MATHEMATICS OF COMPUTATION
VOLUME 62, NUMBER 205
JANUARY 1994, PAGES 363–383

AN INDUCTIVE SCHEMA FOR COMPUTING CONJUGACY CLASSES IN PERMUTATION GROUPS

GREG BUTLER

ABSTRACT. An approach to computing the conjugacy classes of elements of large permutation groups is presented in detail, and a prototype implementation is described. The approach builds on recent efficient algorithms for computing conjugacy classes of p-groups, and for computing Sylow subgroups of large permutation groups. Classes of elements of composite order are determined by recursively analyzing quotients of centralizers of p-elements.

1. INTRODUCTION

The conjugacy classes of elements are an important piece of information about the structure of a group. As such, it is a prerequisite for many algorithms such as those for computing the lattice of subgroups, the automorphism group, and the character table.

For large permutation groups, it has been feasible to attempt the computation of the conjugacy classes ever since the development [1, 20] of algorithms to compute centralizers and to test conjugacy of two elements. However, to date, there has been no systematic way of finding the class representatives. The state-of-the-art has been to consider randomly chosen elements, often several thousand such elements, and often with failure to locate a representative of every class.

The systematic approach presented here was first suggested to us by Sims in 1976 in discussions on this problem. The idea is implicit in many papers on the classification of finite simple groups. Given a group G, we consider each prime p dividing the order of G in turn. The classes of elements of order p^r, for all possible values of r, are determined by computing a Sylow p-subgroup, analyzing its classes, and then determining their fusion in G. The classes of elements of composite order $p^r t$, where t is coprime to p, are determined by computing the centralizer $C_G(z)$, for each class representative z of order p^r, and analyzing the classes of the centralizer, or the classes of the centralizer modulo a normal p-subgroup such as $\langle z \rangle$. For permutation groups, there is a

Received by the editor February 15, 1991 and, in revised form, August 28, 1991 and August 3, 1992.

1991 *Mathematics Subject Classification.* Primary 68Q40, 20–04, 20B99.

Key words and phrases. Conjugacy class, permutation group, algorithm.

This work was supported in part by the Australian Research Council, and benefitted from the hospitality of Lehrstuhl II für Mathematik (Informatik), Universität Bayreuth.

0025-5718/94 $1.00 + $.25 per page

natural quotient of the centralizer given by its action on the cycles of z. Again, the fusion of these classes in G must be determined.

The major contributions of this paper are

- a new theoretical result, Theorem 5.1.1, which eliminates conjugacy testing when determining the classes of elements of composite order;
- a strategy to process the primes in an appropriate order and restrict the composite orders to those involving only certain primes;
- a framework for the overall computation so that the existing theory and ad hoc hand methods can be applied by a computer;
- a prototype implementation in Cayley, v3.7 [8] to demonstrate the feasibility of the computer application of the inductive schema, and to identify the computational bottlenecks.

A study of the complexity of the problem is not included. It is difficult to see how the computation of the conjugacy classes could avoid the computation of centralizers, or testing whether two elements are conjugate. The complexity of these simpler problems is still open [13].

The implementation of the approach has been made possible by recent progress [7] in computing Sylow subgroups of large permutation groups; in converting the representation of a p-group from a permutation group to a pc presentation [4, 6, 14]; and in computing the conjugacy classes of elements of a p-group given by a pc presentation [9].

The prototype implementation computes the 116 classes of PSL(5, 3) of degree 121 and order 237,783,237,120 = $2^9 3^{10} 5\ 11^2 13$ in 14 minutes on a Sun Sparcstation, and computes the 60 classes of Conway's second sporadic simple group Co2 of degree 2300 and order 42,305,421,312,000 = $2^{18} 3^6 5^3 7\ 11\ 23$ in 2.5 hours.

The paper continues by presenting the necessary background on conjugacy classes and on computational group theory, and then describing the inductive schema in overview. The following sections then discuss the theory, algorithms, and present examples, for each of the major subproblems. The prototype implementation is described and some experimental results presented. Opportunities for further work are discussed before concluding the paper.

2. Background

This section presents the necessary background notation, definitions, and computational tools which we will need. The reader is referred to [12, 22] for elementary definitions and results from group theory. The engineering aspects of developing and implementing the algorithm require some appreciation of what can be done cheaply, or not so cheaply, using the current state-of-the-art algorithms for subtasks.

Let G be a finite group. Let g be an element of G. Let $K_G(g)$ denote the *conjugacy class* $\{h^{-1}gh \mid h \in G\}$ of g in G, and define $Kr_G(g) = \bigcup_{\langle z\rangle=\langle g\rangle} K_G(z)$ to be the *rational class* of g in G. The rational class is a disjoint union of conjugacy classes $K_G(g^{m_i})$, for certain integers $m_0=1, m_1, \ldots, m_s$ between 1 and the order of g. The length of each conjugacy class is determined by the centralizer $C_G(g)$, and the integers m_i are determined by the structure of the abelian group $N_G(\langle g\rangle)/C_G(g)$.

Define the *Galois group*, $\mathrm{Gal}_G(g)$, of $g \in G$ to be the subgroup of $\mathrm{Aut}(\langle g\rangle)$ isomorphic to $N_G(\langle g\rangle)/C_G(g)$. The classes within the rational class $Kr_G(g)$ are in 1-1 correspondence with the cosets of the Galois group, $\mathrm{Gal}_G(g)$, in the automorphism group, $\mathrm{Aut}(\langle g\rangle)$.

Consider the structure of the automorphism group of a cyclic group Z_n of order n.

Proposition 2.0.1 ([16, Satz 13.19, page 84]). (1) $\mathrm{Aut}(Z_n)$ *is isomorphic to the multiplicative group of* $\mathbb{Z}/n\mathbb{Z}$.

(2) *If* $n = n_1 n_2 \cdots n_l$ *with the* n_i *pairwise coprime, then*

$$\mathrm{Aut}(Z_n) \simeq \mathrm{Aut}(Z_{n_1}) \times \mathrm{Aut}(Z_{n_2}) \times \cdots \times \mathrm{Aut}(Z_{n_l}).$$

(3) *For* $p > 2$, $\mathrm{Aut}(Z_{p^r})$ *is a cyclic group of order* $p^{r-1}(p-1)$.

(4) *For* $r \geq 3$,

$$\mathrm{Aut}(Z_{2^r}) \simeq Z_{2^{r-2}} \times Z_2$$

and we can take $5 \bmod 2^r$ *and* $-1 \bmod 2^r$ *as generators of the respective direct factors.*

(5) $\mathrm{Aut}(Z_4) \simeq Z_2$, *and* $\mathrm{Aut}(Z_2)$ *is the trivial group.* □

Let ε_g denote the obvious isomorphism between $\mathrm{Aut}(\langle g\rangle)$ and the multiplicative group of $\mathbb{Z}/n\mathbb{Z}$, where $n = |g|$. Hence, $\varepsilon_g(\mathrm{Gal}_G(g))$ is a subgroup of the multiplicative group of $\mathbb{Z}/n\mathbb{Z}$. If $x \in N_G(\langle g\rangle)$ or x is a coset in $N_G(\langle g\rangle)/C_G(g)$, then $\varepsilon_g(x)$ will denote the corresponding element of $\varepsilon_g(\mathrm{Gal}_G(g))$. If $n = qt$, where q and t are coprime, then $\mathrm{Aut}(Z_n) \simeq \mathrm{Aut}(Z_q) \times \mathrm{Aut}(Z_t)$. Hence, there is a projection $\downarrow_t^n$ from the multiplicative group of $\mathbb{Z}/n\mathbb{Z}$ to the multiplicative group of $\mathbb{Z}/t\mathbb{Z}$. We denote the inverse embedding by $\uparrow_t^n$.

Corollary 2.0.1. *Let* S *be a Sylow* p*-subgroup of* G *and let* $g \in G$ *have order* p^r.

(1) *For* $p > 2$ *and* $r > 1$, *regard* $\mathrm{Aut}(Z_{p^r})$ *as* $Z_{p^{r-1}} \times Z_{(p-1)}$. *There exists* $g_1 \in S \cap K_G(g)$ *such that* $\mathrm{Gal}_S(g_1)$ *is the* $Z_{p^{r-1}}$*-component of* $\mathrm{Gal}_G(g_1)$.

