Some computations with finitely presented groups

George Havas

ARC Centre for Complex Systems,

School of Information Technology and Electrical Engineering,

The University of Queensland, Queensland 4072, Australia.

http://www.itee.uq.edu.au/∼havas

Email: havas@itee.uq.edu.au

Partially supported by the Australian Research Council

St Andrews, September 2003

Group theory is a particularly rich area for the design, implementation, analysis and application of practical computational methods.

Algorithmic processes have been developed across the various branches of the subject and they find wide application.

A brief history of some aspects:

1945 Turing suggested using his "automatic computing engine" to enumerate the groups of order 720

1951 MHA Newman suggested a method for estimating the number of groups of order 256

1958 Neubüser started work on a broad range of programs for group theory

1965 Cannon developed group theory algorithms (KDF9 assembler)

1969 GRAPPA (a new general purpose group theory system) prototyped by Cannon (BCPL)

1971 Work commenced on a new system (GROUP), a collaboration between Neubüser and Cannon. This was a large collection of group theory routines driven by a simple command language.

1973 Need for a full scale algebraic programming language recognized

1975 Development of CAYLEY started (FORTRAN)

1982 First production release of CAYLEY.

1986 Development of GAP started (C).

1987 250000 lines of CAYLEY translated (FORTRAN to C)

1992 V3.1 of GAP released

1994 V1.10 of MAGMA released

1995 MAGNUS released

1999 V4 GAP; V2.5 MAGMA

2003 V4.3 GAP; V2.10 MAGMA

Mathematicians have always computed, now often with machine assistance.

I examine some of the more important procedures for finitely presented groups and look at selected applications.

A well-known theorem asserts that, in general, the word problem for fp-groups is undecidable.

Consequently, computing with fp-groups is fundamentally different in nature to computing with groups given in some **concrete** form (for example, permutation groups or groups of matrices over finite fields).

Let G and K be two fp-groups.

Typical of the questions we wish to answer about fp-groups are the following.

- Is G the trivial group?
- Is G finite?
- If G is infinite, is it free?
- If G is finite, what is its order and structure?
- What are the abelian (nilpotent, soluble, perfect) quotients of G?
- Is G abelian (nilpotent, soluble, perfect)?
- Can we construct a small degree permutation representation for G?
- Can we construct a small degree matrix representation for G over some given field?
- Are the groups G and K isomorphic?

Finitely presented groups:

Enumerate cosets (Todd Coxeter)

Present subgroups (Reidemeister Schreier)

Simplify presentations (Tietze transformations)

Compute abelian quotients

Compute nilpotent quotients

Compute soluble quotients

String rewriting (Knuth Bendix)

Theorem proving

Genetic algorithms

Some methodology

References:

John Cannon and George Havas: Algorithms for Groups, Australian Computer Journal **24** (1992) 51–60.

Charles Sims: Computation with finitely presented groups, Cambridge University Press (Encyclopedia of Mathematics and its applications 48), 1994.

Many more recent research papers.

The GAP manual.

http://turnbull.mcs.st-and.ac.uk/circa/gapstuff/gapfiles/fpres.html

What can we do with finitely presented groups?

Many techniques used to compute with fp-groups may be conveniently described under three headings.

- Todd-Coxeter or coset enumeration based methods
- Knuth-Bendix or term-rewriting methods
- Quotient group methods

Coset Enumeration and Related Algorithms

Given an fp-group $G = \langle x_1, \ldots, x_r \mid R_1, \ldots, R_s \rangle$
(where $R_1, \ldots, R_s$ are words in the generators $x_1, \ldots, x_r$),

and given a subgroup H of G, $H = \langle h_1, \ldots, h_t \rangle$
(where $h_1, \ldots, h_t$ are also words in the generators),

coset enumeration procedures try to construct a permutation representation for G, corresponding to the action of G by (right) multiplication on the (right) cosets of H, by means of a trial-and-error process.

The cosets are traditionally identified with the integers $1, \ldots, n$, where coset 1 always corresponds to the given subgroup H.

A new coset k is defined as the image of some existing coset i under (right) multiplication by some generator x_j of G or by an inverse x_j^{-1}.

The cosets must satisfy the following conditions:

(a) coset 1 must be mapped to itself by each of $h_1, \ldots, h_t$;

(b) each coset j must be mapped to itself by each of the defining relators $R_1, \ldots, R_s$ and by each product $x_k x_k^{-1}$.

The action of the G-generators on the cosets is stored in a two-dimensional array known as a **coset table**.

Enforcement of rules (a) and (b) yields values for some hitherto unknown coset table entries (**deductions**) and, also, the identification of cosets which have been multiply defined (**coincidences**).

