CHARACTER RELATIONS
RESEARCH FOR THE LOUSIANA STATE UNIVERSITY REU 2002

HOLLY SPENCE

Abstract. Fix a finite group G. For a subgroup H of G let $\chi_{G/H}$ be the fixed point character of G acting by left multiplication on G/H. A *character relation* is an equation of the form $\sum n_H \chi_{G/H} = 0$ with $n_H \in \mathbb{Z}$, as H runs over the conjugacy classes of subgroups of G. This paper describes a computer program search using the alebra program Magma to find all character relations for any group G which embeds as a transitive subgroup in the symmetric group, $\mathrm{Sym}(n)$. This program gives results for $n \leq 8$.

1. Introduction

Fix G a finite group. For G-sets S_1, S_2 define $S_1 + S_2$ to be their disjoint union. This is another G-set which we can consider to be a formal "sum." In general, S is a "sum" (union) of orbits (transitive G-sets): $S = S_1 \sqcup S_2 \sqcup \cdots \sqcup S_t$. Each S_i, looks just like G/H_i where $H_i = Stab_G(s_i)$ for a chosen $s_i \in S$. If H_i and H_j are conjugate in S then $S_i = G/H_i$ and $S_j = G/H_j$ are isomorphic G-sets which we consider to be the same G-sets. Thus any S is simply $S = \sqcup n_H G/H$ as H runs over the conjugacy classes of subgroups of G and n_H is a non-negative integer. For example, if $n_H = 2$ then $S = 2G/H$ denotes two copies of a transitive G-set, G/H. If $n_H = 0$ for all G/H then S is the empty G-set. Then the collection of all (isomorphism classes) of finite G-sets forms an additive monoid with respect to $+ = \sqcup$, and the empty set plays the role of the identity, 0. The transitive G-sets G/H generate this collection. Now allow the n_H's to be negative. This means we declare a formal symbol, $-G/H$ to be a formal inverse of G/H. Now that we have the identity, addition and a formal subtraction, the set of formal sums $\sum n_H G/H$, $n_H \in \mathbb{Z}$, form an additive moniod. This

---

Date: July 26, 2002.

The LSU Research Experience for Undergraduates Program is supported by a National Science Foundation grant, DMS-0097530 and a Louisiana Board of Regents Enhancement grant, LEQSF (2002-2004)-ENH-TR-17.

group is called the Burnside ring [1] of G. Let A be a set consisting of one subgroup H for each conjugacy class of subgroups of G. We now have the following definitions.

Definition 1.1. The *Burnside ring*, $B(G)$, is the collection of all formal sums

$$\sum_{H \in A} n_H G/H$$

where $n_H \in \mathbb{Z}$.

The Burnside ring is a free-abelian group of rank r, where r is the number of conjugacy classes of subgroups of G. Moreover, the transitive G-sets G/H for H in A form a $\mathbb{Z}$-basis for $B(G)$.

Definition 1.2. The *fixed point character*, $\chi_{G/H}$, of G on G/H is defined by

$$\chi_{G/H}(g) = |\{\gamma H \mid g\gamma H = \gamma H\}|$$

where $\gamma \in G$. For $S = \sqcup n_H G/H$, where $n_H \in \mathbb{Z}$, define

$$\chi_S(g) = \sum n_H \chi_{G/H}(g).$$

Remark: When all $n_H \geq 0$ then S is a G-set, and $\chi_S(g)$ is actually the number of S fixed by g. But when one or more n_H are negative then S is sums and differences of G-sets and $\chi_S(g)$ is only a sum and difference of the fixed point characters $\chi_{G/H}(g)$, and is not itself a fixed point character.

Definition 1.3. A *character relation* is

$$\sum_{H \in A} n_H \chi_{G/H}(g) = 0$$

for $n_H \in \mathbb{Z}$ $\forall g \in G$. The relation is non-trivial if for at least one $H, n_H \neq 0$.

