SOME COMPLEXITY PROBLEMS IN ALGEBRAIC COMPUTATIONS

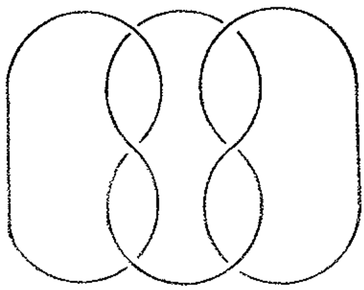
George Havas

Mathematics, IAS, Australian National University

1. Introduction

A presentation for an algebraic structure is a set of generators together with a set of defining relations amongst the generators such that every element of the structure can be expressed in terms of the generators (and the operations applicable to the structure) and such that any relation satisfied by the generators follows from the defining relations. Presentations have important theoretical and practical applications in algebra, particularly with respect to groups, but also with respect to rings, lattices, Lie algebras and other structures.

For example a reef knot:



leads in a natural way (see [5]) to an associated group G (the knot group) which can be presented thus:

$$G = \left\langle x_1, x_2, x_3, x_4, x_5, x_6 \mid x_3^{-1}x_1x_2x_1^{-1} = x_1^{-1}x_2x_4x_2^{-1} \right.$$
$$\left. = x_2^{-1}x_4x_1x_4^{-1} = x_6^{-1}x_4^{-1}x_5x_4 = x_4^{-1}x_5^{-1}x_6x_5 = x_5^{-1}x_6^{-1}x_3x_6 = 1 \right\rangle .$$

Any particular structure has an infinite number of different presentations. Certain types of presentations are particularly useful. For example desirable features may include independent generators, minimal generating sets, independent relations, minimal number of relations and short relations. On the other hand presentations with other specific features may be wanted. For example for a nilpotent group a consistent commutator power presentation may be required, for a finite group a complete multiplication table may be appropriate and, correspondingly, for any structure a complete operation table may be desired.

In this paper some computational problems encountered in the construction and application of presentations are described.

2. Concise presentations

When a presentation is being constructed, unless a special type of presentation is wanted, the primary aim is usually to obtain a concise presentation. Presentations for groups can be computed from group representations (see [2]), from topological structures and from other presentations (see [6]). In this section efforts at determining concise presentations in the latter cases are described.

The usefulness of knot groups is that they can be used to distinguish knots. Knots with distinct knot groups are different. Easy methods are available for computing presentations for knot groups. Unfortunately it is difficult to distinguish groups given by presentations (algorithmically impossible in general). To distinguish knot groups, polynomial matrices (square, of dimension equal to the number of relations, n say, in the group presentation) can be associated with the groups. The Alexander polynomial, the determinant of an $(n-1) \times (n-1)$ minor of a polynomial matrix associated with the group, is a group invariant, and polynomials are easy to distinguish, giving us a method to distinguish knots.

Computer programs have been written to input knots, and then calculate knot groups, associated matrices, Alexander polynomials and other invariants in an attempt to classify knots (see for example [7, §5]). In the program referred to the determinant calculation routine has an execution time proportional to $n^5 \ (\log_e n)^2$, where n is the dimension of the matrix.

Since the dimension of the associated matrix equals the number of relations, it is clear that presentations with minimal numbers of relations are desirable for such calculations. Look at the presentation for the group of the reef knot given in Section 1. It is a presentation easily computed (by hand or machine) but highly redundant. From the first relation we can deduce that $x_3 = x_1x_2x_1^{-1}$. We can eliminate x_3 from the presentation and keep repeating the process till we obtain a 3 generator 3 relator presentation:

$$G = \langle x_1, x_2, x_6 \mid x_6^{-1}x_1x_2^{-1}x_1^{-1}x_6^{-1}x_1x_2x_1^{-1}x_6x_1x_2x_1^{-1} = x_1^{-1}x_2x_6^{-1}x_1x_2^{-1}x_1^{-1}x_6x_1x_2x_1^{-1}x_6x_2^{-1}$$
$$= x_2^{-1}x_6^{-1}x_1x_2^{-1}x_1^{-1}x_6x_1x_2x_1^{-1}x_6x_1x_6^{-1}x_1x_2^{-1}x_1^{-1}x_6^{-1}x_1x_2x_1^{-1}x_6 = 1\rangle .$$

The elimination procedure is straightforward and takes little time or computer space, and gives a much better presentation for eventual polynomial matrix determination calculations.