(2) *For* $p > 2$ *and* $r > 1$, *the* $Z_{(p-1)}$*-component of* $\mathrm{Gal}_G(g)$ *projects faithfully to the* $Z_{(p-1)}$*-component of* $\mathrm{Gal}_G(g^p)$.

(3) *For* $p = 2$, *there exists* $g_1 \in S \cap K_G(g)$ *such that* $\mathrm{Gal}_S(g_1) = \mathrm{Gal}_G(g_1)$. □

For cases (1) and (3) of the corollary, we may take $g_1 \in S \cap K_G(g)$ such that $C_S(g_1)$ is a Sylow p-subgroup of $C_G(g_1)$ and the normalizer $N_S(\langle g_1\rangle)$ is as large as possible. In case (2) of the corollary, if $C_G(g) = C_G(g^p)$, then take an element $x \in G$ which normalizes g^p and (modulo the centralizer) generates the $Z_{(p-1)}$-component of $\mathrm{Gal}_G(g^p)$. The generator of the $Z_{(p-1)}$-component of $\mathrm{Gal}_G(g)$ is a power of x.

Proposition 2.0.2. *If* $H \leq G$ *and* $h \in H$, *then* $\mathrm{Gal}_H(h) \leq \mathrm{Gal}_G(h)$. □

Corollary 2.0.2. *Let* $H \leq G$, $h \in H$ *and* $g \in G$ *such that* $|h| = |g| = n$. *If* $\varepsilon_h(\mathrm{Gal}_H(h)) \not\leq \varepsilon_g(\mathrm{Gal}_G(g))$, *then* h *is not conjugate in* G *to* g. □

Hence, some properties of the Galois groups, such as their order, may allow one to deduce that the elements are not conjugate, and therefore avoid a conjugacy test.

The inductive schema as implemented determines the rational classes of a permutation group G. The information it stores about each rational class includes a representative element g; the list of powers $1, m_2, \ldots, m_s$ such that g^{m_i} are the representatives of the classes within the rational class; the Galois group $\mathrm{Gal}_G(g)$; and appropriate elements of $N_G(\langle g \rangle)$, for the case when g is a p-element. Hence, the schema also determines the conjugacy classes of G.

For computational purposes, a permutation group G acting on the set of points Ω is described by a *base* and *strong generating set*. A base for G is a sequence of points $B = [\beta_1, \beta_2, \ldots, \beta_k]$ in Ω such that the only element in G fixing every point β_i is the identity element. A strong generating set relative to B is a set T of elements of G which contains a set of generators for each stabilizer $G_{\beta_1, \beta_2, \ldots, \beta_{i-1}}$, $1 \leq i \leq k$. For more details see [18].

For computational purposes, a p-group is described by a *power-commutator presentation*, or pc presentation. This is a presentation of the form

$$H = \langle a_1, a_2, \ldots, a_n | a_i^p = u_i, \text{ for } 1 \leq i \leq n, [a_j, a_i] = v_{ij}, \text{ for } 1 \leq i < j \leq n \rangle,$$

where the u_i are words in $\{a_{i+1}, a_{i+2}, \ldots, a_n\}$, and the v_{ij} are words in $\{a_{j+1}, a_{j+2}, \ldots, a_n\}$.

Let us begin by describing some of the more expensive computations which might arise as subproblems in determining the conjugacy classes.

P1: Given a base and strong generating set for a group G, and an element $z \in G$, compute a base and strong generating set for $C_G(z)$.

P2: Given a base and strong generating set for a group G, and two elements $g_1, g_2 \in G$, determine whether g_1 is conjugate to g_2 in G, and if so, determine a conjugating element.

Both of the computations **P1** and **P2** are generally quite efficient. However, the algorithms [1] are backtrack searches and are subject to combinatorial explosion of the size of the search tree. So in bad cases, the cost could be two or three orders of magnitude worse than the average cost. Furthermore, if the determination of the conjugacy classes requires thousands of conjugacy tests, then the cost accumulates, and the chance of a bad case increases. Hence, a major aim of any approach to determining the conjugacy classes should be to minimize the number of conjugacy tests in the permutation group G.

P3: Given a base and strong generating set for a group G, and a prime p dividing the order of G, compute a base and strong generating set for a Sylow p-subgroup of G.

The algorithm [7] for **P3** requires a small number of centralizer computations, and its total cost is essentially the cost of these computations. Hence, it may suffer from a bad case for the centralizer algorithm. On the other hand, the determination of the conjugacy classes requires only one Sylow subgroup computation for each prime.

The following computations can be done very efficiently.

P4 [3, 4, 5, 6]: Given a base and strong generating set for a group G acting on Ω, set up any of the following homomorphisms: the action of G on an invariant subset Δ of Ω; the action of G on an invariant partition Υ of Ω; for a p-group or soluble group G, the isomorphism between G and the group defined by a pc presentation of G. Allow the computation of the image, kernel, image of an element, preimage of an element, image of a subgroup, and preimage of a subgroup.

P5 [9, 17]: For a p-group G defined by a pc presentation, compute any of the following: conjugacy classes of elements; centralizer of an element; determine if two elements are conjugate, and if so, determine a conjugating element; normalizers, centre.

The following properties can be used to decide that two elements are not conjugate. Hence an explicit conjugacy test in a permutation group is only performed when all the conditions (for which the information is readily available) have been checked.

(1) If g_1, g_2 have distinct cycle structures, then they are not conjugate in G.
(2) If, for some integer t, the powers g_1^t, g_2^t are not conjugate in G, then g_1, g_2 are not conjugate in G.
(3) If $g_1 \in H$ and $g_2 \in G$, where $H \leq G$, such that the order of the centralizer $C_H(g_1)$ does not divide the order of $C_G(g_2)$, then g_1, g_2 are not conjugate in G.
(4) If $g_1 \in H$ and $g_2 \in G$, where $H \leq G$, such that $\mathrm{Gal}_H(g_1)$ is not isomorphic to a subgroup of $\mathrm{Gal}_G(g_2)$, then g_1, g_2 are not conjugate in G.

3. Overview

The main ideas behind the approach are

- For a given prime p, a representative of each rational class of p-elements can be found in a fixed Sylow p-subgroup S of G.
- Representatives of the rational classes of elements of order $p^r t$, where t is coprime to p, can be found in the centralizers of the rational class representatives of elements of order p^r.
- The rational class representatives of elements of order $p^r t$ can be found by taking suitable preimages of the rational class representatives of elements of order t in the quotient of the centralizer given by its action on the cycles of the element of order p^r.

Rational classes are determined, as this reduces the effort in computing centralizers and analyzing the rational classes of their quotients. The rational classes of a Sylow subgroup S help to reduce the number of conjugacy tests in G, when determining the fusion in G of the S-rational classes. The Galois group in S helps determine the Galois group in G for p-elements, and reduce the number of conjugacy tests.

The approach can treat the primes p dividing the order of G in terms of "increasing difficulty", so that, for the last prime, it is not necessary to analyze the centralizers of p-elements to find their roots. The inductive schema is outlined in Algorithm 1.

Algorithm 1

Input : a finite permutation group G with a base and strong generating set;
Output : the rational classes of G;
begin
let $|G| = p_1^{n_1} p_2^{n_2} \cdots p_d^{n_d}$;
for each prime p_i **do**
"determine the G-rational classes of p_i-elements"
$S :=$ Sylow p_i-subgroup of G;
let $f_1 : S \to \overline{S}$, where $\overline{S}$ is defined by a pc presentation of S;
compute the rational classes of $\overline{S}$ (and hence, of S);
determine the fusion of the S-rational classes in G;
for each rational class $Kr_G(z)$ of p_i-elements **do**
"determine the G-rational classes of roots of z of order $t p_i^r$, where $|z| = p_i^r$ and t only involves the primes $p_{i+1}, p_{i+2}, \ldots, p_d$"
$C := C_G(z)$;
let $f_2 : C \to \overline{C}$, the action of C on the cycles of z;
determine the $\overline{C}$-rational classes of elements whose order only involves the primes $p_{i+1}, p_{i+2}, \ldots, p_d$;
lift the representatives of $\overline{C}$-rational classes to roots of z and determine their conjugacy in G;
end;
end;
end.