The procedure terminates when

(i) for each coset i, the action of each generator x_j and inverse x_j^{-1} on i is known; and

(ii) rules (a) and (b) are satisfied.

Such procedures were used in hand computation prior to the development of computers.

Beginning in 1952, different versions of the procedure have been adapted for machine computation and it is very much used in computational group theory.

However, despite its antiquity, our understanding of the relationship between a given presentation for G and the performance of a particular version of coset enumeration when applied to that presentation is poor.

Introductory descriptions of the procedure are given by Cannon, Dimino, Havas and Watson (1973), Johnson (1980), Leech (1970b, 1984) and Neubüser (1982).

Sims (1994) gives a formal account of coset enumeration in terms of automata and rational languages.

Performance of the procedure is very sensitive to changes in the rules used to introduce new cosets.

For a given coset enumeration procedure, there is no computable bound, in terms of length of input and a hypothetical index, to the number of cosets which need to be defined in the coset enumeration process to complete the enumeration.

(The existence of such a bound would violate the unsolvability of the word problem for finitely presented groups.)

Further, Sims (1994) has proved that there does not exist a polynomial bound, in terms of the maximum number of cosets defined, for the number of coset tables which may be constructed using simple coset table operations similar to those employed in a coset enumeration procedure.

This result indicates that the running time of a coset enumeration procedure, as a function of available space, may be unpleasant.

In the 1973 paper we identify a number of factors that affect the efficiency of an enumeration.

The next major study of coset enumeration procedures was carried out by Sims (1994).

After extensive experimentation, I introduced new coset enumeration strategies, which exhibit dramatically better performance than previous versions when applied to many "difficult" enumerations.

The performance of a coset enumeration program, in doing a particular enumeration, is measured in large part by the **maximum** number of cosets that are simultaneously defined and by the **total** number of cosets defined during the course of the enumeration.

The maximum gives a direct measure of the storage requirements.

For example, consider the presentation

$$\langle x, y, z \mid x^y x^{-3}, y^z y^{-2}, z^x z^{-4} \rangle$$

for a group of order 210, essentially due to Mennicke (1959).

The subgroup $\langle x \rangle$ has index 105.

The methods described in 1973 define as many as a maximum of 1230970 cosets and a total of 1250191 to complete this enumeration, and the most space-economical of those methods requires a maximum of 127846 cosets and a total of 128218.

The new procedure, with a default strategy, requires a maximum of 2854 cosets and total 2859, while individual tuning can improve on that.

In favourable circumstances, the current generation of programs may successfully complete enumerations where the index of H in G is more than one billion.

Unfortunately, it is easy to construct presentations for the trivial group which defeat these programs.

Coset enumeration is the basis of the standard computational technique used when trying to prove that an fp-group G is finite.

If the procedure terminates, given a subgroup H of G known to be finite, we immediately deduce that G is finite and, moreover, we obtain a bound on the order of G.

If G is not only finite but also sufficiently small so that the cosets of the trivial subgroup may be enumerated, the resulting coset table provides us with a solution to the word problem for G.

The range of applicability of current coset enumeration programs is limited mainly by the memory required to store the coset table.

Since, in the case of non-pathological enumerations, the space required is roughly proportional to the index of H in G, a next step is a procedure capable of enumerating the double cosets HxL of subgroups H and L of G.

Linton (1991a) implemented a double coset enumeration procedure, first suggested by Conway (1984), for the case where H is a "large" subgroup of G and L is a small subgroup in which detailed structural computation is possible.

Since the number of double cosets HxL is often a small fraction of the number of single cosets Hx, this technique, when applicable, offers potentially great space savings.

Classical coset enumeration procedure constructs a permutation representation for G on the cosets of H.

Linton (1991b) describes a version of coset enumeration which constructs a matrix representation for G over a designated field k (usually a finite field).

In the simplest interpretation, Linton's algorithm constructs the permutation module corresponding to the action of G on the cosets of a subgroup H.

So far I have assumed that we are given some subgroup H of G.

How do we proceed when we are unable to identify useful subgroups by direct inspection of the presentation?

If G is an fp-group then, for each positive integer n, there exist only finitely many subgroups H of G of index n.

More precisely, given a homomorphism $\phi : G \to Sym(n)$, of G into the symmetric group of degree n, such that $\phi(G)$ is transitive, then $H^\phi = \{g \in G \mid 1^{\phi(g)} = 1\}$ is a subgroup of index n in G.

Such homomorphisms may be constructed by enumerating coset tables.

Given a generating set for G and a positive integer n, there is a one-to-one correspondence between the subgroups having index n in G and the set of **standard** coset tables having n rows (where the entries in standard tables satisfy certain ordering conditions).

