]);
            Add(allrels, StructuralCopy(BB[2]));
        od;
        return rec( n := reps.n,
                    dimension := reps.dimension,
                    P := reps.P,
                    Q := reps.Q,
                    S := reps.S,
                    U := sols,
                    relations := allrels
                    );
    fi;
    return CalculateU( reps, B[1] );
end;

# The highest level function to look for U matrices. It attempts to
# read from disk. If unsuccessful, it will call DoCalculateU to
# calculate it.
FindU := function(reps)
    local bob, filename, B, S, Q, P, StartTime, out, str, chars,
          stillLeft, U, char, dimension, n, mu, lambda, rep, thisPart,
          Parts, joe, unextendables, f, b, r, y, l, XX, p, x, i,
          relations, parts;

    Parts:=[];
    Sort(reps,repLT);
    f := JoinStringsWithSeparator(
           Concatenation([ "U"],
                 List(reps, i-> JoinStringsWithSeparator(
                        Concatenation( List(i.lambda, String),
                               ["_"],List(i.mu, String)) ,""),"" )
                 ),"--");

    bob := Concatenation(outdir, f);
    unextendables := "U--not";
    joe := Concatenation(outdir, unextendables);
```

```
    if IsReadableFile( bob ) then
        Read(bob);
        return EvalString("rep");
    elif not IsReadableFile( joe ) then
        PrintTo(joe, "parts␣:=␣[];");
    fi;

    Read(joe);
    thisPart := List(reps, i->[i.lambda, i.mu]);
    if Position(EvalString("parts"), thisPart) <> fail then
        return false;
    fi;

    str := "";
    out := OutputTextString(str,true);
    AppendTo(out, "rep␣:=␣rec(\n#␣\n");

    StartTime := Runtime();
    B := CalculateU(reps);
    AppendTo(out, "#␣Time␣taken:␣",Runtime()- StartTime,"\n#␣\n");
    if B = false then
        Parts := EvalString("parts");
        Add(Parts,thisPart);
        PrintTo(joe,Concatenation([ "parts␣:=␣", String(Parts),";" ]));
        return false;
    else
        AppendTo(out, "#␣\n#␣Matrices:\n#␣\n");
        AppendTo(out, "P␣:=\n");
        AppendTo(out, Pretty(B.P),",\n");
        AppendTo(out, "Q␣:=\n");
        AppendTo(out, Pretty(B.Q),",\n");
        AppendTo(out, "S␣:=\n");
        AppendTo(out, Pretty(B.S),",\n");
        AppendTo(out, "U␣:=\n");
        AppendTo(out, Pretty(B.U),",\n");
        AppendTo(out, "relations␣:=\n");
        AppendTo(out, B.relations,"\n);\n");
        CloseStream(out);

        for p in Flat(IndeterminateMatrix(B.n)) do
            XX:= List(Flat(B.U), i->String(i));
            l:=Length(String(p));
            stillLeft := false;
            for x in XX do
                for i in [1..Length(String(x))-l+1] do
                    y := Concatenation([x,"␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣"]);
                    if y{[i..i+l-1]}=String(p) then
                        stillLeft := true;
                    fi;
                od;
            od;

            if stillLeft then
                str := Concatenation(
```

```
                                String(p)," := Indeterminate(Rationals,\"",
                                String(p),"\");\n",
                                str);
                fi;
            od;
            PrintTo(bob, str);
            return B;
    fi;
end;

# Internal function for recursively creating all representations of a
# certain dimension.
RepsMatchingDim := function(dim,Reps)
    local x, AllReps;

    if dim = 0 then return []; fi;
    if Length(Reps) = 0 then return []; fi;
    AllReps := RepsMatchingDim(dim, Reps{[2..Length(Reps)]});

    if Reps[1].dimension = dim then
        Add(AllReps,[Reps[1]]);
    elif Reps[1].dimension < dim then
        Append(AllReps, List(
                RepsMatchingDim(dim - Reps[1].dimension, Reps),
                y -> Concatenation([Reps[1]],y)));
    fi;
    return AllReps;
end;

# Return a list of all possible representations of a certain
# dimension.
RepsOfDim := function(dim,n)
    local x, SymReps, CoxReps;

    if dim = 0 or n=0 then return []; fi;
    CoxReps := RepsMatchingDim(dim,GetMatForAllCoxPartsOf(n));
    SymReps := RepsMatchingDim(dim,GetMatForAllSymPartsOf(n+1));
    return [CoxReps, SymReps];
end;

# Return a list of all characters of S_n in a certain dimension
# obtainable by restricting from S_{n+1}
SymCharactersOfDim := function(dim,n)
    local x, possChars, SymReps;