As an example of the inductive approach, consider the group PSL(4, 2) of all nonsingular 4×4 matrices over GF(2). This group has a permutation representation of degree 15. Table 1 lists the information about its rational classes. The notations 7AB and 15AB indicate that the rational class contains two classes. All other rational classes contain a single conjugacy class. The

TABLE 1. Rational classes of PSL(4,2)

Rational Class	$\lvert C_G(g)\rvert$	Length	Cycles	Fusion
1A	$2^6 3^2 5\ 7$	1	1^{15}	
2A	$2^6 3$	105	$2^4 1^7$	
2B	$2^5 3$	210	$2^6 1^3$	
3A	$2^2 3^2 5$	112	3^5	$1 \sim 2$
3B	$2\ 3^2$	1120	$3^4 1^3$	$1 \sim 2$
4A	2^4	1260	$4^2 2^2 1^3$	$1 \sim 3$
4B	2^3	2520	$4^3 2^1 1^1$	$1 \sim 3$
5A	$3\ 5$	1344	5^3	$1 \sim 2 \sim 3 \sim 4$
6A	$2^2 3$	1680	$6^2 3^1$	$1 \sim 5$
6B	$2\ 3$	3360	$6^1 3^2 2^1 1^1$	$1 \sim 5$
7AB	7	2880	$7^2 1^1$	$1 \sim 2 \sim 4,\ 3 \sim 5 \sim 6$
15AB	$3\ 5$	1344	15^1	$1 \sim 2 \sim 4 \sim 8,\ 7 \sim 11 \sim 13 \sim 14$

column headed *Length* gives the length of each conjugacy class. The column headed *Fusion* indicates which powers of the representative are conjugate in G.

In Table 2 is a commentary on the execution of Algorithm 1 for this group. The word "class" refers to a rational class. The primes are processed in order from largest to smallest. The routine *pelts* determines the rational classes of p-elements in G; the routine *roots* determines the rational classes of elements of composite order which are roots of a p-element, but only for orders involving just the smaller primes. The commentary comes from a prototype implementation in Cayley, v3.7, so a group order such as $2^2 3^2 5$ is printed as SEQ(2, 2, 3, 2, 5, 1). The times refer to a Sun Sparcstation, and are rounded to the nearest second.

There are several major subproblems to be solved. Problems 2 and 3 have technical solutions, but Problems 1 and 4 require a strategy to be developed.

TABLE 2. Execution of Algorithm 1 for PSL(4, 2)

```
Sylow 7 subgroup took 0 seconds
pelts took 1 second, did 2 conjugacy tests, fused 1 S-class into 1 G-class
   found class 7AB


Sylow 5 subgroup took 0 seconds
pelts took 1 second, did 1 conjugacy test, fused 1 S-class into 1 G-class
   found class 5A


in roots, the centralizer has order SEQ( 3, 1, 5, 1 )
restriction to cycles: order SEQ( 3, 1 ) degree 3
      in quotient found image of class 15AB
      recursive treatment of quotient took 0 seconds
      in roots, the quotient has 1 class
roots took 1 second


Sylow 3 subgroup took 0 seconds
pelts took 2 seconds, did 4 conjugacy tests, fused 4 S-classes into 2 G-classes
   found class 3B; found class 3A


in roots, the centralizer has order SEQ( 2, 1, 3, 2 )
restriction to cycles: order SEQ( 2, 1, 3, 1 ) degree 7
      in quotient found image of class 6B
      recursive treatment of quotient took 0 seconds
      in roots, the quotient has 1 class
roots took 1 second


in roots, the centralizer has order SEQ( 2, 2, 3, 2, 5, 1 )
restriction to cycles: order SEQ( 2, 2, 3, 1, 5, 1 ) degree 5
      in quotient found image of class 6A
      recursive treatment of quotient took 1 second
      in roots, the quotient has 1 class
roots took 1 second


Sylow 2 subgroup took 0 seconds
pelts took 5 seconds, did 0 conjugacy tests, fused 15 S-classes into 4 G-classes
   found class 2A; found class 2B; found class 4A; found class 4B


Total time 17 seconds
```

Problem 1: Rational classes of p-elements. Determine the rational classes in G of elements of prime power order.

Problem 2: Classes within a rational class. Determine the classes within each rational class $Kr_G(g)$. That is, the powers $m_1 = 1, m_2, \ldots, m_s$ such that $Kr_G(g)$ is the disjoint union of the $K_G(g^{m_i})$.

This problem is treated in two parts (a) and (b).

Problem (2a): For a rational class of p-elements with representative g, determine the Galois group $\mathrm{Gal}_G(g)$.

Problem 3: Classes of elements of composite order. Given an element z of prime-power order p^r, determine the rational classes and classes of elements y where $y^t \in Kr_G(z)$, and t is coprime to p.

This includes

Problem (2b): For a rational class of elements of composite order with representative g, determine the Galois group $\mathrm{Gal}_G(g)$.

Problem 4: Ordering the primes. Determine the order in which the primes p_i dividing $|G|$ should be processed.

The following sections treat the p-elements (that is, Problems 1 and 2a), the elements of composite order (that is, Problems 3 and 2b), and the ordering of the primes.

4. Elements of prime-power order

This section looks at finding the representatives of the rational classes of elements of order p^r. Determining the classes within a rational class of such elements is done by determining the Galois group of the element. The problems dealt with here are

Problem. Determine the rational classes $Kr_G(g)$, where the order of g is a power of p.

Problem. Determine the classes within each rational class $Kr_G(g)$, where the order of g is a power of p, by determining $\mathrm{Gal}_G(g)$.

The best we have been able to achieve for the first problem is a framework and strategy for finding the rational class representatives. There are examples where our solution takes a long time, so the problem may be regarded as still open. We do arrange that this strategy determines the p-part of the Galois group as a by-product.

The theory for the second problem is well developed, so what is required is a careful organization of the theory into an algorithm.

4.1. Theory.

Proposition 4.1.1 [16, Hilfssatz 2.5, page 418]. *Let S be a Sylow subgroup of G. Suppose K and L are subsets of S such that $K^s = K$ and $L^s = L$ for*

all $s \in S$, *and suppose that* $g \in G$ *conjugates* K *to* L. *Then there exists* $h \in N_G(S)$ *such that* $K^h = L$. □

Corollary 4.1.1. *Let* p *be a prime dividing the order of* G. *Let* S *be a Sylow* p*-subgroup of* G.

(1) *If* S *is abelian, then fusion of the elements of* S *in* G *is completely determined by fusion in* $N_G(S)$.

(2) *Fusion amongst elements of* $Z(S)$ *in* G *is completely determined by* $N_G(S)$. □

Proposition 4.1.2. *Let* $g \in G$ *be a* p*-element and let* S *be a fixed Sylow* p*-subgroup of* G. *Then there exists* $g_1 \in S \cap K_G(g)$ *such that* $C_S(g_1)$ *is conjugate to a Sylow* p*-subgroup of* $C_G(g)$. □

Hence, the roots of g_1 in S will contain representatives of all the G-classes of roots of g, where the root is a p-element.

4.2. **Algorithm for Problem 1.** The rational classes of S are actually determined in the group $\overline{S}$ described by a pc presentation. The $\overline{S}$-classes are computed, and then conjugacy tests in $\overline{S}$ are performed to determine in which class a power of a class representative $\overline{g}$ lies, thus calculating the rational class and the abelian group $\mathrm{Gal}_{\overline{S}}(\overline{g})$. The results are lifted back to S.

A straightforward approach to determine the G-rational classes of p-elements is to apply algorithms which test conjugacy in G of each S-rational class representative with each G-class representative found so far. However, these tests are too expensive, and the number of S-rational classes can be large (for example, several thousand rational classes for a group of order p^{15}).

An approach which does more analysis of the Sylow subgroup and various normalizers might be organized as in Algorithm 2. The analysis produces an order in which to treat the S-rational classes. This order is given in *list*. It also produces a corresponding sequence of sets of rational classes, called *away*, such that if the *list*[i]th S-rational class fuses to an S-rational class earlier in the order, then every member of *away*[i] also fuses to an earlier member of *list*.