The **low index subgroup algorithm**, discovered independently in the sixties by Sims and Schaps, enumerates all n-row standard coset tables using a combination of coset enumeration and backtrack search methods (Dietze and Schaps, 1974; Neubüser, 1982; Sims, 1994).

The low index subgroup algorithm outputs either a generating set for each subgroup of index n, or a list of all transitive permutation representations of G having degree n.

Various versions of the low index subgroup algorithm exist.

In favourable circumstances, it may be used to find all subgroups having index up to 100 or even more.

For example, a Cannon-Sims implementation is able to compute all conjugacy classes of subgroups in the Coxeter group $\langle a, b, c, d \mid a^2, b^2, c^2, d^2, (ab)^5, (bc)^3, (cd)^3, [a, c], [b, d], [a, d]\rangle$ with index not exceeding 240 quite readily.

Since this technique is applicable to infinite groups, it provides us with a tool for proving that an fp-group is infinite.

The structure of each quotient H/H', where H is a subgroup produced by the low index algorithm, is examined for the presence of infinite factors.

Having the coset table for a subgroup H of the fp-group G enables us to construct a presentation for H.

A lemma of Schreier describes a generating set for H in terms of the generators of G and a system of coset representatives for H in G.

(Such coset representatives may be read off the coset table for H in G.)

The Reidemeister rewriting process permits us to rewrite the relators of G and their conjugates as words in these Schreier generators.

The ensuing relators constitute a presentation for H.

Again, the coset table contains all the information needed to perform Reidemeister rewriting.

Details are given by Johnson (1980) and Neubüser (1982).

A variation of the Reidemeister rewriting process, which rewrites H-elements given as words in the generators of G as words in an arbitrary generating set for H, was described by Benson and Mendelson (1966) (see also Neubüser, 1982).

Using this theoretical basis, programs for constructing presentations of subgroups of finite index in fp-groups have been implemented by a number of people, including Havas (1974) and Arrell and Robertson (1984).

Subgroup presentations produced by a Reidemeister-Schreier process generally involve large numbers of generators and relators and are poorly suited for human or computer use.

A theorem of Teitze asserts that, given presentations for two isomorphic groups, repeated application of three simple rules (**Tietze transformations**) will suffice to transform one presentation into the other.

However, there is no general algorithm for identifying the order in which the Tietze transformation rules should be applied.

Presentation simplification programs which take "bad" presentations and produce "good" presentations for groups have been developed by Havas, Kenne, Richardson and Robertson (1984), Robertson (1988) and Havas and Ollila(1993).

Coset enumeration over a subgroup H constructs a permutation representation for G.

We can now study the quotient of G defined by this representation using permutation group methods.

Conversely, given a finite group G in some concrete representation, we may use the group multiplication to construct coset tables.

For such a group G with moderate order, Cannon (1973) shows how to construct a compact presentation for G from a coset table so that we can then apply fp-group methods to G.

Term-rewriting Methods

A specialization of the Knuth-Bendix term-rewriting procedure (Knuth and Bendix, 1970) has been applied to fp-groups.

Starting with a finite presentation for a monoid M, the Knuth-Bendix procedure for strings (**KB-procedure**) attempts to construct a **confluent** presentation for M.

A confluent presentation for M consists of a system of rewriting rules which convert any element of M into a unique normal form.

The KB-procedure has been studied extensively.

Some applications to groups are given by Gilman (1979) and Le Chenadec (1986), while Sims (1994) investigates the procedure in detail.

A major success of the KB-procedure in group theory was its application by Sims (1987) to verifying nilpotency of an fp-group.

The nilpotent quotient algorithm is used initially to construct a polycyclic presentation for the nilpotent quotient Q of G.

Using the relations of this presentation as an initial set of rewrite rules, and a special term-ordering, Sims was able to develop an effective algorithm for verifying the triviality of the kernel of the quotient Q. (If the kernel is nontrivial, the algorithm fails to terminate.)

An advantage of the KB-procedure over coset enumeration is that it may sometimes be used to construct a confluent presentation for an infinite group.

In the case of a finite fp-group, coset enumeration is usually the most efficient method for constructing a confluent presentation.

Epstein, Holt and Rees (1991) describe a practical algorithm based on the KB-procedure for constructing a solution to the word problem for groups known as **automatic** groups.

This class of groups has solvable word problem and includes many important families of groups which arise naturally in geometry and topology (for example, hyperbolic and Euclidean groups).

Holt and Rees (1992) developed a program which endeavours to determine whether or not two fp-groups G and K are isomorphic.

The program alternates between attempting to construct an isomorphism between G and K, and attempting to prove nonisomorphism by discovering a structural difference.

The Knuth-Bendix procedure is used to construct a **reduction machine** for each of G and K.