    SymReps := RepsOfDim(dim,n)[2];
    possChars := [];
    for x in SymReps do
        UniteSet(possChars, [Sum(List(x, i->i.rchar))]);
    od;
    return possChars;
end;

# Enumerate all restrictions to S_n from both W_n and S_{n+1}
```

```
Restrictions := function(i)
    local b, LS, LC, rep, SP, CP, r;
    Print("\n##########################\n\n");
    LC:=[];      LS:=[];
    CP := GetMatForAllCoxPartsOf(i);
    SP := GetMatForAllSymPartsOf(i+1);

    Print("Cox(B_",i,")\n\n");
    for r in [1..Length(CP)] do
        b:=CoxCharacterOfRestrictedRep(CP[r]);
        Add(LC,b);
        Print(CP[r].lambda, CP[r].mu, " restricts to:\n",
              b, "\n\n");
    od;
    Print("--------------------------\n");
    for rep in GetMatForAllSymPartsOf(i) do
        Print(rep.lambda," is in the restrictions of:\n");
        for r in [1..Length(CP)] do
            if Position(LC[r],rep.lambda)<>fail then
                Print(CP[r].lambda,CP[r].mu,"\n");
            fi;
        od;
        Print("\n");
    od;
    Print("\n################\n\n");
    Print("S_",i+1,"\n\n");
    for r in [1..Length(SP)] do
        b:=SymCharacterOfRestrictedRep(SP[r]);
        Add(LS,b);
        Print(SP[r].lambda, " restricts to:\n",
              b, "\n\n");
    od;
    Print("--------------------------\n");
    for rep in GetMatForAllSymPartsOf(i) do
        Print(rep.lambda," is in the restrictions of:\n");
        for r in [1..Length(SP)] do
            if Position(LS[r],rep.lambda)<>fail then
                Print(SP[r].lambda,"\n");
            fi;
        od;
        Print("\n");
    od;
end;
```

References

[1] E. al Aamily, A. O. Morris, and M. H. Peel. The representations of the Weyl groups of type b_n. *J. Algebra*, 68(2):298–305, 1981.

[2] C. T. Benson and L. C. Grove. *Finite reflection groups.* Bogden & Quigley, Inc., Publishers, Tarrytown-on-Hudson, N.Y., 1971.

[3] Marin R. Bridson and Karen Vogtmann. Homomorphisms from automorphism groups of free groups. *Bull. London Math. Soc.*, 35:785–792, 2003.

[4] David S. Dummit and Richard M. Foote. *Abstract algebra.* Prentice Hall Inc., Englewood Cliffs, NJ, 1991.

[5] Joan L. Dyer, Edward Formanek, and Edna K. Grossman. On the linearity of automorphism groups of free groups. *Arch. Math. (Basel)*, 38(5):404–409, 1982.

[6] Edward Formanek and Claudio Procesi. The automorphism group of a free group is not linear. *J. Algebra*, 149(2):494–499, 1992.

[7] William Fulton and Joe Harris. *Representation theory*, volume 129 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1991. A first course, Readings in Mathematics.

[8] Larry C. Grove. *Groups and characters.* Pure and Applied Mathematics (New York). John Wiley & Sons Inc., New York, 1997. A Wiley-Interscience Publication.

[9] Daan Krammer. The braid group B_4 is linear. *Invent. Math.*, 142(3):451–486, 2000.

[10] Wilhelm Magnus, Abraham Karrass, and Donald Solitar. *Combinatorial group theory*. Dover Publications Inc., New York, revised edition, 1976. Presentations of groups in terms of generators and relations.

[11] S. J. Mayer. On the characters of the Weyl group of type C. *J. Algebra*, 33:59–67, 1975.

[12] Jens L. Mennicke. Finite factor groups of the unimodular group. *Ann. of Math. (2)*, 81:31–37, 1965.

[13] John Milnor. *Introduction to algebraic K-theory*. Princeton University Press, Princeton, N.J., 1971. Annals of Mathematics Studies, No. 72.

[14] Bernhard Neumann. Die Automorphismengruppe der freien Gruppen. *Mathematische Annalen*, 107:367–386, 1932.

[15] Jakob Nielsen. Die Isomorphismengruppe der freien Gruppen. *Mathematische Annalen*, 91:169–209, 1924.

[16] D. Ž. Đoković and D. Doniz. The complex indecomposable representations of $\mathrm{Aut}(F_2)$ of degree less than six. *Internat. J. Algebra Comput.*, 10(2):229–260, 2000.