We take *away*[i] to be the set of the S-rational classes containing roots of the *list*[i]th rational class. If the representative z of the *list*[i]th S-rational class is conjugate in G to an earlier S-rational class, with representative z', then every root of z is conjugate to a root of z'. We order *list* so that the centralizer in S of the earliest such z' does contain a representative of each root of z' that is a p-element, and we list the roots of z' before z and its roots.

Our current analysis categorizes the S-rational classes by the order of the elements, and their cycle structure (as elements of G). Within each category, the analysis sorts the S-rational classes in decreasing order of their centralizer order (so that the roots of a discarded S-rational class can be safely discarded), and within a centralizer order it sorts the S-rational classes in decreasing order of their normalizer order (so that the p-part of the Galois group in G is known). The first S-rational class with each distinct cycle structure is placed at the start of *list*. The remaining S-rational classes are then listed. The S-rational classes of elements of order p are listed using the sorted order. After a S-rational class

of elements of order p—with representative z—and before the next S-rational class of elements of order p, we list the roots of z in a recursive fashion.

Algorithm 2

Input : a permutation group G;
a prime p dividing the order of G;
a Sylow p-subgroup S of G;
maybe a *bound* on the total number of elements in the classes;
Output : the G-rational classes, KRp, of p-elements;
begin
determine the rational classes of S;
perform an analysis and determine *list*, *away*;
KRp := empty;
nogood := nullset;
for i := 1 to length(*list*) **do**
if *list*[i] is not in *nogood* **then**
let g be the representative of the *list*[i]th rational class of S;
if g is not in any class in KRp **then**
add $Kr_G(g)$ to KRp;
if number of elements in $KRp \geq$ *bound* **then**
return;
end
else
nogood := *nogood* $\cup$ *away*[i];
end;
end;
end;
end.

4.3. **Example.** Consider the group PSp(4,7) and prime p=2. The Sylow subgroup S has order 2^8 and pc presentation

$$\langle a_1, a_2, \ldots, a_8 \mid a_1^2 = a_4^2 = a_5^2 = a_7^2 = a_8^2 = 1,\ a_2^2 = a_3^2 = a_6^2 = a_8,$$
$$[a_2, a_1] = a_4,\ [a_3, a_1] = a_5,\ [a_3, a_2] = [a_5, a_2] = a_6,$$
$$[a_4, a_3] = a_6 a_8,\ [a_5, a_4] = a_7 a_8,\ [a_6, a_1] = a_7,$$
$$[a_6, a_2] = [a_6, a_3] = [a_6, a_4] = [a_6, a_5]$$
$$= [a_7, a_2] = [a_7, a_3] = a_8 \rangle$$

S has 22 (nontrivial) rational classes with 7 distinct cycle structures as shown in Table 3. The values of *list* and *away* for this example are

```
list is [ 1, 2, 11, 10, 18, 19, 20,
          21, 22, 13, 14, 12, 15, 3, 4, 16, 5, 17, 6, 7, 8, 9 ]
away is [ { 1 }, { 2 }, { 11 }, { 10 }, { 18 }, { 19 }, { 20 },
          { 21 }, { 22 }, { 13 }, { 14 }, { 12 }, { 15 }, { 3 },
          { 16, 4 }, { 16 }, { 17, 5 }, { 17 }, { 6 }, { 7 }, { 8 }, { 9 } ]
```

TABLE 3. Rational classes of Sylow 2-subgroup of Sp(4,7)

Rational Class	Rep	Order	Cent	$\|N/C\|$	Square	Cycles
1	a_8	2	2^8			$2^{192}1^{16}$
2	a_7	2	2^7			2^{200}
3	a_1	2	2^5			$2^{192}1^{16}$
4	a_5	2	2^5			2^{200}
5	a_4	2	2^5			2^{200}
6	a_3a_7	2	2^5			2^{200}
7	a_2a_7	2	2^5			2^{200}
8	a_1a_8	2	2^5			2^{200}
9	$a_2a_3a_5$	2	2^4			2^{200}
10	a_4a_5	4	2^6	2	$K(a_7)$	4^{100}
11	a_6	4	2^6	2	$K(a_8)$	$4^{96}2^41^8$
12	$a_4a_5a_8$	4	2^6	2	$K(a_7)$	4^{100}
13	a_3	4	2^5	2	$K(a_8)$	$4^{96}2^41^8$
14	a_2	4	2^5	2	$K(a_8)$	$4^{96}2^41^8$
15	a_1a_6	4	2^4	2	$K(a_7)$	4^{100}
16	a_1a_3	4	2^3	2	$K(a_5)$	4^{100}
17	a_1a_2	4	2^3	2	$K(a_4)$	4^{100}
18	a_2a_3	8	2^6	2	$K(a_6)$	$8^{48}4^21^8$
19	$a_2a_3a_7$	8	2^5	2^2	$K(a_6)$	$8^{48}4^22^4$
20	$a_1a_2a_3$	8	2^4	2	$K(a_4a_5)$	8^{50}
21	a_3a_4	8	2^4	2^2	$K(a_6)$	$8^{48}4^22^4$
22	a_2a_5	8	2^4	2^2	$K(a_6)$	$8^{48}4^22^4$

There is one G-rational class still to find after processing the cycle structures. It is S-rational class 12 with representative $a_4a_5a_8$. The computation of the Sylow subgroup takes 6 seconds; the conversion to a pc presentation takes 2.3 seconds; the calculation of the classes of $\overline{S}$ takes 0.2 seconds; the analysis of the S-rational classes takes 12 seconds; the computation of the eight centralizers in G takes 15 seconds; and the five conjugacy tests take 10 seconds.

4.4. **Algorithm for Problem 2a.** This section presents the determination of the classes within the rational class of z of order p^r. This is done by calculating the Galois group, $\mathrm{Gal}_G(z)$. Let S be a Sylow p-subgroup of G containing z. For $p = 2$, the representative of the G-rational class may be chosen to satisfy Corollary 2.0.1(3), whence S determines the Galois group. So here we will handle the case where $p > 2$. Even in this case, we can choose the representative to satisfy Corollary 2.0.1(1) and know the p-part of the Galois group.

The algorithm processes the subgroup lattice of $\mathrm{Aut}(Z_{p^r})$ below $Z_{(p-1)}$ in a top-down breadth-first fashion. Each subgroup is cyclic, generated by some j mod p^r, and corresponds to a conjugacy test of z with z^j. Working top-down allows us to terminate at the first positive conjugacy test. The algorithm produces a list of integers j representing powers z^j to guide the conjugacy testing, and a default set of generators for the Galois group (in $\mathbb{Z}/p^r\mathbb{Z}$). The default generators are the generators of $\mathrm{Gal}_S(z)$. The list is used to determine

the $Z_{(p-1)}$-component of the Galois group. The integers j in the list are taken in turn until a positive conjugacy test in G between z and z^j occurs. This integer j (regarded mod p^r) is the generator of the $Z_{(p-1)}$-component. If there is no positive conjugacy test, then the component is trivial.

The list is produced from a primitive root of p^r. The lattice of subgroups of $\mathrm{Aut}(Z_{p^r})$ below $Z_{(p-1)}$ is listed in a breadth-first fashion starting with $Z_{(p-1)}$. This list is then refined to

(1) eliminate subgroups whose order does not divide the order of G;
(2) eliminate subgroups which do not lie above a known Galois group, such as $\mathrm{Gal}_{N_G(S)}(g)$, and in this case the generators of the known Galois group are added to the default set;
(3) eliminate those subgroups which do not lie below the preimage of $\mathrm{Gal}_G(g^p)$, in those cases where the strategy of §4.2 means that $\mathrm{Gal}_G(g^p)$ is known;
(4) eliminate the subgroups whose order is divisible by m, if it is known that G contains no elements of order m.

It is a matter of heuristics to determine in which order each layer of the subgroup lattice should be processed. One wants to process first the power most likely to give a positive conjugacy test.