These two reduction machines are used to systematically construct homomorphisms $\theta : G \rightarrow K$ and then to test each such homomorphism θ for the properties of being surjective and injective.

The nonisomorphism testing relies on finding some structural difference by comparing various kinds of quotients of G and K. When applied to special classes of groups, such as automatic groups, it can be quite successful.

For example, Holt and Rees report that it was able to quickly settle the isomorphism question for all but two pairs in a collection of about 30 pairs of link groups.

It resolved the question for the last two pairs with more difficulty, taking some hours on a Sun 3/60.

Sims (1991) employed the KB-procedure to deduce non-obvious relations in two groups.

In each case, the relations could not have been discovered using the current generation of coset enumeration procedures.

This is one of the first reported instances where the KB-procedure outperforms coset enumeration when both are potentially applicable.

Quotient Group Methods

An important technique for studying an fp-group G involves constructing homomorphic images of G, which may be permutation groups,matrix groups or members of some class of fp-groups having solvable word problem.

A successful coset enumeration over the cosets of some subgroup yields a homomorphism onto a permutation group, while the low index algorithm systematically searches for homomorphisms into the symmetric group $Sym(n)$, for small n.

Now we examine techniques for directly constructing abelian, nilpotent and soluble quotients of an fp-group.

In each case, a confluent presentation for the quotient group is constructed.

Note that, if the particular quotient is equal to G, this effectively solves the word problem for G.

The structure of a finitely generated abelian group A may be obtained by computing the Smith normal form of its relation matrix (an integer matrix).

Relatively efficient algorithms for computing this form for large matrices have been described by Havas and Sterling (1979) and Havas, Holt and Rees (1993).

Thus, given an arbitrary fp-group G, the structure of its maximal abelian quotient, G/G', is readily obtained.

Hand computations in the sixties led to the development by Macdonald (1974) and Wamsley (1974) of algorithms for computing finite nilpotent p-quotients of G, where p is a prime dividing the order of G/G'.

Starting with an exponent-p-quotient of G/G', the algorithms successively extend a current p-quotient H by an elementary abelian group that is centralized by H.

Since the extension theory is particularly simple in this situation, it is possible to design extremely effective algorithms.

Nice descriptions of are given by Newman (1976), Havas and Newman (1980) and Newman and O'Brien (1995).

The algorithm outputs the p-quotient in terms of a **power-commutator presentation**, a special confluent presentation.

The p-quotient algorithm relies critically on a particular string rewriting procedure known as **commutator collection**, where the rewrite rules are the relations of a power-commutator presentation (Felsch, 1976; Havas and Nicholson, 1976; Leedham-Green and Soicher, 1990; and Vaughan-Lee, 1990b).

The algorithm has been extensively applied to the investigation of Burnside groups (see Vaughan-Lee, 1990a).

As illustrations of the power of current implementations, the p-quotient algorithm has computed (a pcp for) the three-generator restricted Burnside group of exponent 5, a group with class 17 and order 5^{2282} and the two-generator restricted Burnside group of exponent 7, a group with class 28 and order 7^{20416}.

Building on the p-quotient algorithm, Leedham-Green and Newman designed and implemented an algorithm for generating descriptions of p-groups.

In his PhD thesis at the Australian National University, O'Brien significantly refined these methods and successfully applied them to determine all groups with order dividing 256: there are 56092 groups of order 256 (O'Brien, 1990, 1991).

A number of variations on the original p-quotient algorithm have been made.

Thus, a general nilpotent quotient program (with no dependency on a prime p and allowing infinite quotients) was prototyped by Sims in Mathematica and an implementation was developed by Nickel at ANU.

Havas, Newman and Vaughan-Lee (1990) produced an analogue of the group nilpotent quotient algorithm for graded Lie algebras.

This has applications to p-groups where the quotients are too large to be handled by the group program and has been used to investigate questions related to the Burnside problem.

Finally, Vaughan-Lee has developed a variation for finitely presented associative algebras.

A more difficult area is the computation of soluble quotients.

Wamsley (1977), Leedham-Green (1984) and Plesken (1987) proposed generalizations of the nilpotent quotient algorithm to a soluble quotient algorithm.

Success has been reported in specific cases by Neubüser and Sidki (1988), Newman and O'Brien (1992) and Havas and Robertson (1994), all using combinations of previously described programs.

The Plesken algorithm was implemented by Wegner at St Andrews as part of his PhD thesis and had some success.

Niemeyer, in her PhD work at ANU, developed some other effective soluble quotient algorithms.

She used it to do various interesting computations with $B(2,6)$, a soluble group of order $2^{28}3^{25}$.

Baumslag, Cannonito and Miller (1981) outlined a method for constructing polycyclic quotients.

While their interest was purely theoretical, Sims (1990b) has further developed their ideas and implemented them in the special case of metabelian quotients.