2. Finding All Character Relations

We want to find all character relations for a given group G. In order to do this we will first find a representative from each conjugacy class of subgroups H in G. After we have obtained the H_i's we can then find the transitive G-sets $G/H_1 \ldots G/H_r$ and compute the fixed point characters $\chi_{G/H_1} \cdots \chi_{G/H_r}$. We want to compute the fixed point characters for all elements $g \in G$, but by the following lemma we only

---

[1] This group also has a multiplication, that we are not concerned with, but it is therefore actually a ring. We will use Burnside ring simply as a name for the additive group.

need to compute the fixed point characters for a representative from each conjugacy class of elements of G.

Lemma 2.1. *$\chi_S(\gamma g \gamma^{-1}) = \chi_S(g)\ \forall \gamma, g \in G$.*

Proof. The element g fixes $s \in S$ iff $\gamma g \gamma^{-1}$ fixes γs. $\square$

By this lemma the character values of conjugate elements will be the same. For example, if we had two conjugate elements g_3 and g_4 then for any H, the fixed point values $\chi_{G/H}(g_3) = \chi_{G/H}(g_4)$. So if we compute $\chi_{G/H}(g_3)$ there is no need to compute $\chi_{G/H}(g_4)$. We can draw the following table.

Conjugacy Classes of Subgroups	Transitive G-sets	Fixed Point Characters	Conjugacy Classes of Group Elements
			$g_1 \dots g_t$
H_1	G/H_1	χ_{G/H_1}	$\chi_{G/H_i}(g_j)$
H_2	G/H_2	χ_{G/H_2}	
$\vdots$	$\vdots$	$\vdots$	
H_s	G/H_s	χ_{G/H_s}	

Table 1. General Fixed Point Character Table

Example 2.2. For example, let G be the symmetric group of degree 3: $\mathfrak{S}_3 = \{e, (12), (13), (23), (123), (132)\}$. The conjugacy classes of elements in $\mathfrak{S}_3$ are represented by the identity, e, (12) representing the transpositions and (123) representing the 3-cycles. We now want a representative from each conjugacy class of subgroups. Note that every proper subgroup of $\mathfrak{S}_3$ is cyclic. So we have $H_1 = e, H_2 =< (12) >$, $H_3 =< (123) >, H_4 = G$. The following table gives the fixed point characters of the elements. We can put this information into the table.

Conjugacy Classes of Subgroups	Transitive G-sets	Fixed Point Characters	Conjugacy Classes of Group Elements		
			e	(12)	(123)
$H_1 = e$	G/H_1	χ_{G/H_1}	6	0	0
$H_2 =< (12) >$	G/H_2	χ_{G/H_2}	3	1	0
$H_3 =< (123) >$	G/H_3	χ_{G/H_3}	2	0	2
$H_4 = G$	G/H_4	χ_{G/H_4}	1	1	1

Table 2. Fixed Point Character Table for $\mathfrak{S}_3$

We can see by looking at this table that

$$\chi_{G/H_1} - 2\chi_{G/H_2} - \chi_{G/H_3} + 2\chi_{G/H_4} = 0$$

This is a character relation as defined previously. The vector of coefficients is $(1, -2, -1, 2)$. Then using left multiplication on the character matrix M we get

$$(1, -2, -1, 2) \begin{pmatrix} 6 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 0 & 2 \\ 1 & 1 & 1 \end{pmatrix} = (0, 0, 0)$$

Therefore, $(1, -2, -1, 2)$ is in the left nullspace of M.

Thus character relations correspond exactly to vectors in the left nullspace of the character matrix. We now want an integer basis for this nullspace.