If $x \in G$ normalizes the element g, then $|\varepsilon_g(x)|$ divides $|x|$. So knowing the orders of elements in G may restrict the choice of subgroups of $\mathrm{Aut}(Z_n)$, where $n = |g|$, which could be isomorphic to $\mathrm{Gal}_G(g)$. The relevant orders of elements would be known if the primes were processed from small to large.

4.5. **Examples.** The group $PSL(5, 3)$ has an element z of order 121. The subgroup lattice of Z_{110} is given in Figure 1, so the first list of powers to consider is [2, 4, 32, 112, 56, 81, 120]. However, the group order is not divisible by 11^3, so we only need to look at the $Z_{(p-1)}$-component isomorphic to Z_{10}. (The Sylow 11-subgroup also tells us that the Z_{11}-component is trivial.) Hence, the list to consider is [112, 81, 120]. Furthermore, if we have determined the classes of $Kr_G(z^{11})$, we know the $Z_{(p-1)}$-component is a subgroup of Z_5.

Hence, the list to consider is [81]. (In this example, $C_G(z) = C_G(z^{11})$, so we could avoid all conjugacy testing by checking whether the element x, which conjugates z^{11} to its 4th power, conjugates z to its 81st power.)

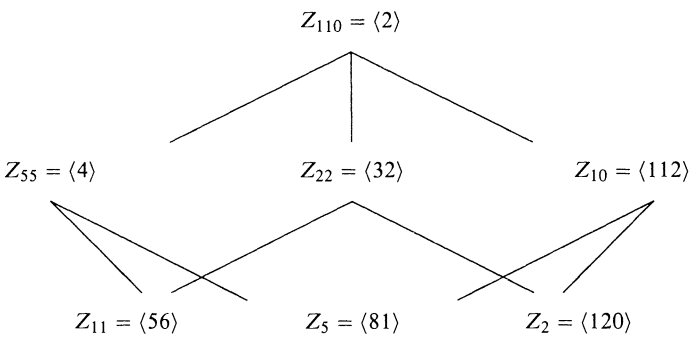

FIGURE 1. Nontrivial subgroups of cyclic group $Z_{110} = \mathrm{Aut}(Z_{121})$

5. Elements of composite order

In this section we assume that representatives of the rational classes of p-elements of G are known. Let z be such a representative of order p^r. We treat the following problems:

Problem. Find representatives y of the rational classes of G where $|y| = p^r t$, t is coprime to p and $y^t \in K_G(z)$.

Problem. Determine the classes within each rational class $Kr_G(y)$, where $|y| = p^r t$, t is coprime to p and $y^t \in Kr_G(z)$, by determining the Galois group $\mathrm{Gal}_G(y)$.

5.1. **Theory.** Without loss of generality we can restrict the search to elements y with $y^t = z$ and hence $y \in C_G(z)$. Let $C = C_G(z)$. Let Υ be the partition of Ω determined by the cycles of z. Let $f : C \to C\downarrow_\Upsilon$, $N = \ker(f)$, and $\overline{C} = C/N$. Note that $z \in N$, and that N is an abelian p-group, all of whose elements have order dividing p^r [7]. Suppose that

(‡) $\qquad y \in G$ has order $p^r t$, t is coprime to p, and $y^t = z$.

Then $\overline{y} = f(y) \in \overline{C}$ has order t.

Theorem 5.1.1. *Suppose* y_1, y_2 *satisfy* (‡). *Then* $y_1 \sim_G y_2$ *if and only if* $\overline{y}_1 \sim_{\overline{C}} \overline{y}_2$.

Proof. ($\Rightarrow$) If $y_1^g = y_2$, then $(y_1^t)^g = y_2^t$, so $g \in C_G(z)$. Modulo N we have $(\overline{y}_1)^{\overline{g}} = \overline{y}_2$, so $\overline{y}_1 \sim_{\overline{C}} \overline{y}_2$.

($\Leftarrow$) Suppose $g \in C$ such that $\overline{y}_1^{\overline{g}} = \overline{y}_2$. Without loss of generality assume that $y_1 = y$ and $y_2 = yn$, for some $n \in N$. Let $L = \langle y, n\rangle$ and $Z = \langle z\rangle$, which is central in L. The group L has a normal abelian Sylow p-subgroup $N \cap L$ and a quotient $\langle \overline{y}\rangle$ which is cyclic of order t coprime to p. Hence, L is soluble. Furthermore, $\langle yZ\rangle$ and $\langle ynZ\rangle$ are Hall subgroups of order t of L/Z, and hence they are conjugate in L/Z. Hence, $\langle y\rangle$ and $\langle yn\rangle$ are conjugate in L. We can assume the conjugating element lies in N, and that it conjugates yN to yN. Hence, there exists $n_1 \in N \cap L$ such that $y^{n_1} = yn$. □

The above result must be tempered by the fact that several class representatives which arise as roots of z may lie in the same rational class. The normalizer of z acts on C and $\overline{C}$, and its action determines whether class representatives $\overline{y}_1$ and $\overline{y}_2$ of $\overline{C}$ give rise to the same rational class.

Theorem 5.1.2. *Suppose* y_1, y_2 *satisfy* (‡). *Let* $g \in N_G(z)$ *be an element conjugating* z *to* z^m, $m \neq 1$. *Then* $y_1 \sim_G y_2^m$ *if and only if* $\overline{y}_1^g \sim_{\overline{C}} \overline{y}_2^m$. □

In the case where z is a p-element, $p \neq 2$, the Galois group of z is cyclic. Let $g \in G$ induce a generator of $\mathrm{Gal}_G(z)$. The action of g on $\overline{C}$ will fuse certain rational classes of $\overline{C}$. These, therefore, each give rise to the same rational class of G. Furthermore, there will be some smallest power g^t of

g such that $\bar{y}_1^{g^i} \in Kr_{\overline{C}}(\bar{y}_1)$. The Galois group $\mathrm{Gal}_G(y_1)$ is isomorphic to $\langle \varepsilon_z(g^l) \uparrow_{p^r}^{p^r t} \rangle \times \mathrm{Gal}_{\overline{C}}(\overline{y_1}) \uparrow_t^{p^r t}$.

5.2. **Algorithm.** The first task is to lift the class representative $\bar{y}$ from the quotient to an element of G such that it is actually a root of the p-element z. As the prime p and the order t of $\bar{y}$ are coprime, this is straightforward. Algorithm 3 gives the details.

Algorithm 3

Input : a permutation group G;
a p-element $z \in G$ of order p^r;
an element $\bar{y}$ of order t, coprime to p, in $C_G(z)$-quotient by N;

Output : the class representative in yN which is a tth root of z;

begin
lift the element $\bar{y}$ to an element $y \in G$;
let $at + bp^r = 1$;
$x := y^{bp^r}$; "$x^t = identity$ and $x = yy^{-at} \in yN$"
result is xz^a;
end.

The classes within the G-rational class of y are determined by the action of the normalizing elements of z on $\overline{C}$. This simultaneously tells us which $\overline{C}$-classes are fused by the action. For p-elements, we have actual elements of G which generate the Galois group, so we can let them act explicitly. For the G-rational class of composite elements, all we require is to identify the Galois group as a subgroup of $\mathbb{Z}/(p^r t)\mathbb{Z}$. Algorithm 4 outlines the method. The special cases are common and help us reduce the number of conjugacy tests in $\overline{C}$.