The situation with some other presentations is more complex. The Reidemeister-Schreier method (see [6]) determines a presentation for a subgroup H of a group G given a presentation for G and the coset table of H in G.

The Reidemeister-Schreier presentation is usually not in a useful form. The number of Schreier generators is of the order of $n_g[G : H]$ where n_g is the number of generators of G; the number of Reidemeister relators is of the order $n_r[G : H]$ where n_r is the number of relators in the presentation of G. There are usually many redundant generators in the sense that there are many relators containing exactly one occurrence of a particular generator. Such generators can be removed by substitution.

The most obvious approach is to remove one redundant generator at a time from each relator, as in the knot theory example above. However this turns out to be very time-consuming in computer implementation. Three possible techniques are discussed below.

(a) ELIMINATION TECHNIQUE 1

Eliminate one generator at a time. Unfortunately the Reidemeister-Schreier method frequently gives us presentations with hundreds of generators and relators, and hundreds of redundancies, and this approach takes too long.

(b) ELIMINATION TECHNIQUE 2

The eliminations are batched together to save computing time. Each relator is examined for any generator occurring exactly once (provided the relator is independent of previously discovered, but not yet eliminated, redundant generators). The shorter relators are examined first so that the generators selected for elimination will tend to have short generator strings as their equivalents.

Actual elimination is only done after all relators have been examined and a batch of redundant generators found. After elimination new redundancies may appear and the process is repeated till no further redundancies are found.

(c) ELIMINATION TECHNIQUE 3

A compromise between Techniques 1 and 2 is to eliminate redundant generators with values of the same length at the one time, doing this elimination for increasing length. This has the advantage of ensuring that all length zero and one eliminations are done as soon as possible. Length zero and one eliminations are most desirable for they do not increase relator lengths, whereas higher length eliminations may well increase relator lengths, and usually do.

We might well expect Technique 1 to be superior in that we can select for elimination the generator with the shortest equivalent string at each stage so that it seems reasonable to imagine that the resulting presentation should be more concise than that given by the other techniques. Let us look at some test examples.

The following groups and subgroups (taken from [4]) were used in test program runs.

(a) $G_1 = \langle a, b \mid a^3 = b^6 = (ab)^4 = (ab^2)^4 = (ab^3)^3 = a^{-1}b^{-2}a^{-2}b^{-2}a^{-1}b^{-2}ab^2ab^2ab^2 = 1\rangle$.

(G_1 is presentation for $PSL(3, 3)$, of order 5616 .)

$H_1 = \langle a, b^2\rangle$ is the Hessian group of order 216 , $[G_1 : H_1] = 26$.

(b) $G_2 = \langle a, b \mid a^4 = b^4 = (ab)^4 = (a^{-1}b)^4 = (a^2b)^4 = (ab^2)^4$

$$= (a^2b^2)^4 = [a, b]^4 = (a^{-1}bab)^4 = 1\rangle .$$

(G_2 is a presentation for $B_{4,2}$, of order 4096 .)

$H_2 = \langle a, b^2\rangle$, $|H_2| = 64$, $[G_2 : H_2] = 64$,

(c) $G_3 = \langle a, b, c \mid a^{11} = b^5 = c^4 = (bc^2)^2 = (abc)^3 = (a^4c^2)^3 = b^2c^{-1}b^{-1}c = a^4b^{-1}a^{-1}b = 1\rangle$.

(G_3 is a presentation for M_{11} , of order 7920 .)

$H_3 = \langle a, b, c^2\rangle$ is $PSL(2, 11)$ of order 660 , $[G_3 : H_3] = 12$.

(d) $G_4 = \langle a, b, c \mid a^{11} = b^5 = c^4 = (ac)^3 = b^2c^{-1}b^{-1}c = a^4b^{-1}a^{-1}b = 1 \rangle$.