3. Using Magma to Find All Character Relations for All Groups

Any group G of order n is (isomorphic to) a transitive subgroup of $\mathfrak{S}_n$. So our method of running through all groups will be to find all transitive subgroups of $\mathfrak{S}_n$. It does happen that some groups embed as transitive subgroups of $\mathfrak{S}_n$ for more than one n; our program allows this duplication. The following is a copy of the program written in Magma followed by a commented copy of the same program. The comments are found between /* and */. The command(s) described in each comment can be found directly after the comment that describes it.

```
PROGRAM
SetOutputFile("CharacterRelations");
for x in [1..9] do
A:=Sym(x);
printf "SYMMETRIC GROUP OF DEGREE %o", x;
print "";
print "_________________________________________________________";
B:=Subgroups(A);
for i in [1..#B] do
if IsTransitive(B[i]`subgroup) then
G:=B[i]`subgroup;
print "";
print "";
M:=[];
Sequence:=[];
printf "Group G=B[%o], Order G=%o", i, B[i]`order;
print "";
```

```
print"************************";
S:=Subgroups(G);
for j in [1..#S] do
H:=S[j]‘subgroup;
print "";
printf "H=S[%o], Order H=%o", j, S[j]‘order;
print "";
printf "H is %o", S[j]‘subgroup;
print "";
PChar:=PermutationCharacter(G,H);
for k in [1..#PChar] do
Append(~M, #PChar);
Append(~Sequence, PChar[k]);
end for;
end for;
n:=#PChar;
m:=#M div n;
my_matrix:=Matrix(m, n, Sequence);
print "";
print "THE PERMUTATION CHARACTER MATRIX IS";
my_matrix;
print "";
print "THE KERNEL OF THE MATRIX";
Kernel(my_matrix);
end if;
end for;
end for;
UnsetOutputFile();

COMMENTED PROGRAM
        /* Start redirecting all Magma output to a file
           (specified by "CharacterRelations") because
           the terminal  window is too small to hold all
           of the information. */

SetOutputFile("CharacterRelations");

        /* Create a loop to run through the symmetric
           groups from degree 1 to 9. */

for x in [1..9] do
```

```
        /* Declare A to be the symmetric group of degree x.
           printf is the command used to print both information
           we typed and information the program will give us.
           %o, for object, is typed in the places where the
           computer is going to compute the object that belongs
           in the blank. For each %o we must tell the computer
           what object is to be printed in that space.  In this
           case the value of x will be printed in the space.   */

A:=Sym(x);

        /* Output which symmetric group the program is in. */

printf "SYMMETRIC GROUP OF DEGREE %o", x;
print "";
print "_________________________________________________________________";

        /* Find representatives for the conjugacy classes of
           subgroups for the group  A.  Call this list B.
           The subgroups are returned as a sequence of records
           where the i-th record contains: A representative
           subgroup H for the i-th conjugacy class (field name
           subgroup), the order of the subgroup (field name order),
           the number of subgroups in the class (field name length),
           and (optionally) a presentation for H (field name
           presentation). */

B:=Subgroups(A);

        /* Run through B, which is just a list of records, and
           pull out each subgroup, then check to see if it is
           transitive.  If it is transitive then call it group G.
           If it is not transitive, then completely disregard it.
           In order to pull a subgroup out of a record we must
           know the record name, r, and the field name, fieldname,
           of what we want.  We then use the command r`fieldname.
           In this case the record name is B and the field name is
           subgroup. */

for i in [1..#B] do
if IsTransitive(B[i]`subgroup) then
G:=B[i]`subgroup;
```

```
print "";
print "";

        /* Declare M to be an empty list that will later be
           appended to.  The data that will be appended to this
           list is the number of conjugacy classes of elements
           of G.  This list will later be used to find the number
           of rows, m, for the matrix of permutation characters.
           We reset the list M to the empty list at this point
           because the size of the permutation character matrix