An implementation of the general algorithm (Sims 1994, chapters 9 and 10) involves a sophisticated combination of many algorithms, including Gröbner basis techniques (Buchberger, 1985).

At the DIMACS Workshop on Groups and Computation in 1991, Holt and Rees demonstrated a program **quotpic** for finding certain quotients of finitely presented groups.

A backtrack search attempts to construct a homomorphism between the given fp-group G and nominated permutation representations of selected finite groups.

In particular, this program may be used to identify small perfect groups that occur as quotients of G.

This work builds on the classification by Holt and Plesken (1989) of all perfect groups of order up to a million.

Having found a representation of G, the program converts it to a regular representation, and then attempts to construct larger quotients of G by performing elementary abelian extensions using either a Reidemeister-Schreier or p-quotient algorithm.

The portion of the subgroup lattice obtained is represented graphically and may be manipulated interactively.

Some motivating problems

Nice presentations

Finite simple groups

The Burnside problem

Nice presentations

What is a nice presentation?

(Beauty is in the eye of the beholder)

Aesthetics:

Symmetric

Concise

Good to compute with

Minimal generating set

Minimal set of relators

Bad news: These criteria often clash

Concise presentations may be necessarily asymmetric.

Concise presentations may be hard to compute with and even hard to find.

How few relators will suffice to present a finite group?

There is plenty of theory:

The *deficiency* of a finite presentation $P := \{X \mid R\}$ of G is $|R| - |X|$.

The deficiency of G, def(G), is the minimum of the deficiencies of all finite presentations of G.

For a finite group G we denote the Schur multiplier by $M(G)$.

G is said to be *efficient* if def(G) = rank($M(G)$).

In fact def$(G) \geq d_G(R/R') \geq d(R/[R,F])$.

Kovács (1995): "one can expect that inefficient groups are for more common than efficient ones."

In general it seems to be a hard problem to decide whether a given group is efficient.

The problems of proving specific groups efficient and of finding inefficient groups have been much studied.

All known examples of finite groups of deficiency zero can be generated by at most 3 elements.

The first examples which cannot be generated by 2 elements were found by Mennicke (1959); others have been found by Wamsley (1970), Post (1978) and Johnson (1979).

Only a very few of these examples are known to have prime-power order.

By systematic and substantial use of implementations of algorithms, we (H, Newman and O'Brien, 1994) produced

many more examples of p-groups with deficiency zero which have generator number 3.

We proved that a p-group with generator number 3 and trivial multiplicator has order at least p^8.

There are none of order 2^8 and 14 isomorphism types of order 3^8.

How many of these have deficiency zero?

Knowing the groups turns out to be of little direct help in searching for balanced presentations.

Instead we systematically generate appropriate balanced presentations and study these in some detail.

How can we decide whether such a presentation presents a group of interest?

As is well-known there is no algorithm for deciding whether a finite presentation defines a finite group.

However, if the presentation defines a finite group then there are procedures which will, in principle, prove this fact.

Of these coset enumeration is generally the best in practice.

Current implementations of coset enumeration can enumerate **billions** of cosets.

But we can't afford to enumerate a billion cosets too often.

So we precede use of coset enumeration by faster tests which filter out presentations which have larger finite quotients than the ones being sought.

In practice, we calculate the p-quotient determined by each presentation to an appropriate class.

If this quotient has the required order, we calculate the largest soluble quotient.

If this is also correct, we try to prove that the group is finite.

This method led to balanced presentations for 10 of the 14 groups of order 3^8.

For the remaining groups we gave balanced presentations which define them as pro-3-groups and as soluble groups, but we were not able to prove that they define the group.

We published 4 explicit candidates: $\{a, b, c$:

11: $cac^{-1}b^{-1}aba,\ \ bacba^{-1}c^{-1}b,\ \ cb^{-1}acbca^{-1}\}$

12: $acab^{-1}c^{-1}ab,\ \ b^2a^{-1}c^{-1}acb,\ \ ca^{-1}b^{-1}cabc\}$

13: $acab^{-1}c^{-1}ab,\ \ acbc^{-1}ba^{-1}b,\ \ b^{-1}abc^2a^{-1}c\}$

14: $a^3 = [c^{-1}, b],\ \ b^3 = a^{-1}cabc^{-1}b^{-1},$
$c^3 = [a^{-1}, b^{-1}][a^{-1}, c][b, c]\}$

Theorem 1. *(H, Holt, Kenne, Rees, 1999):*
#14 presents an infinite group.

Method of proof:

By using the automatic groups program (based on Knuth-Bendix rewriting), we show #14 is automatic, and deduce that it is infinite by considering the language of its word-acceptor.

(Some of these computations are hard.)

