Experiments in coset enumeration

George Havas and Colin Ramsay *

Abstract. Coset enumeration, based on the methods described by Todd and Coxeter, is one of the most important tools for investigating finitely presented groups. Such methods do not, in general, constitute an algorithm. Each problem has to be addressed individually, and determining whether or not an acceptable solution for a difficult problem can be found using given resources requires an empirical approach (i.e., experimentation). We discuss some of the ideas involved, and illustrate with examples which emphasize some of the possibilities.

1991 Mathematics Subject Classification: primary 20F05; secondary 20-04.

1. Introduction

Coset enumeration is long established as a technique for the investigation of finitely presented groups. It was used well before the days of electronic computers, apparently first by Moore [14], and was later popularised by Todd and Coxeter [17]. The first computer implementation was that of Haselgrove in 1953. This, along with other early implementations, is described by Leech [10]. Detailed accounts of the techniques used in coset enumeration can be found in [1, 6, 12, 15, 16], together with some experimental results.

Coset enumeration takes a finitely presented group and a finitely generated subgroup as input, and attempts to find the index of the subgroup in the whole group. In principle, it will succeed whenever this index is finite. However, the computational complexity of coset enumeration is not well-understood. Even using a fixed strategy, there is no computable bound on the number of cosets which need to be defined to obtain a complete coset table (in terms of the length of the input and a hypothetical index). Further, Sims [16] has proved that there is no polynomial bound, in terms of the maximum number of cosets, for the number of coset tables which can be derived by coset table operations such as those used in enumeration programmes.

*The work of both authors was supported by the Australian Research Council. We are grateful to The University of Queensland's High Performance Computing Unit for access to the Origin 2000 supercomputer.

Sims' result implies that the running time of coset enumeration procedures, in terms of the space available, may be very high. Pathological examples of, for example, the trivial group exist which exhibit such behaviour. However, the main factor determining whether or not we can succeed in completing an enumeration is the amount of memory available; i.e., the number of cosets we can define. So the maximum and total number of cosets defined during an enumeration are often used as metrics, with smaller values being regarded as 'better'. Usually, the running time is proportional to the total number of cosets, and is not a limiting factor. There is generally a close correlation between maximum and total cosets, so we choose to concentrate on only one, total cosets, in this paper.

One contributing factor to our less than perfect understanding of coset enumeration is the freedom of choice of which coset to define next that we have during an enumeration. This freedom implies that the space of possible enumeration sequences, for a given coset enumeration problem, is extremely large. It is not clear how best to generate an enumeration which is space efficient, or from which an understandable formal proof can be extracted. In this paper we discuss how features of a modern implementation of coset enumeration can be used to explore the enumeration space and to gain insight into this challenge.

The coset enumeration implementation which we used was the one by Havas and Ramsay [8], which is an enhanced version of the one described in [6]. The enumerations were performed on an SGI Origin 2000 computer which was equipped with enough memory to allow the definition of some hundreds of millions of cosets when necessary.

Our enumerator has a large number of parameters, with a wide choice of settings; we discuss the ones relevant to our examples. The most important are `CT` and `RT`, which indicate the relative weightings attached to defining new cosets using the coset table (traditionally called the Felsch strategy) and the relator tables (traditionally known as the HLT strategy). Each unit in `CT` indicates one definition, which is tested in all essentially different positions in all the relators until all consequent deductions and coincidences have been found and processed. Each unit in `RT` indicates one coset being scanned at all relators, with definitions made as necessary to close all scans. If both `CT` and `RT` are non-zero, then `RT` definitions and deductions are stacked, and tested at the start of the next `CT`-phase (so 'lookahead' is performed after each `RT` phase).

The `Mend` parameter controls a technique due to Mendelsohn [13], wherein `RT`-style scanning is performed using all cyclic shifts of the relators; it can be set on or off. It is occasionally very beneficial in reducing maximum and total cosets, including enumerations for one of our examples.

As discussed in [6], some number of the relators can be treated as subgroup generators, and scanned and closed at coset 1 (i.e., the subgroup) at the start of the enumeration. This is controlled by the `No` parameter; we will always use all or none of the relators, so we will say that `No` is on or off.