(G_4 is a better presentation for M_{11} .)

$H_4 = \langle a, b, c^2 \rangle$ is again PSL(2, 11) of order 660 , $[G_4 : H_4] = 12$.

(e) $G_5 = \langle a, b, c \mid a^3 = b^7 = c^{13} = (ab)^2 = (bc)^2 = (ca)^2 = (abc)^2 = 1 \rangle$.

(G_5 is $G^{3,7,13}$, a presentation for PSL(2, 13) , of order 1092 .)

$H_5 = \langle ab, c \rangle$ is dihedral of order 26 , $[G_5 : H_5] = 42$.

(f) $G_6 = \langle a, b, c \mid a^3 = b^7 = c^{14} = (ab)^2 = (bc)^2 = (ca)^2 = (abc)^2 = 1 \rangle$.

(G_6 is $G^{3,7,14}$, of order 2184 .)

$H_6 = \langle ab, c \rangle$ is dihedral of order 28 , $[G_6 : H_6] = 78$.

In fact it is easy to read presentations for H_5 and H_6 off from the presentations for G_5 and G_6 , but it is still interesting to observe the behaviour of the techniques in these cases.

In Table 1, we list properties of G_i and H_i , where:

column (a) indicates the number of generators in the presentation;

column (b) indicates the number of relators; and

column (c) indicates the length of the longest relator.

TABLE 1

Subgroup	Index $[G_i : H_i]$	Presentation for G_i			Original Reidemeister-Schreier Presentation for H_i		
		a	b	c	a	b	c
H_1	26	2	6	19	27	156	14
H_2	64	2	9	16	65	576	16
H_3	12	3	8	18	25	96	15
H_4	12	3	6	11	25	72	11
H_5	42	3	7	13	85	294	13
H_6	78	3	7	14	157	546	14

Using the elimination techniques described above on these presentations we obtain the results listed in Table 2.

TABLE 2

Subgroup	Technique 1			Technique 2			Technique 3		
	a	b	c	a	b	c	a	b	c
H_1	3	37	114	3	35	80	3	37	114
H_2	2	76	160	2	85	264	2	81	132
H_3	3	19	42	4	27	157	4	25	115
H_4	3	11	24	3	13	84	3	11	24
H_5	3	13	134	3	12	182	3	13	142
H_6	3	16	80	3	16	56	3	16	80

We might say one presentation is better than another, if it has fewer generators, fewer relators and/or shorter relators. The table indicates that sometimes the heuristically better technique produces a "worse" presentation.

Sample timings for each of the three techniques are as follows. For H_1 Technique 3 took 1.5 times as long as Technique 2 while Technique 1 took 2.4 times as long. For H_5 Technique 3 took 1.6 times as long as Technique 2 while Technique 1 took 8.5 times as long. These timing considerations justify the selection of Technique 2 for program implementation.

With the aid of further presentation simplification techniques, presentations of the kind shown in Table 3 have been obtained using elimination Technique 2.

TABLE 3

Subgroup	Final program presentation		
	a	b	c
H_1	2	29	168
H_2	2	67	204
H_3	4	22	111
H_4	3	10	21
H_5	3	13	65
H_6	2	12	66

As first sight, it may seem that none of these presentations is of much use. The original Reidemeister-Schreier presentation has too many generators and too many relators while even the final presentation has relators which are too long, and perhaps too many relators. But looking at the final output presentation shows us otherwise. Of the examples given, the presentation for H_2 looks the worst. However, the first four relations in the presentation produced for H_2 are $b^2 = a^4 = (ab)^4 = (a^2b)^4 = 1$ and these alone suffice for a presentation of H_2. In a similar fashion three of the first four relations of the presentation produced for H_6 are $a^2 = (ab)^2 = b^{14} = 1$, indeed a presentation of H_6.

If we can select a likely presentation for H from the relators, it is easy, using coset enumeration, to determine whether the remaining relators are consequences of the selected set. So even when we get an apparently ungainly presentation form the Reidemeister-Schreier program we may well be able to extract a useful presentation from it as long as we have a small enough generating set.