           is different for each group.  */

M:=[];

        /* Declare Sequence to be an empty list that will later
           be appended to.  The data that will be appended to
           this list is the list of permutation characters.  This
           list is also reset as the empty list for each group. */

Sequence:=[];

        /* The conjugacy classes of subgroups of A, the
           symmetric group of degree x, are listed as B[i]'s.
           At this point we find out which i we are looking
           at and what the order of the group is. To obtain
           the order of B[i] we must use the method of calling
           information out of a record once again.  This time
           the record name is the same, and the field name is
           order. We also use the printf command again.  This
           time the value of i will go in the first space and
           the value of B[i]`order (the order of B[i]) will go
           in the second space.  */

printf "Group G=B[%o], Order G=%o", i, B[i]`order;
print "";
print"************************";

        /* Declare the conjugacy classes of subgroups
           of G to be S. */

S:=Subgroups(G);
```

```
        /* Run through the record S and pull the
           subgroups out of the record.  Call each
           subgroup H. */

for j in [1..#S] do
H:=S[j]‘subgroup;
print "";

        /* Output which S[j] we are looking at and
           the order of H. */

printf "H=S[%o], Order H=%o", j, S[j]‘order;
print "";

        /* Output more information about the subgroup H. */

printf "H is %o", S[j]‘subgroup;
print "";

        /* Declare PChar to be the character of G afforded
           by the permutation representation of G given by
           the action of G on the right cosets of H in G. */

PChar:=PermutationCharacter(G,H);

        /* Run through PChar and append to lists M and
           Sequence.  By the end of the loop, M will
           contain a list of the number of conjugacy classes
           of elements in G, #PChar, and M will have mn number
           of copies of #PChar.  By the end of the loop,
           Sequence will contain all of the permutation
           characters, PChar[k] that will be put into the m
           by n matrix. The ˜ simply tells MAGMA to do more
           than just add to the list M, but also to continue
           to call the appended to list, M. */

for k in [1..#PChar] do
Append(˜M, #PChar);
Append(˜Sequence, PChar[k]);

        /* End the loop that creates a list of all permutation
           characters for the group, and creates the list M. */
```

```
end for;

        /* End the loop that gathers all subgroups, H, and
           computes the permutation characters. */

end for;

        /* Declare n to be the number of columns of the
           permutation character matrix.  This number is the
           same as the number of conjugacy classes of elements
           of G. */

n:=#PChar;

        /* Declare m to be the number of rows of the
           permutation character matrix. This number
           is the same as the length of list M divided
           by n. */

m:=#M div n;

        /* Create an m x n matrix, my_matrix, that contains
           the list Sequence, the permutation characters.
           Given integers m, n >= 0 and a sequence Q of length
           mn containing elements of a ring R, the command
           Matrix(m, n, Q) returns the m x n matrix over R
           whose entries are the entries of Q, in row-major
           order.*/

my_matrix:=Matrix(m, n, Sequence);
print "";
print "THE PERMUTATION CHARACTER MATRIX IS";

        /* Output the permutation character matrix. */

my_matrix;
print "";
print "THE KERNEL OF THE MATRIX";

        /* Output an integral basis of the kernel of
           my_matrix. The kernel is the left nullspace
```

```
           of my_matrix.*/

Kernel(my_matrix);

        /* End the loop that finds G, the transitive
           subgroups of A. */

end if;

        /* End the loop that runs through the conjugacy
           classes of subgroups of A. */

end for;

        /* End the loop that runs through the symmetric
           groups from degree 1 to 9. */

end for;

        /* Stop redirecting all Magma output to a file. */
UnsetOutputFile();
```