Also discussed in [6] is the idea of preferentially filling any gaps of length one noted when scanning relators, subject to a 'fill-factor' to ensure that some constant proportion of the coset table is always kept completed. Our programme implements this idea and allows gaps to be queued for later use (as in [6]), filled immediately, or filled immediately and the consequent deduction made immediately. We will say that `Fi` is on or off and, if on, explicitly state which fill mechanism we used. Unless stated otherwise, the default fill-factor will be used.

2. Can we do it?

Consider the group $W = \langle x, y \mid x^3y^4x^5y^7 = x^2y^3x^7y^8 = 1\rangle$. In 1996, D.L. Johnson [3] reported the failure of coset enumeration to answer a question of Malcolm Wicks in 1987; namely, is W the cyclic group of order 11, C_{11}? More recently, Havas, Holt, Kenne and Rees [7] have shown that $W \cong C_{11}$ by combining the Knuth-Bendix process (originally described in [9]) with the Todd-Coxeter process. They also note that the single coset of $\langle y\rangle$ (which generates W) and the eleven cosets of $\langle x\rangle$ (which is trivial) can be enumerated in, respectively, 22 million and 14 million cosets using coset enumeration alone.

These results leave unanswered the question of whether or not we can enumerate the cosets of $\langle 1\rangle$ in W directly. The number of cosets required for an enumeration tends to grow as the size of the subgroup shrinks. (Note that the result quoted above, where enumeration over $\langle x\rangle$ takes fewer cosets than over $\langle y\rangle$, is not typical.) Experience suggests that parameter combinations which work well for one subgroup often work well for other subgroups of the same group. This suggests that we should consider selecting parameters for $W/\langle 1\rangle$ on the basis of those that work well for $W/\langle x\rangle$ or $W/\langle y\rangle$. (For brevity here we denote the enumeration of the cosets of the group G over the subgroup H by G/H.)

To test this idea, we performed each of the enumerations $W/\langle x\rangle$ and $W/\langle y\rangle$ 121 times, varying the `CT` and `RT` parameters exponentially from 2^0, 2^1 to 2^{10}. This gives good coverage of the parameter space in a reasonably small number of examples, although, of course, we have no easy way of knowing whether or not we have missed any 'interesting' combinations. Both `Mend` and `No` were set on, `Fi` was set off, and the enumerator given enough memory for 50 million cosets (i.e., 800MB of memory). The total numbers of cosets defined (in thousands, to the nearest thousand) are given in Tables 1 and 2 (for brevity, omitting `RT` values 512 and 1024, which are not significantly different to numbers obtained for RT $= 256$); "o/f" means that the enumeration attempt overflowed the available space.

Table 1: Total cosets counts (thousands) for $W/\langle x\rangle$

	RT								
CT	1	2	4	8	16	32	64	128	256
1	14262	14225	14227	14229	14232	14234	14241	14246	14251
2	14671	14271	14237	14229	14232	14234	14241	14246	14250
4	15611	14681	14308	14240	14232	14234	14241	14246	14251
8	14318	15622	14713	14312	14233	14235	14241	14247	14251
16	11458	14441	15706	14703	14244	14235	14242	14247	14251
32	8072	10059	14379	15677	14681	14424	14416	14427	14432
64	11402	6864	7895	11945	15553	14939	14681	14682	14688
128	24958	13012	6925	7877	11992	16070	15435	15180	15171
256	35621	24934	12928	8879	8632	12519	17295	16631	16343
512	o/f	34681	35009	18365	12691	10645	15464	19101	19122
1024	o/f	o/f	40411	47665	30264	20811	18162	16159	15137