The final programmed algorithm, including the coset enumeration, took 6.5 seconds CPU time on a CDC 6600 to find the presentation for H_1.

3. Specific presentations

Some specific presentations, such as commutator power presentations and operation tables are not concise. This leads to problems in computer representation and manipulation of these presentations.

Different styles of computer representation have been successfully used (for example, directory accessed arrays for commutator power presentations [1], list structures for Lie algebra operations tables [7, §6]). No comparative studies have as yet been made to determine optimal computer representations for these presentations. The importance of these considerations is illustrated by the increased range of the presentation-based Todd-Coxeter algorithm achieved partly by virtue of use of a linked list structure instead of a simple array structure (see [3]). This improvement yields significant reductions in execution time requirements and facilitates storage allocation. Use of simple and circular lists does the same for the Reidemeister-Schreier program.

The computation of a specific commutator power presentation or a specific Lie algebra operation table involves the construction of a presentation for a super-structure (which has the desired structure as an epimorphic image) followed by reduction to a presentation of the desired structure. In the reduction process, redundant generators and suitable eliminating relations are discovered and the redundant generators are removed from the presentation.

Both the discovery of the redundant generators and their elimination are complex processes. As regards relation discovery, it is important to keep the size of the super-structure as small as possible (to minimize the number of redundancies) and it is important to find the relations as simply as possible. These things are dependent on the actual problem in hand and the details of the mathematical algorithm, so are not discussed here.

The elimination of the redundant generators from the presentation is a problem resembling the elimination of redundant generators from Reidemeister-Schreier presentations discussed in Section 2. John Wamsley reports that in the computation of the consistent commutator power presentation for $\overline{B}(5, 2)$ (see [8]), at one stage some 1000 relations were to be discovered and eliminated. He found that the optimal batch size for elimination was about 200. On either doubling or halving the batch size he found that the total execution time approximately doubled (and the memory space required increased). Imagine how terrible it would be to use a batch size of 1 or 1000.

The same kind of problem arises in the computation of Lie algebra operation tables. Using a one by one elimination technique leads to the manipulation of intermediate presentations requiring more then twice the storage space needed by the final presentation. Using batched eliminations we require hardly any additional storage space at any intermediate stage (including space for the presentation and for the batched relations) than we do at the final stage. Also the execution time is much less.

4. Conclusions

In the application of computers to pure mathematics, consideration must be given on the first hand to the mathematical algorithms to be used. In addition, because of limitations on computer time and space, careful thought must be given to such computational matters as data representation and technical details of algorithm implementation.

References

[1] Bayes, A.J., Kautsky, J. and Wamsley, J.W. Computation in nilpotent groups (application). In *Proc. Second Internat. Conf. Theory of Groups*, Canberra, 1973. Lecture Notes in Mathematics, 372, 82-89. Springer-Verlag, Berlin, Heidelberg, New York, 1974.

[2] Cannon, John J. Construction of defining relators for finite groups. *Discrete Math.* 5, 105-129, 1973.

[3] Cannon, John J., Dimino, Lucien A., Havas, George and Watson, Jane M. Implementation and analysis of the Todd-Coxeter algorithm. *Math. Comp.* 27, 463-490, 1973.

[4] Coxeter, H.S.M. and Moser, W.O.J. *Generators and Relations for Discrete Groups.* Ergebnisse der Mathematik und ihrer Grenzgebiete, 14. Springer-Verlag, Berlin, Heidelberg, New York, 1957, 1965, 1972.

[5] Crowell, R.H. and Fox, R.H. *Introduction to Knot Theory.* Ginn, 1963.

[6] Havas, George. A Reidemeister-Schreier program. In *Proc. Second Internat. Conf. Theory of Groups*, Canberra, 1973. Lecture Notes in Mathematics, 372, 347-356. Springer-Verlag, Berlin, Heidelberg, New York, 1974.

[7] Havas, George. Computational approaches to combinatorial group theory. PhD thesis, University of Sydney, Sydney, 1974.

[8] Havas, George, Wall, G.E. and Wamsley, J.W. The two generator restricted Burnside group of exponent five. *Bull. Austral. Math. Soc.* 10, 459-470, 1974.