4. Results

(1) The final output for this program can be found at http://www.math.lsu.edu/reu2002.

(2) For this procedure Magma can only run through $n = 1$ to 8. We get a partial list for $\mathfrak{S}_9$.

(3) Although n runs only up to 8 we can still obtain groups G in $\mathfrak{S}_8$, with subgroups $H \in G$ of large index in G. The largest we obtain when only considering $n = 1$ to 8 is $G = \mathfrak{S}_8$ and $H = e$; the index is 8!=40,320. If we consider the partial list of groups found when looking for transitive subgroups in $\mathfrak{S}_9$, we can obtain $G = \mathfrak{S}_9$ and $H = e$; the index is 9!=362,880, but the program stops before we are give a character matrix for this group. The group of highest order that we are given a complete character matrix for has order 181,440, which is 9!/2. So the group is the alternating group $\mathfrak{A}_9$, and the highest index is 181,440 where $H = e$. The fact that we can only run this procedure through $\mathfrak{S}_8$ simply means that we can only say we have a complete list of character relations for all transitive groups of degree eight or less.

(4) The first character relation ocurs in $\mathfrak{S}_3$. Note that $\mathfrak{S}_1$ and $\mathfrak{S}_2$ are cyclic. In fact, when G is cyclic the only character relation is the trivial one.

5. Interpreting Character Relations

Let $\mathbb{Q}$ be the field of rational numbers. Let V be a finite dimensional vector space on which G acts by linear transformation. If W is another such space, then V and W can be added to give $V \bigoplus W$. Thus the set of all (isomorphism classes of) finite dimensional $\mathbb{Q}$ -vector spaces in which G acts is an additive monoid. (The additive identity is the vector space 0). For each element V in this monoid, let $-V$ denote a formal inverse. We agree that $V_1 - W_1 = V_2 - W_2$ if and only if the G-space $V_1 \bigoplus W_2 \cong V_2 \bigoplus W_1$.

Definition 5.1. $R_{\mathbb{Q}}(G)$= the ring [2] of rational linear representations of G is the collection of all formal differences $V - W$.

By a famous theorem of E. Artin, the rank t of $R_{\mathbb{Q}}(G)$is the number of conjugacy classes of cyclic subgroups of G. A linear representation of G on V is a map $\rho : G \to Aut(V)$. There are two main facts about $R_{\mathbb{Q}}(G)$:

(1) Two elements $V_1 - W_1$ and $V_2 - W_2$ in $R_{\mathbb{Q}}(G)$ are equal if and only if $\forall g \in G$ trace$(\rho_{V_1}(g))-$trace$(\rho_{W_1}(g))$ =trace$(\rho_{V_2}(g))-$trace$(\rho_{W_2}(g))$. Their linear characters agree.

(2) Let $G/H = \{\gamma_1 H, \dots, \gamma_k H\}$. For each coset $\gamma_i H$ we choose a symbol $\overline{\gamma_i}$ and let $U_{G/H} = \mathbb{Q}\overline{\gamma_1} + \cdots + \mathbb{Q}\overline{\gamma_k}$ on which G acts by sending $\overline{\gamma_i}$ to $\overline{\gamma_j}$ where $g\gamma_i H = \gamma_j H$. In other words, G permutes a basis of $U_{G/H}$ in exactly the same way as G permutes the cosets in G/H. This action on a basis extends to an action of G on U. In this way each transitive G-set G/H gives rise to a linear representation of G on U.

There exists a ring homomorphism, $\Phi : B(G) \to R_{\mathbb{Q}}(G)$ induced by sending the transitive G-set G/H to the G-vector space $U_{G/H}$ defined above. The permutation character of G/H in $B(G)$ will be the same as the linear character of $U_{G/H}$ $R_{\mathbb{Q}}(G)$. Similarly, the difference of two permutation characters equals the difference of the two corresponding linear characters. Thus, a non-trivial character relation comes from a non-zero element in $B(G)$ whose image in $R_{\mathbb{Q}}(G)$has linear character equal to zero. This means that the non-zero element in $B(G)$ maps to

---

[2] $R_{\mathbb{Q}}(G)$is actually a ring, but we will not be concered with the ring structure. We will simply use the representation ring as a name for the additive group.

0 in $R_{\mathbb{Q}}(G)$; thus we have found a nontrivial element in the kernel of Φ.

6. Acknowledgments

The author thanks Dr. Robert Perlis for his guidance and help on this project and writing this paper. The author also thanks Dr. Neal Stoltzfus for his guidance and help writing the program for this project.

Mathematics Department, Eastern Illiniois University, Charleston, Illinois 61920
E-mail address: `hespence@eiu.edu`