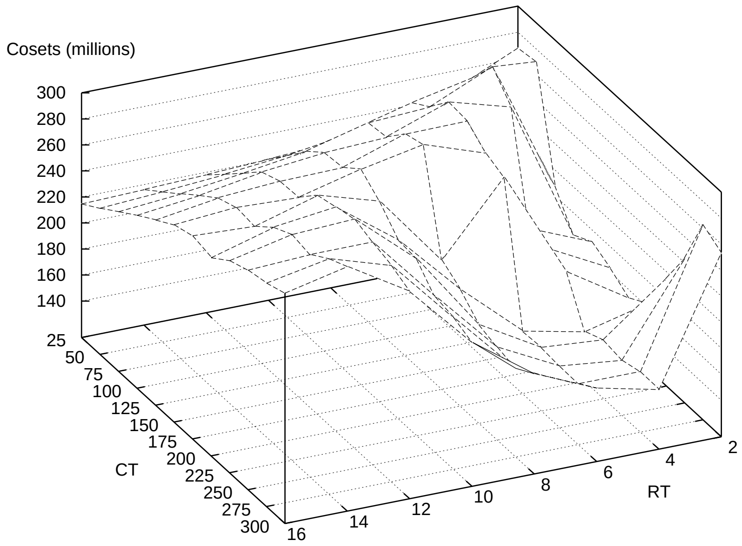
Table 2: Total cosets counts (thousands) for $W/\langle y\rangle$

	RT								
CT	1	2	4	8	16	32	64	128	256
1	30172	30093	30071	30063	30068	30072	30085	30098	30108
2	28214	30191	30099	30076	30068	30072	30085	30098	30108
4	26263	28222	30204	30105	30068	30072	30084	30097	30107
8	29179	26319	28234	30212	30068	30072	30084	30098	30107
16	30984	29330	26290	28372	30165	30074	30083	30097	30106
32	31022	31087	29207	26212	28310	30094	30084	30097	30107
64	29382	31176	34018	28926	26511	28271	30260	30260	30282
128	40451	29472	27616	35021	29543	26131	28075	30593	30581
256	o/f	36678	26261	22981	26492	28962	26836	27136	31218
512	o/f	o/f	31861	25873	21282	16871	14864	16216	27471
1024	o/f	o/f	o/f	37749	27671	24853	23903	21016	20072

These results highlight two parameter combinations for further investigation; namely, the `CT:64; RT:2` enumeration for $W/\langle x\rangle$, and the `CT:512; RT:64` enumeration for $W/\langle y\rangle$.

For the enumeration $W/\langle 1\rangle$, we started by varying CT from 25 to 300 in steps of 25 and RT from 2 to 16 in steps of 2. The other parameters were as above, and the enumerator was given enough memory for 300 million cosets (i.e., 4800MB of memory). All 96 of these enumerations completed successfully; the results are plotted in Figure 1. More detailed investigations, varying CT and RT in steps of 1, revealed that the `CT:185; RT:4` enumeration is a local minimum, requiring 139 330 097 cosets. A similar search of the area around `CT:512; RT:64` revealed another local minimum at `CT:761; RT:59`, requiring 99 222 961 cosets.

Figure 1: Total cosets counts for $W/\langle 1\rangle$



So, in this case at least, we see that we can use an 'easy' enumeration to guide us as to how best to do a more difficult one. Although our tests were done on a machine with adequate memory, note that, if we had been working on a machine which allowed the definition of only, say, 150 million cosets, the technique discussed would have been crucial to success.

Despite what Figure 1 might suggest, in general the number of cosets required is not a 'smooth' function of the parameters. Figure 2 of the next section witnesses this on a much smaller scale. Additionally, despite being guided to good enumerations of $W/\langle 1\rangle$ by examining those for $W/\langle x\rangle$ and $W/\langle y\rangle$, the ratios of the number of cosets required for these three enumerations, for the same parameters, is very variable.