Derek Holt has used the same method to prove #13 is infinite.

What about soluble groups? Is there a bound on the derived length of deficiency zero groups?

In response to a problem posed by Johnson and Robertson (1977 and Problem 8.12, Kourovka notebook), efficient presentations for soluble groups with increasing soluble length have been sought.

The following example provides the first published deficiency zero presentation of a group with derived length seven.

Theorem 2. *(HHKR 1999):*

$G_7 = \langle x, y \mid x^4y^{-3},\ x^{-2}y^{-1}x^{-1}y^{-1}(xy)^2xy^{-1}xy \rangle$

is finite and has derived length seven.

The presentation was found by using a variant of a method described by Campbell and Robertson (1984), starting with a pair of permutation generators having orders 3 and 4 which generate a group having derived length six.

Coset enumeration shows that the subgroup generated by x has index $2^7 3^8$ in the central quotient G_7^* of G_7 obtained by replacing the first relator with the two relators x^4 and y^3.

This suffices to show that G_7 is finite.

The order of G_7 may be determined by constructing a presentation for the derived subgroup of G_7 (which has index 3) and performing a coset enumeration over the trivial subgroup of that presentation.

Thus G_7 has order $2^{10}3^9$.

The derived length of G_7 was found by using the ANU SQ algorithm available in GAP.

What about simple groups? Are finite simple groups efficient?

Various presentations for simple groups have been catalogued: ATLAS, symmetric, 2 generator, and efficient.

A survey of some results for simple groups of order less than one million is given by Campbell, Robertson and Williams (1989).

There were two simple groups of order $\leq 10^6$ whose deficiency remained to be determined, namely $L_3(5)$ and $S_4(4)$.

Note: both $L_3(5)$ and $S_4(4)$ have trivial multiplier; if efficient they have balanced presentations.

Theorem 3. *(Campbell, H, Hulpke, Robertson)*
$L_3(5)$ *is efficient.*

Our Approach

As starting points we have presentations for the minimal generating pairs for these groups.

A *minimal generating pair* for G is a pair $x, y \in G$ such that $G = \langle x, y \rangle$, $|x| = 2$ and $|y|$ is minimal under all such y for a given involution x.

It is sufficient to consider such pairs up to automorphisms.

Lemma 4. *Let G be a simple group with trivial Schur multiplier. Suppose G has a presentation of the form*

$$P = \left\{a, b \mid a^2 = 1, b^p = 1, w(a, b) = 1\right\}$$

with p prime. Then G has deficiency zero.

Method of proof:

For a word w in the free group generated by a and b, let $e_a(w)$, $e_b(w)$ be the exponent sums of a and b in w.

(For $e_b(w) \equiv \frac{1}{2} \pmod{p}$) consider a new word $\bar{w}(a,b) = w(a,b)a^{1-e_a(w)}b^{\frac{p+1}{2}-e_b(w)}$.

$\left\{a, b \mid a^2b^p = 1, \bar{w}(a,b) = 1\right\}$ is a deficiency zero presentation for G.

Computational Approach

Suppose G is a finite simple group with a presentation of the form

$$\left\{a, b \mid a^2 = 1, b^p = 1, w_i(a, b) = 1, i = 1, \ldots, k\right\}$$

with p prime.

A naïve approach is to replace the $w_i(a, b)$ $(i = 1, \ldots, k)$ by fewer relations by replacing the two relations $w_s(a, b)$ and $w_t(a, b)$ by $g^{-1}w_s(a, b)gw_t^{\pm 1}(a, b)$ for some word $g = g(a, b)$.

Such an approach has been used successfully in the past by Kenne (1986) and Jamali and Robertson (1989).

We automated this approach in an attempt to find a presentation for $L_3(5)$ of the given form.

Despite improved computing facilities, both in terms of hardware and software, since the earlier successful attempts described in CRW89 we did not find a presentation of the required form using this technique.

We therefore tried another approach.

We looked for words $w(a,b)$ such that $w(a,b) = 1$ in $L_3(5)$ and $e_a(w)$ is odd, $e_b(w) \equiv 1 \pmod 3$.

To do this we wrote a general program in GAP to attempt to find such words in any perfect group with a $(2,p)$ generating pair.

Using that program we find for the generating set given by the presentation (Campbell and Robertson, 1984):
$\{a, b \mid a^2 = b^3 = (ab)^{31} = (ab)^9 aBab((aB)^3 ab)^3 aB =$
$(ab)^4 (abaB)^2 (ab)^5 (aB)^4 (abaB)^2 (aB)^3 = 1\}$
(where $B = b^{-1}$) for $L_3(5)$ that $w(a, b) =$

ababaBaBabaBaBababababaBaBabaBaBababababaBababaBabab

of syllable length 50 is a shortest such word with $e_a(w) \equiv 1$ (mod 2), $e_b(w) \equiv 1$ (mod 3).