Algorithm 4

Input : a permutation group G;
a p-element z of G, $p \neq 2$, $|z| = p^r$;
the relevant rational classes of $\overline{C}$, the quotient of $C_G(z)$ by N;
a rational class representative $\bar{y}$ of order t, coprime to p, in $\overline{C}$;

Output : the rational class in G of tth roots of z in yN;
the fusion of $\overline{C}$-classes and rational classes to $\bar{y}$ under the induced action of $N_G(\langle z \rangle)$;

begin
determine y such that $y^t = z$;
in $\overline{C}$ compare cycle structure, class length, and Galois group, to determine the set, $poss$, of $\overline{C}$-rational classes which could fuse to $\bar{y}$;
$g :=$ generator of $\mathrm{Gal}_G(z)$;
if g is identity **then**
$\mathrm{Gal}_G(y) \uparrow^{p^r t} := \mathrm{Gal}_{\overline{C}}(\bar{y}) \uparrow_t^{p^r t}$;
else if $poss = \{Kr(\bar{y})\}$ **and** $Kr(\bar{y})$ has just one class

then

$\mathrm{Gal}_G(y) \uparrow^{p^r t} := \mathrm{Gal}_G(z) \uparrow_{p^r}^{p^r t} \times \mathrm{Gal}_{\overline{C}}(\overline{y}) \uparrow_t^{p^r t}$;

else

for $i := 1$ **to** $|\mathrm{Gal}_G(z)|$ **do**

$h := g^i$;

locate class of $\overline{y}^h$ in $\overline{C}$;

if class is in $Kr_{\overline{C}}(\overline{y})$ **then** break; **end**;

end;

$\mathrm{Gal}_G(y) \uparrow^{p^r t} := \langle \varepsilon_z(h) \uparrow_{p^r}^{p^r t} \rangle \times \mathrm{Gal}_{\overline{C}}(\overline{y}) \uparrow_t^{p^r t}$,

and we know the integer m such that $\overline{y}^h \in K_{\overline{C}}(\overline{y}^m)$;

end;

end.

For 2-elements, the structure of the Galois group, $F = \mathrm{Gal}_G(z)$, may be more complicated. However, if the group is cyclic then we proceed as in Algorithm 4. On the other hand, if $F = \langle g_1 \rangle \times \langle g_2 \rangle \simeq Z_{2^s} \times Z_2$, then there are two possible cases for the subgroup H of F which normalizes y. If $\overline{y^{g_2}} \in Kr_{\overline{C}}(\overline{y})$, then $H \simeq K \times Z_2$, where K is a subgroup of $\langle g_1 \rangle$, and so K can be computed by Algorithm 4. If $\overline{y^{g_2}} \notin Kr_{\overline{C}}(\overline{y})$, then H is cyclic, and is generated either by some power g_1^i or by a product $g_1^i g_2$. It can be computed by a modified version of Algorithm 4.

The centralizer of y in G is computed within the preimage of $C_{\overline{C}}(\overline{y})$. This could be improved by actually determining the action of this group on the kernel N, and computing the fixed points of y.

5.3. **Examples.** Consider the group $G = \mathrm{PSL}(4,2)$ and the prime $p = 5$. G has one rational class 5A of p-elements. Let z be a representative of this class. Then $\overline{C} = \langle \overline{y} \rangle$ is cyclic of order 3, and hence, $\overline{C}$ has one rational class of elements of order $t = 3$. This rational class consists of two classes with representatives $\overline{y}$ and $\overline{y}^2$. $\mathrm{Gal}_{\overline{C}}(\overline{y})$ is trivial. In G, the element z is normalized by g of order 4. In its action on $\overline{C}$, g conjugates $\overline{y}$ to $\overline{y}^2$, so $\mathrm{Gal}_G(y) = \langle \varepsilon_y(g) \rangle$, and g conjugates y to y^2.

Consider the group $G = \mathrm{PSp}(4,7)$ and the prime $p = 7$. G has a rational class 7C of p-elements. Let z be a representative of this class. Then $\overline{C}$ has order $2^3 7^2$ and degree 64. It has four classes of 2-elements, as shown in Table 4.

TABLE 4. Rational classes of 2-elements in $C_{\mathrm{PSp}(4,7)}(7C)$-quotient

Rational Class	$\lvert C_{\overline{C}}(\overline{y})\rvert$	Length	Cycles	Fusion
$\overline{14F}$	2^3	49	2^{32}	
$\overline{14C}$	$2^2 7$	14	$2^{30}1^4$	
$\overline{14D}$	$2^2 7$	14	$2^{30}1^4$	
$\overline{28C}$	2^2	98	4^{16}	$1 \sim 3$

There is an element $g \in G$ of order 6 normalizing z. In its action on $\overline{C}$, the element g fuses the $\overline{C}$-rational classes of involutions $\overline{14C}$ and $\overline{14D}$, so $\mathrm{Gal}_G(14CD) = \langle \varepsilon_{14CD}(g^2) \rangle$.

6. Ordering the primes

This section treats another strategic aspect of the computation, namely:

Problem. Determine the order in which the primes p_i dividing $|G|$ should be processed.

Our current strategy is to process them in increasing order of their exponent, and within the primes of equal exponent to process them in decreasing order. The rationale is that the p-elements, for the smaller primes p such as 2 and 3, have larger, more complex centralizers, and so the recursive treatment of the quotients of their centralizers should be avoided, or attention restricted to a small set of element orders t. The primes with high exponent generally have corresponding Sylow subgroups with a large number of rational classes, and so the fusion of p-elements in G may be difficult to determine.

The prime 2 is always last, in order to avoid p-elements with noncyclic Galois groups.

The order is assigned a priori using just the group order. The order could be much more flexible. After computing the Sylow subgroups and analyzing their rational classes, we might have a much better idea of the relative "difficulty" of processing the primes and their centralizers. We could also select the obvious G-rational class representatives (based, for example, on their distinct cycle structures) and compute their centralizers and quotients. This would give a clearer picture of which quotients might be "difficult" to analyze.

7. Experimental results

This section considers the performance of the prototype implementation. A detailed breakdown of performance is given in order to identify the bottlenecks in the computations. It compares the inductive schema with known algorithms—the calculation of orbits under the action of conjugation for small groups; the random method for moderate-size permutation groups; the top-down approach for soluble groups given by a conditioned pc presentation [17, 19]—to see whether the known algorithms should be used when analyzing the classes of a quotient which arises during the inductive schema.

The prototype implementation comprises over 2000 lines of Cayley code. The information it stores about each rational class is (a) a representative element g; (b) the order n of the elements; (c) the cycle structure of the elements; (d) the list of powers $1, m_2, \ldots, m_s$ such that g^{m_i} are the representatives of the classes within the rational class; (e) the corresponding list of sets of integers $\{n_{ij}\}$ such that $g^{n_{ij}}$ is in the same class as g^{m_i}—hence, the first set is the Galois group; (f) the centralizer of g; (g) the length of a class; (h) the generator(s) of the Galois group as integers mod n; and (i) the generator(s) of the Galois group as elements of G, for the case when g is a p-element. The power map for the G-rational classes of p-elements is computed and stored, because it is useful in avoiding conjugacy tests when fusing the S-rational classes in G. However, we have not extended this to information of the power map for all classes.