Table 3: Variations between the three enumerations

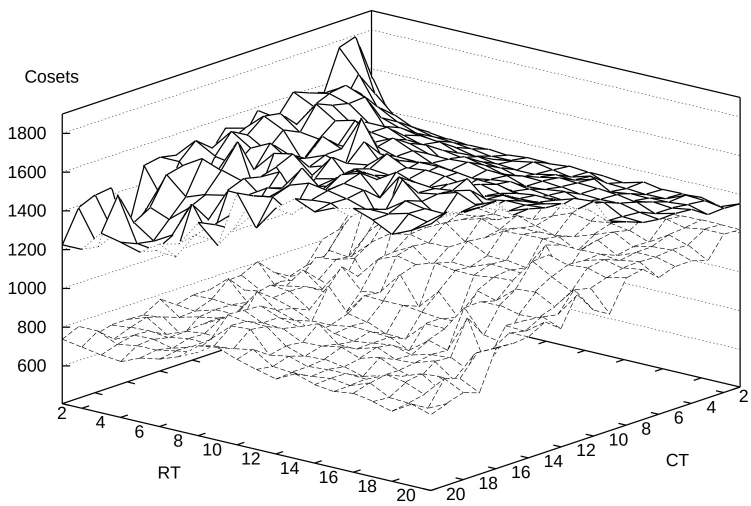
		$W/\langle y\rangle$		$W/\langle x\rangle$		$W/\langle 1\rangle$
`CT`	`RT`	Total	Normalised	Total	Normalised	Total
2	128	30098364	0.142	14246404	0.067	212011726
185	4	28331117	0.203	9096142	0.065	139330097
760	59	19357727	0.112	19305998	0.112	172843398
761	59	19379546	0.195	19283983	0.194	99222961

These points are illustrated in Table 3, where we list the total cosets required (both actual and normalised to the count for $W/\langle 1\rangle$) for the minimum parameters noted. We also include data for a parameter combination adjacent to one of the minima, and for a point in the 'flat' region where `CT` is low and `RT` high. Finally, it is important to note that, despite our success, we do not know whether or not we have, in fact, found the best possible parameter combination.

3. How well can we do it?

Consider the group $(8,7 \mid 2,3) = \langle a,b \mid a^8 = b^7 = (ab)^2 = (a^{-1}b)^3 = 1\rangle$. This has order 10752, so the octahedral subgroup $\langle a^2, a^{-1}b\rangle$ has index 448 in it. Leech [11] notes that this enumeration is "far from straightforward", seemingly due to the difficulty of establishing that the given relations imply $(a^2b^4)^6 = 1$. This enumeration is one of those discussed by Cannon, Dimino, Havas and Watson [1] and by Havas [6]; the best enumeration listed there required the definition of 840 cosets.

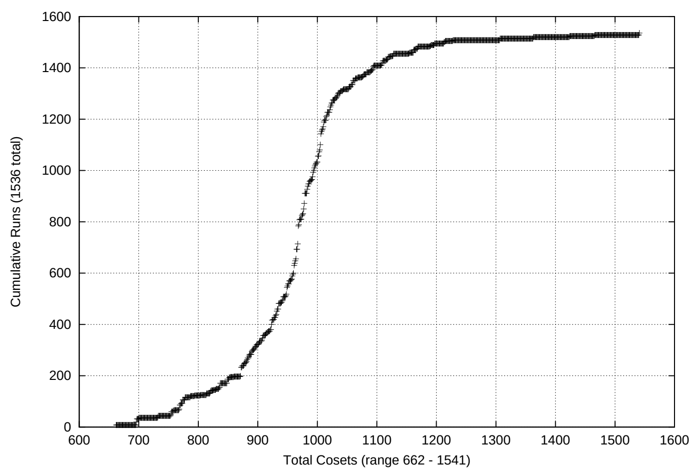
Figure 2: Minimum and maximum total cosets for $(8,7 \mid 2,3)/\langle a^2, a^{-1}b\rangle$



The effect of permuting the order of the relators and of rotating the relators was discussed in [1], where it was shown that this can yield large variations in the number of cosets required in an enumeration. Our enumerator has the ability to generate and use all possible relator orderings, relator rotations, and relator inversions. For $(8,7 \mid 2,3)$, this amounts to a total of $(4!)(1.1.2.2)(2^4) = 1536$ possibilities. For each combination of `CT` and `RT` from 1 to 25 in steps of 1, we tested all of these possibilities and recorded the maximum and minimum values of the total number of cosets defined. `Mend` was set off, and `No` and `Fi` were set on, with fill definitions and their consequent deductions being made immediately. The results, for `CT` and `RT` up to 20, are plotted in Figure 2.