A coset enumeration on the resulting presentation

$$\mathcal{L} = \left\{a, b \mid a^2 = 1, b^3 = 1, w(a, b) = 1\right\}$$

shows that the index of $\langle b \rangle$ is 124000, which implies that $\langle \mathcal{L} \rangle \cong L_3(5)$.

Lemma 4 now shows that $L_3(5)$ has an efficient presentation $\bar{\mathcal{L}} = \left\{a, b \mid a^2b^3 = 1, \bar{w}(a, b) = 1\right\}$; and $\bar{w}(a, b)$ can be written as a word of length 50, eg:

$\bar{w}(a, b) =$ *AbAbABAbAbABaBaBAbAbABAbAbABaBaBababaBababaBABAB*

What about $S_4(4)$?

So far this type of approach has failed. Deficiency one presentations exist.

What about presentations for simple groups in general?

Presentations for many sporadic simple groups and their automorphism groups have been published.

Indeed, until now, the Thompson group Th was the only sporadic simple group S for which there was no published presentation for S or $\mathrm{Aut}(S)$.

In 2001 we (H, Soicher, Wilson) completed the determination of a presentation for Th ($\cong \mathrm{Aut}(Th)$).

In the process, we also found presentations for ${}^3D_4(2)$, ${}^3D_4(2){:}\,3$, $G_2(3){:}\,2$, and $C_{Th}(2A)$.

The proof of correctness of the presentation for Th uses a coset enumeration of 143,127,000 cosets.

Theorem 5. *Consider the group T presented by generators a, b, c, d, e, s, t, u and relators* (1) – (6)*:*

$$a^2, b^2, c^2, d^2, e^2, (ab)^3, (ae)^2, (bc)^3, (bd)^2, (be)^2,$$

$$a = (cd)^4, (ce)^2, (de)^3, (bcde)^8, \tag{1}$$

$$s^7, [s, a], [s, b], [s, c], (sd)^2, [e, s] = e^{s^3}, \tag{2}$$

$$t^3, [t, a], [t, b], [t, c], [t, d], [t, e], s^t = s^2, \tag{3}$$

$$u^2 = ac, [u, a], [u, c], [u, e], (ded^u)^2,$$

$$[u, (ac)^b] = e, [u^d, (ac)^b] = ue(ac)^b u^d ec, \tag{4}$$

$$t^u = t^{-1}, \tag{5}$$

$$[e, u^{s^2}], ac = (us)^3 = [u, s]^4, (du^{s^2})^4 = acc^d c^{des^{-1}} c^{des^2}. \tag{6}$$

Then the following hold:

1. $T \cong Th$.

2. *Generators* a, b, c, d, e, s *together with relators* (1), (2) *is a presentation for* ${}^3D_4(2)$.

3. *Generators* a, b, c, d, e, s, t *together with relators* (1), (2), (3) *is a presentation for* ${}^3D_4(2) : 3$.

4. *Generators* a, b, c, d, e, u *together with relators* (1), (4) *is a presentation for* $G_2(3) : 2$.

5. *Generators* a, c, d, e, s, t, u *together with the relators from* (1) – (6) *not involving* b *is a presentation for* $C_{Th}(2A)$.

Method of proof:

By coset enumeration and computations with the 248-dimensional $GF(2)$-matrix representation of Th available from the ATLAS of Finite Group Representations (Wilson, 1998, 1999).

Burnside (1902) considered the freest group on d generators with exponent n (denoted $B(d,n)$) and (in effect) asked:

is $B(d,n)$ finite and, if so, what is its order?

Many people have investigated aspects of these and related questions in detail.

Presently it is well-known that $B(d,n)$ is finite if n is 1, 2, 3, 4 or 6 (or if d is 1), and no others are known to be finite.

$B(d,n)$ is known to be infinite for $d \geq 2$ and $n \geq 8000$.

This focusses particular attention on exponent 5.

Is $B(2,5)$ finite? (Its largest finite quotient, $R(2,5)$, has order 5^{34}.)

First problem to solve: Burnside groups satisfy identical relations; how do you find finite presentations for them (if they exist)?

Another problem: 5^{34} is a big number:

$$582076609134674072265625.$$

Suffice it to say, not much progress has been made on $B(2,5)$ itself.

There is a natural relationship between order relations and Engel relations: $[x, \underbrace{y, y, ..., y}_{n}]$.

Vaughan-Lee (1997) showed that if G is a group of exponent 5, and if G satisfies the Engel-4 identity $[x, y, y, y, y] = 1$ (for all $x, y \in G$) then G is locally finite.