TABLE 5. Performance of prototype implementation of inductive schema

G	$\|G\|$	Degree	Number		Conjugacy Tests in G Time			Total Time for		Total	Random Method
			$\|Kr_G\|$	$\|K_G\|$	No.	Total	Worst	*pelts*	*roots*	Time	Time
Sp(4,2)	$2^4 3^2 5$	15	10	10	7	0	0	7	2	13	8
L(4,2)	$2^6 3^2 5\ 7$	15	11	13	7	0	0	11	4	20	8
L(3,4)	$2^6 3^2 5\ 7$	21	7	9	11	0	0	12	0	16	12
M24	$2^{10} 3^3 5\ 7\ 11\ 23$	24	21	26	11	0	0	53	12	77	16
U(3,3)	$2^5 3^3 7$	28	9	13	11	0	0	10	3	17	18
L(3,5)	$2^5 3\ 5^3 31$	31	13	29	14	0	0	19	7	31	18
L(5,2)	$2^{10} 3^2 5\ 7\ 31$	31	17	26	12	2	2	68	9	87	18
Sp(4,3)	$2^6 3^4 5$	40	14	19	11	0	0	20	9	34	22
L(4,3)	$2^7 3^6 5\ 13$	40	24	28	64	11	3	96	17	122	52
U(4,2)	$2^6 3^4 5$	45	14	19	10	0	0	20	9	35	22
L(3,7)	$2^5 3^2 7^3 19$	57	12	21	23	0	0	25	4	36	52
Sp(6,2)	$2^9 3^4 5\ 7$	63	29	29	25	0	0	55	23	84	102
L(6,2)	$2^{15} 3^4 5\ 7^2 31$	63	39	59	57	2844	2719	3418	55	3485	248
U(3,4)	$2^6 3\ 5^2 13$	65	8	21	19	0	0	22	4	30	44
L(3,8)	$2^9 3^2 7^2 73$	73	11	71	39	16	2	100	8	114	80
Sp(4,4)	$2^8 3^2 5^2 17$	85	17	26	25	6	1	56	14	79	136
L(4,4)	$2^{12} 3^4 5^2 7\ 17$	85	29	83	32	94	27	282	64	358	428
HS	$2^9 3^2 5^3 7\ 11$	100	22	24	19	0	0	50	13	74	186*
L(5,3)	$2^9 3^{10} 5\ 11^2 13$	121	48	114	216	106	2	716	81	810	988
U(3,5)	$2^4 3^2 5^3 7$	126	11	13	25	0	0	28	5	39	92
Sp(4,5)	$2^6 3^2 5^4 13$	156	24	33	29	9	2	62	35	104	234
L(4,5)	$2^7 3^2 5^6 13\ 31$	156	29	48	24	480	62	749	50	809	1280
U(5,2)	$2^{10} 3^5 5\ 11$	165	29	46	31	366	213	478	52	539	180
HS	$2^9 3^2 5^3 7\ 11$	176	22	24	21	3	1	65	18	100	176
Co3	$2^{10} 3^7 5^3 7\ 11\ 23$	276	32	37	65	57	2	247	87	369	408
U(4,3)	$2^7 3^6 5\ 7$	280	16	19	50	78	21	187	31	222	290
U(3,7)	$2^7 3\ 7^3 43$	344	19	57	26	33	3	102	22	131	524
Sp(4,7)	$2^8 3^2 5^2 7^4$	400	33	51	35	540	157	696	82	789	980
G(2,4)	$2^{12} 3^3 5^2 7\ 13$	416	23	32	23	51	6	256	70	367	544
U(3,8)	$2^9 3^4 7\ 19$	513	11	27	16	97	28	239	30	275	600
Sp(4,8)	$2^{12} 3^4 5\ 7^2 13$	585	27	82	52	433	89	1155	129	1297	4000
U(3,9)	$2^5 3^6 5^2 73$	730	19	91	105	1978	88	2018	90	2115	2446
U(4,4)	$2^{12} 3^2 5^3 13\ 17$	1105	30	94	90	18962	4006	19239	449	19703	29500
U(3,11)	$2^5 3^2 5\ 11^3 37$	1332	19	47	50	1204	75	1299	176	1484	8472
Tits	$2^{11} 3^3 5^2 13$	1755	16	22	16	260	48	703	185	1153	3032
Suzuki	$2^{13} 3^7 5^2 7\ 11\ 13$	1782	36	43	43	1657	184	2720	944	3842	6248
Held	$2^{10} 3^3 5^2 7^3 17$	2058	25	33	35	1816	135	2786	755	3851	13041*
U(3,13)	$2^4 3\ 7^2 13^3 157$	2198	27	183	69	11045	1275	11428	571	12009	44285*
Co2	$2^{18} 3^6 5^3 7\ 11\ 23$	2300	56	60	40	2606	253	7370	1320	9440	8700
G(2,5)	$2^6 3^3 5^6 7\ 31$	3906	36	44	52	1876	120	4190	3234	8133	28131

Table 5 and Table 6 (next page) present the performance of the prototype. Table 5 gives factual information about the group—including the order, degree, number of rational classes, number of classes—and then an overview of the prototype's performance—the number of actual conjugacy tests in G, the total time required for these conjugacy tests, and the worst single time for a conjugacy test; the total time taken to determine the classes of p-elements of G (including the computation of the Sylow subgroup), the time taken to compute the classes of composite elements; and then the total time taken by the prototype. For comparison, the last column shows the total time required by the random method to compute the conjugacy classes—an asterisk indicates that the method failed to find all the classes after considering 5000 random elements. Each individual time for a conjugacy test is rounded down. Table 6 breaks down the total times for *pelts* and *roots* according to the primes (and lists the primes in the order in which they were processed). All times are in seconds of CPU time on a Sun Sparcstation.

TABLE 6. Breakdown of performance by primes of prototype implementation

G	Total Time	Primes in order as processed prime p, *pelts* time, *roots* time														
L(3,5)	31	31	5	0	3	1	4	5	8	3	2	5	0			
L(3,7)	36	19	2	0	3	3	2	7	16	2	2	4	0			
L(3,8)	114	73	10	0	7	34	8	3	3	0	2	53	0			
L(4,2)	20	7	1	0	5	1	1	3	3	3	2	6	0			
L(4,3)	122	13	3	0	5	1	2	3	75	15	2	17	0			
L(4,4)	358	17	4	2	7	4	4	5	94	26	3	35	32	2	145	0
L(4,5)	809	31	8	0	13	5	2	3	13	29	5	706	19	2	17	0
L(5,2)	87	31	4	0	7	5	3	5	2	1	3	16	5	2	41	0
L(5,3)	810	13	6	13	5	2	8	11	29	0	3	630	60	2	49	0
L(6,2)	3485	31	5	0	5	1	10	7	70	14	3	2805	31	2	537	0
U(3,5)	39	7	2	0	3	4	3	5	17	2	2	5	0			
U(3,7)	131	43	10	0	3	3	13	7	53	9	2	36	0			
U(3,8)	275	19	17	0	7	24	7	3	55	23	2	143	0			
U(3,9)	2115	73	41	0	5	393	63	3	1534	27	2	50	0			
U(3,11)	1484	37	93	0	5	105	52	3	77	96	11	933	28	2	91	0
U(3,13)	12009	157	127	0	3	40	212	7	9743	312	13	1379	47	2	139	0
U(4,3)	222	7	6	0	5	6	0	3	147	31	2	28	0			
U(4,4)	19703	17	51	12	13	73	14	3	77	206	5	18686	217	2	352	0
U(5,2)	539	11	3	0	5	2	3	3	423	49	2	50	0			
Sp(4,4)	79	17	2	0	5	13	8	3	4	6	2	37	0			
Sp(4,5)	104	13	4	0	3	8	19	5	34	16	2	16	0			
Sp(4,7)	789	5	14	0	3	23	44	7	601	38	2	58	0			
Sp(4,8)	1297	13	23	7	5	10	0	7	241	53	3	149	79	2	732	0

TABLE 7. Comparative times for small groups

G	$\|G\|$	Degree	Action	Random	Inductive
L(2,2)	6	3	0.0	0.1	2
L(2,3)	12	4	0.0	1.1	2
L(2,4)	60	5	0.1	1.4	3
L(2,5)	60	6	0.1	1.5	4
L(3,2)	168	7	0.1	1.7	4
L(2,7)	168	8	0.1	1.8	4
L(2,9)	360	10	0.3	2.1	6
L(2,8)	504	9	0.5	2.0	6
L(2,11)	660	12	0.9	2.3	7
PSp(4,2)	720	15	0.9	4.4	13
L(2,13)	1092	14	1.4	2.5	8
L(2,17)	2448	18	3.9	3.2	8
L(2,19)	3420	20	6.3	3.4	10
L(2,16)	4080	17	7.4	3.2	12
L(3,3)	5616	13	8.7	3.0	11
L(2,23)	6072	24	13.4	4.2	12
L(2,31)	14880	32	43.2	5.7	15
L(4,2)	20160	15	29.3	3.6	20
L(3,4)	20160	21	40.7	6.5	16
PSp(4,3)	25920	40	75.8	10.9	34
L(2,32)	32736	33	125	6.8	29
L(2,64)	262080	65	480	48.5	77
L(3,5)	372000	31	1445	8.5	31
PSp(4,4)	979200	85	10200	40.2	79
L(2,128)	2097024	129	36400	121	266

The results in Table 5 demonstrate that the number of actual conjugacy tests being performed is under control (except in the cases of $PSL(5, 3)$ and $PSU(3, 9)$, where the Sylow 3-subgroup causes problems), and that the major performance bottleneck is the very long time required by some individual cases of the conjugacy test algorithm. Indeed, for $PSU(4, 4)$ there are four conjugacy tests between elements of order 5 where each test takes over 3800 seconds.