The parameters `CT:15; RT:3` yield the smallest minimum number of cosets, namely 662. This is an appreciable reduction on the 840 previously achieved, almost halving the number of redundant cosets. Interestingly, parameter combinations which yield short enumerations also seem to yield long enumerations as the presentation is varied. This suggests that, in the general case, an unfortunate choice of presentation can lead one badly astray when searching for the best enumeration. Note that `CT` low with `RT` high strategies, where most of the definitions are made in closing the relators, are conservative, in that, although they do not yield short enumerations, they do not yield overlong ones either. The minimum range noted for one parameter combination for the number of cosets was 1220–1313 for `CT:1; RT:19`, and the maximum was 796–1879 for `CT:19; RT:23`.

Figure 3: Distribution of total cosets for $(8,7 \mid 2,3)/\langle a^2, a^{-1}b\rangle$



For the `CT:15; RT:3` parameters (which give the smallest minimum), the presentation as given needs 1004 cosets, while if the relators are sorted in length-increasing order only 965 cosets are needed. For these parameters the number of cosets needed for all 1536 equivalent presentations was also recorded, and the resulting distribution is shown in Figure 3. This distribution is highly skewed, with 67.3% of the runs taking 1000 or less cosets and 91.7% taking 1100 or less. Thus, randomly changing the initially given presentation into one of its equivalents has a better than 2 in 3 chance of yielding a superior enumeration.

4. Can we prove it?

Consider the Fibonacci group $F(2,7) =$

$$\langle a,b,c,d,e,f,g \mid ab=c,\ bc=d,\ cd=e,\ de=f,\ ef=g,\ fg=a,\ ga=b\rangle.$$

The first formal proof that $F(2,7) \cong C_{29}$, the cyclic group of order 29, was given by Havas [5], and was derived from the workings of an index 1 enumeration over one of the generators. This enumeration used a total of 327 cosets, and the proof extended over several pages. It is reasonable to expect that the better the coset enumeration, the better any extracted proof. This focuses attention on trying to do even relatively easy enumerations, such as this, as well as possible. Leech [12] reports on a hand enumeration which took 129 cosets, and was then pruned to 55 cosets. (This was reduced to 53 by Edeson [2]. Interesting new work on pruning by Felsch, Hippe and Neubüser [4] includes the result that this enumeration can be done in 50 cosets.) Leech's enumeration yields a shorter (but no less impenetrable) proof, and Leech remarks that 'the initial hand number of 129 will serve as a target for "intelligent" computer algorithms'. The enumerator discussed by Havas [6] achieved a total of 169.

For $F(2,7)$, direct `RT`-style definitions do not seem beneficial, so our tests were done with the parameters `CT:1000; RT:0`. We also set `Mend` off and `No` on. Since all the relators of $F(2,7)$ have length three, gaps of length one will be common, so we set `Fi` on. However, the effects of `Fi` are very variable for $F(2,7)$, so we do not use the default value of the fill-factor. Instead, we regard the fill-factor as a useful means of exploring the space of possible enumerations; values in the range 10–30 were used.

Another useful source of variation, particularly effective with `CT`-style definitions, is the order of the columns (i.e., the generators) in the coset table. For our programme, this depends upon the order in which the generators are listed to the enumerator, and we have 7! possibilities for $F(2,7)$. Due to the symmetry of the presentation for $F(2,7)$, we can perform the enumeration over any generator; so there are 7 possibilities here.

Combining all these different possibilities with the $7!2^7 3^7$ different equivalent presentations (which can be obtained by permuting the order of relators, formally inverting relators and rotating relators) gives a very large number of enumerations; far too many to exhaustively test. To sample these, we select possible enumerations at random, and use the selected presentation as a starting point in a test of equivalent presentations. We allow each test to run for up to a few days before terminating it and starting another run. To maximise throughput, we give the enumerator only enough memory for, say, 150 cosets. Most enumerations quickly overflow and abort, allowing another to be started; this technique allowed approximately 1000 enumerations to be tested per second.

Over the course of 21 runs, a total of 3710×10^6 enumerations were performed. Of these, 5245 beat Leech's target, with 152 taking less than 120 cosets. Our best result, of 112 cosets, betters the target comfortably.