By a result of Traustason (1995) this implies that Engel-4 5-groups are locally finite.

Newman and Vaughan-Lee (1999) have determined the exact orders of the largest m-generator Engel-4 group of exponent 5.

All of these things rely on many computations, both hand and machine, in groups, Lie algebras and Lie superalgebras.

What is the situation for Engel-4 groups more generally?

It is reasonable to hypothesize that exponent-p Engel-4 groups are locally finite, or that Engel-4 groups generated by elements of finite order are locally finite.

As a first step, Michael Vaughan-Lee and I are trying to prove that the largest 2-generator Engel-4 group of exponent 7 is finite.

If so it has order $7^{11} = 1977326743$ and class 6.

We appear to need to enumerate the $7^6 = 117649$ cosets of a subgroup which we can prove finite.

So far the best result we have is that a quotient is finite:
$\langle a, b \mid a^7, b^7, (ab)^7, (aB)^7, \text{Engel-4}, (a, b, b, b)\rangle$ has order
$7^8 = 5764801$ and class 5.

This relies on a difficult coset enumeration (index 343, more than 100 million cosets needed).

Warnings about how hard these kinds of things can be come from various proofs for results in groups with "small" exponent.

The proof that the 2-generator Engel-4 group of exponent 5 is finite is quite difficult, and very different from what a proof for exponent 7 would look like.

The exponent 5 analogue of the index 343 enumeration is easy.

For 2-generator, exponent 6, Engel-4 we need more than 200000 cosets to do an index 6 enumeration.

New results for $B(2,6)$ show how hard computational proofs are in that context.

Hall (1956) showed that $B(2,6)$ is finite; his proof uses about 2^{124} sixth powers.

(The order of $B(2,6)$ is $2^{28}3^{25} = 227442304239437611008$.)

H, Newman, Niemeyer and Sims (1999) give a lower bound: at least 22 sixth powers are needed.

We expect that 22 is closer to the truth than 2^{124}.

So we consider some quotients of $B(2,6)$.

We consider finite presentations as groups with exponent six, i.e. $\{X \mid \mathcal{R}, \text{exp } 6\}$ with $X, \mathcal{R}$ finite.

A finite set of sixth power relations suffice in place of the exponent six condition.

We study presentations $\{X \mid \mathcal{R}, \mathcal{S}\}$ where $\mathcal{S}$ is a finite set of sixth powers.

We seek 'small', even irredundant or minimal, sets $\mathcal{S}$ for which $\langle X \mid \mathcal{R}, \mathcal{S}\rangle \cong \langle X \mid \mathcal{R}, \text{exp } 6\rangle$.

Let $C(r,s)$ denote the largest 2-generator group with exponent six generated by elements of orders r and s.

To understand $B(2,6)$ better we look at presentations for the groups $C(2,2)$, $C(2,3)$, $C(3,3)$, $C(2,6)$ and $C(3,6)$.

Let $\{a,b\}$ be a generating set for $B(2,6)$.

The subgroup $H = \langle a^{6/r}, b^{6/s} \rangle$ of $B(2,6)$ is clearly a quotient group of $C(r,s)$.

It turns out that $H \cong C(r,s)$.

The order of H and the index of the normal closure in $B(2,6)$ of $\langle a^r, b^s \rangle$ are easily computed using a polycyclic presentation for $B(2,6)$.

In each case these numbers are the same.

The presentation $\{a, b \mid a^2, b^2, (ab)^6\}$ is a minimal presentation for the group $C(2,2)$, the dihedral group of order 12.

A minimal presentation for $C(2,3)$, a group of order 216, is $\{a, b \mid a^2, b^3, (ab)^6, [a,b]^6\}$.

Therefore the first challenging case to consider is the group $C(3,3)$.

It is a group of order $2^{10}3^3 = 27648$.

From a polycyclic presentation we can deduce that this group has a presentation with 2 generators and 30 relations.

Hall-type proofs show that $C(3,3)$ can be presented as $\{a, b \mid a^3, b^3, \mathcal{S}\}$ with sets $\mathcal{S}$ consisting of about 2^{35} sixth powers.

How well can we improve on this?

$C(3,3)$ certainly needs 5 relations in view of the rank of its multiplier.

We show that a set $\mathcal{S}$ which suffices to define $C(3,3)$ needs at least 5 sixth powers (ie, 7 relations).

Both coset enumeration and rewriting show that the set of 11 "shortest" sixth powers suffices.

With coset enumeration we can reduce this to 8 sixth powers, but we have been unable to close the gap.

The coset enumerations involved require millions of cosets.

The most straightforward proof by enumeration of the 4608 cosets of a cyclic subgroup of order 6 needs a maximum of 85090780 and total of 85659443 cosets.