TABLE 8. Comparative times for soluble permutation groups

G	$\|G\|$	Degree	PCP	Random	Inductive
S_3	$2\,3$	3	0.2	0.0	2.2
S_4	$2^3 3$	4	0.4	0.1	3.7
$S_3 \wr S_3$	$2^4 3^4$	9	3.2	24.2	26.2
$S_3 \wr S_4$	$2^7 3^5$	12	8.7	45.2	81.2
$S_3 \wr V_4$	$2^6 3^4$	12	6.9	89.5	94.4
$S_4 \wr S_3$	$2^{10} 3^4$	12	12.8	196	149
$S_4 \wr V_4$	$2^{14} 3^4$	16	53.5	5000	875
$S_4 \wr S_4$	$2^{15} 3^5$	16	64.3	5280	775
$V_4 \wr S_4$	$2^{11} 3$	16	14.8	7570	990

The times in Table 7 compare the performance of known algorithms with the prototype of the inductive schema when working in a small group. The times are biased towards the known algorithms, as they only compute classes and not rational classes. It is further biased in their favor as the algorithms are implemented in C.

The times in Table 8 consider soluble permutation groups and compare the known algorithms with the inductive schema. The times in the PCP column include the time to compute the conditioned pc presentation, to compute the classes using the presentation, and to lift each class representative back to the permutation group. The information about rational classes, centralizers, and Galois groups is not computed and therefore not included in the times in the PCP column.

8. CONCLUSION AND FURTHER WORK

The major problems brought to light by the prototype implementation are

- the existence of cases where the backtrack algorithm for testing conjugacy in a permutation group takes a very long time;
- the number of conjugacy tests required when the group has a large Sylow p-subgroup, especially when p is not the last prime processed.

The first problem is fundamental, and somewhat outside the scope of future work on the schema itself. The prototype implementation in Cayley cannot make use of some features of the algorithm of [1]. An implementation in C could use the existing facilities of the algorithm which take advantage of known subgroups of the centralizers $C_G(g_1)$ and $C_G(g_2)$ when testing conjugacy of g_1 and g_2. These facilities can not be accessed from Cayley, so the prototype implementation cannot use the fact that it knows such subgroups as $C_G(g_1)$ and $C_S(g_2)$ when fusing p-elements from a Sylow subgroup S.

The second problem might be addressed in several ways using algorithms of [2, 10, 11, 15, 17] to compute normalizers and to compute in soluble groups. When S is abelian, then the fusion in G is determined by the fusion in the normalizer $N_G(S)$, and in cases where the normalizer is soluble, one could

convert to a conditioned pc presentation and easily compute the classes of the normalizer. (Note that for $p = 2$, $N_G(S)$ is always soluble.) Similar techniques in $N_G(S)$ or $N_G(H)$, where H is a normal (elementary abelian) subgroup of S, can also be applied. In general, they will not determine the fusion in G completely, but may help to greatly restrict the number of candidate elements for the representatives of the G-rational classes. Locating the appropriate subgroups H may require much more analysis of the structure of S.

Some of the most expensive conjugacy tests occur when determining the classes within a rational class of p-elements. Hence, it might be beneficial to study the order in which the subgroup lattice of $\mathrm{Aut}(Z_n)$ should be processed. One wants to find positive answers to whether the representative g is conjugate to g^m, as these conjugacy tests tend to be faster than exhaustively determining that no element conjugates g to g^m. (A similar remark could be made about testing conjugacy of a p-element against existing G-rational classes, when more than one G-rational class is compatible with respect to cycle structures, etc.)

The second problem would also benefit from a parallel implementation of the schema because then for each prime we would know a bound on the number of elements in the classes.

Minor gains could be made through determining when to switch from the inductive schema to known methods of determining the classes. In cases where $\overline{C}$ is soluble, or small, one could switch.

The inductive schema is successful in that it requires only a small number of conjugacy tests in the permutation group G. Its general performance, when applied to groups of large order and degree, is far superior to the random method. The effort expended on finding the G-rational classes of composite elements is generally small, and also a small proportion of the total time. The effort of finding the G-rational classes and classes of p-elements can be high if the order of the Sylow subgroup is large. It may require many conjugacy tests and consume the bulk of the total time. However, the main impediment to better performance is the cost of individual conjugacy tests in permutation groups using the backtrack algorithm.

Acknowledgments

This work has taken form over a long period of time, and has benefited from many discussions with colleagues. I would like to acknowledge the contributions of John Cannon, Reinhard Laue, Cheryl Praeger, and Charles Sims.

Bibliography

1. G. Butler, *Computing in permutation and matrix groups* II: *Backtrack algorithm*, Math. Comp. **39** (1982), 671–680.
2. ———, *Computing normalizers in permutation groups*, J. Algorithms **4** (1983), 163–175.
3. ———, *Effective computation with group homomorphisms*, J. Symbolic Comput. **1** (1985), 143–157.
4. ———, *A proof of Holt's algorithm*, J. Symbolic Comput. **5** (1988), 275–283.
5. ———, *Computing a conditioned pc presentation of a soluble permutation group*, TR 392, Basser Department of Computer Science, University of Sydney, 1990.

6. G. Butler and J. J. Cannon, *On Holt's algorithm*, J. Symbolic Comput. **15** (1993), 229–233.
7. ———, *Computing Sylow subgroups of permutation groups using homomorphic images of centralizers*, J. Symbolic Comput. **12** (1991), 443–457.
8. J. J. Cannon, *An introduction to the group theory language, Cayley*, Computational Group Theory (M. D. Atkinson, ed.), Academic Press, London, 1984, pp. 145–183.
9. V. Felsch and J. Neubüser, *An algorithm for the computation of conjugacy classes and centralizers in p-groups*, EUROSAM '79 (E. W. Ng, ed.), Lecture Notes in Comput. Sci., vol. 72, Springer-Verlag, Berlin, 1979, pp. 452–465.
10. S. P. Glasby, *Constructing normalisers in finite soluble groups*, J. Symbolic Comput. **5** (1988), 285–294.
11. S. P. Glasby and M. C. Slattery, *Computing intersections and normalizers in soluble groups*, J. Symbolic Comput. **9** (1990), 637–651.
12. M. Hall, Jr., *The theory of groups*, Macmillan, New York, 1959.
13. C. M. Hoffman, *Group theoretic algorithms and graph isomorphism*, Lecture Notes in Comput. Sci., vol. 136, Springer-Verlag, Berlin, 1982.
14. D. F. Holt, *The calculation of the Schur multiplier of a permutation group*, Computational Group Theory (M. D. Atkinson, ed.), Academic Press, London, 1984, pp. 307–319.
15. ———, *Computing normalizers in permutation groups*, J. Symbolic Comput. **12** (1991), 499–516.
16. B. Huppert, *Endliche Gruppen* I, Springer-Verlag, Berlin, 1967.
17. R. Laue, J. Neubüser, and U. Schoenwaelder, *Algorithms for finite soluble groups and the SOGOS system*, Computational Group Theory (M. D. Atkinson, ed.), Academic Press, New York, 1984, pp. 105–135.
18. J. S. Leon, *On an algorithm for finding a base and strong generating set for a group given by generating permutations*, Math. Comp. **35** (1980), 941–974.
19. M. Mecky and J. Neubüser, *Some remarks on the computation of conjugacy classes of soluble groups*, Bull. Austral. Math. Soc. **40** (1989), 281–292.
20. C. C. Sims, *Determining the conjugacy classes of a permutation group*, Computers in Algebra and Number Theory (G. Birkhoff and M. Hall, Jr., eds.), SIAM-AMS Proc., vol. 4, Amer. Math. Soc., Providence, RI, 1971, pp. 191–195.
21. ———, *Computing the order of a solvable permutation group*, J. Symbolic Comput. **9** (1990), 699–705.
22. H. Wielandt, *Finite permutation groups*, Academic Press, New York, 1964.

CENTRE INTERUNIVERSITAIRE EN CALCUL MATHÉMATIQUE ALGÉBRIQUE, DEPARTMENT OF COMPUTER SCIENCE, CONCORDIA UNIVERSITY, 1455 DE MAISONNEUVE BLVD. WEST, MONTREAL, QUEBEC, CANADA H3G 1M8
E-mail address: gregb@cs.concordia.ca