Of course, these enumerations can be manually pruned. In fact, we can improve the computer results slightly by specifying what the first few definitions are to be. (This is readily done by adding words which are formally trivial to the subgroup generators.) This 'encourages' the enumerator to process particular words; suitable words are chosen by examining the output of previous tests, and seeing which coincidence precipitated the final collapse. Using this technique, our best 'computer' result is 109 cosets.

5. Conclusions

The examples we have discussed indicate that, while we can only explore a very small proportion of the space of possible enumeration sequences, this exploration is worthwhile. It can enable us to complete an enumeration where we would otherwise fail, and it can yield dramatically shorter enumerations, perhaps leading to more insightful formal proofs. It is difficult to give general guidelines on how best to handle a particular enumeration, since cases seem to be highly individualistic. However, the techniques described here should certainly be considered when confronted with any new, difficult problem.

References

[1] John J. Cannon, Lucien A. Dimino, George Havas, and Jane M. Watson. Implementation and analysis of the Todd-Coxeter algorithm. *Mathematics of Computation*, 27(123):463–490, July 1973.

[2] Margaret Edeson. Investigations in coset enumeration. Master's thesis, Canberra College of Advanced Education, 1989.

[3] Group-pub-forum e-mail list (`group-pub-forum@maths.bath.ac.uk`), Problem 11, 1996. See `http://www2.bath.ac.uk/~masgcs/problem/problem11.htm`.

[4] Volkmar Felsch, Ludger Hippe and Joachim Neubüser. Interactive Todd-Coxeter manual. Available among the Share Packages at `http://www-gap.dcs.st-and.ac.uk/~gap`.

[5] George Havas. Computer aided determination of a Fibonacci group. *Bulletin of the Australian Mathematical Society*, 15:297–305, 1976.

[6] George Havas. Coset enumeration strategies. In Stephen M. Watt, editor, *ISSAC'91 (Proceedings of the 1991 International Symposium on Symbolic and Algebraic Computation)*, pages 191–199. ACM Press, 1991.

[7] George Havas, Derek F. Holt, P.E. Kenne, and Sarah Rees. Some challenging group presentations. *Journal of the Australian Mathematical Society (Series A)*, 67:206–213, 1999.

[8] George Havas and Colin Ramsay. Coset enumeration: ACE version 3, 1999. Available as `http://www.csee.uq.edu.au/~havas/ace3.tar.gz`.

[9] D.E. Knuth and P.B. Bendix. Simple word problems in elementary algebra. In *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

[10] J. Leech. Coset enumeration on digital computers. *Proceedings of the Cambridge Philosophical Society*, 59:257–267, 1963.

[11] John Leech. Computer proof of relations in groups. In Michael P.J. Curran, editor, *Topics in Group Theory and Computation*, pages 38–61. Academic Press, 1977.

[12] John Leech. Coset enumeration. In Michael D. Atkinson, editor, *Computational Group Theory*, pages 3–18. Academic Press, 1984.

[13] N.S. Mendelsohn. An algorithmic solution for a word problem in group theory. *Canadian Journal of Mathematics*, 16:509–516, 1964. Corrigendum: *Ibid.*, 17:505, 1965.

[14] E.H. Moore. Concerning the abstract groups of order $k!$ and $\frac{1}{2}k!$ holohedrically isomorphic with the symmetric and the alternating substitution-groups on k letters. *Proceedings of the London Mathematical Society (1)*, 28:357–366, 1897.

[15] J. Neubüser. An elementary introduction to coset-table methods in computational group theory. In *Groups – St. Andrews 1981*, London Mathematical Society Lecture Note Series 71, pages 1–45. Cambridge University Press, 1982.

[16] Charles C. Sims. *Computation with finitely presented groups.* Cambridge University Press, 1994.

[17] J.A. Todd and H.S.M. Coxeter. A practical method for enumerating cosets of finite abstract groups. *Proceedings of the Edinburgh Mathematical Society*, 5:26–34, 1936.

Centre for Discrete Mathematics and Computing,
Department of Computer Science and Electrical Engineering,
The University of Queensland, Queensland 4072, Australia.
Email: {`havas,cram`}`@csee.uq.edu.au`