Chapter 1. Introduction

Computational group theory is primarily concerned with algorithms which investigate the structure of groups. This includes the development of algorithms, their proof of correctness, an analysis of their complexity, implementation of the algorithms, and determining the limits of their application in practice. The field is also concerned with the design of software systems which incorporate the algorithms, and allow their application to problems. This involves the design of a user language; the implementation of an interpreter or compiler, and the provision of decision procedures that choose appropriate default algorithms. These systems and algorithms assist in mathematical research in a wide range of disciplines including theoretical physics, topology, combinatorics, and group theory. The systems are also examples of computer-aided instruction, and embody a wealth of knowledge in the algorithms, and in various libraries of examples.

The field poses many challenging problems. Some of the tasks are provably undecidable. Others are known to be hard in the complexity sense. For many, their status is simply unknown, and any algorithm for the task is a breakthrough. The software problems are also very challenging. Each algorithm may require six months work to implement effectively, to study its performance, and to tune it. The engineering problems of large software systems have to be overcome if the system is to be robust, portable, maintainable, and rapidly modified in the light of new breakthroughs in algorithms and data structures. Yet despite these problems, computational group theory has been a very successful branch of computer algebra, and knowledge based systems.

Over the last decade, and certainly since the last major survey of Cannon(1969), there has been enormous development in the field of computational group theory. Here we bring the reader up to date with the major representations of group-theoretical information and some principles behind the design of algorithms which manipulate them. Systems for computational group theory are briefly surveyed, and some recent applications are presented.

The best general introductions to the field are the proceedings of the 1982 symposium at Durham edited by Atkinson(1984), and the survey of coset table algorithms by Neubüser(1982). An extensive and concise list of known algorithms is provided by Neubüser's article in Buchberger, Collins, and Loos(1982), while Felsch(1978), in its current version, is a complete bibliography of the field.

Machine Representation of Groups

Computation with groups demands that the elements of the groups, and the groups as a whole be represented in the computer by some data structure. Concrete representations of elements, for example, by permutations and matrices are preferred to abstract representations such as symbolic products of generators. The abstract representations often require a deductive approach to infer properties of elements, which is open to problems of undecidability and inefficiency. The critical operations for any representation of elements are multiplication of elements, and

determination of equality of two elements.

The representation of the group has a whole must address the following concerns.

Is the representation of the group compact (that is, space efficient)?

Is it easy (or at least possible) to compute the representation?

Is there a compact representation of an individual element?

Is membership in the group and/or subgroups easily determined?

Is it easy to determine the order of the group, and to enumerate its elements?

How easy is it to construct subgroups, quotients, images and preimages of homomorphisms?

The description of the group in the first place must be finite, so computational group theory is only concerned with *finitely generated* groups.

The most common descriptions of groups use

permutations,

matrices,

power-commutator-presentations, and

finite presentations.

The first three provide concrete representations, while the last is abstract. Other machine representations are lists of elements, coset tables, and character tables. A coset table (when it can be computed) provides a concrete representation for groups given by finite presentations.

The effectiveness of algorithms varies from one representation to another. Hence, it is useful to be able to convert between representations whenever possible. This allows an appropriate representation to be chosen for each problem at hand.

Many of the algorithms for group theory rely on the *divide-and-conquer* paradigm. The problem for a group G is reduced to a problem about a subgroup H, or a quotient Q of G. The reductions can be applied recursively (or iteratively) when a *chain* of subgroups

$$G=G(1) \geq G(2) \geq \cdots \geq G(m+1)=\{\mathit{identity}\}$$

or normal subgroups, respectively, are known. Homomorphisms are another way in which quotients can be obtained for the reduction.

A major benefit of the three concrete representations of groups is that there is naturally a chain of subgroups (in the case of permutations or matrices) or normal subgroups (in the case of power-commutator-presentations).

A study of the divide-and-conquer paradigm in group theory by Butler(1986) also demonstrates the important role of algorithms for

subgroup construction,

coset enumeration, and

homomorphisms,

as building blocks for algorithms employing the paradigm.

Systems

The Cayley system, developed at the University of Sydney by John Cannon(1984), operates at over 150 sites in 21 countries. It is *the* system for computational group theory. The user language has Pascal control structures; facilities for defining algebraic objects; sets, sequences, and mappings of algebraic objects; and over 300 in-built algebraic functions. It handles those algebraic objects needed to define groups, or on which groups act, such as rings of integers, finite fields, vector spaces, modules, matrix rings, groups, polynomial rings, and incidence structures. Cayley allows all the representations of the previous sections, and has an extensive collection of the known algorithms. The system was originally developed in Fortran, but was ported to C in 1987. There are approximately 300,000 lines of code. The system has required about 45 man-years effort to date.

The system CAS was developed in Aachen by Joachim Neubüser and his colleagues(1984) for computations involving group characters. The system supports individual characters, partially defined character tables, complete character tables, arbitrary precision integers, and irrationalities. There are many built-in methods for generating complete character tables (including generic formulae of some Chevalley groups, and a library for the sporadic simple groups), generating individual characters, testing orthogonality relations, computing structure constants, and handling modular characters. The system was originally developed in Fortran, but has now been ported to C.

The SOGOS system was also developed at Aachen, see Laue et al(1984). This system supports soluble groups and p-groups represented by power-commutator-presentations. Many of the first implementations of algorithms manipulating pcp's were part of SOGOS. The system is implemented in Fortran and has a user interface similar to Cayley's.

There is also a Pascal system, CAMAC2 (Leon, 1984b), under development for the investigation of permutation groups and combinatorial objects.

Some Applications

Within mathematics, groups are ubiquitous. Applications arise in geometry, topology, combinatorial theory, number theory (especially algebraic number theory), and, of course, group theory. Space permits only a few examples. Several large simple groups were constructed by computer, most notably, the "Baby Monster" as a permutation group of degree 13 571 955 000 and order 2^{41} 3^{13} 5^{6} 7^{2} 11 13 17 19 23 31 47 by Leon and Sims(1977), and Janko's group of order 2^{21} 3^{5} 5 7 11^{3} 23 31 37 43 as a 112-dimensional matrix group over $GF(2)$ by Norton(1980) and others at Cambridge. Knots have been classified with the aid of computer (see Havas and Kovács,1984), as have certain 3-manifolds (by Richardson and

Rubenstein), *p*-groups (Ascione, Havas, and Leedham-Green, 1977), and crystallographic groups (Brown, Bülow, Neubüser, Wondratschek, and Zassenhaus, 1978) of interest in X-ray diffraction.

The Cayley system is widely used as a teaching aide. Its role at Sydney is described by Cannon and Richardson(1984/5).

The techniques for permutation groups are used in the programs of B.D. McKay(1981) and W. Kocay(1984), which are the cutting edge of practical general graph isomorphism programs. Leon(1979, 1982, 1984a) applies these techniques to determine automorphism groups of error correcting codes and Hadamard matrices, and also in their enumeration. Groups are often used in the construction and investigation of combinatorial objects and their extensions, for example, Magliveras and Leavitt(1984), Leon(1984a).

The period 1979-1981 saw major advances in the theoretical complexity of the graph isomorphism problem by Babai(1979), Luks(1980), and Hoffman(1982) which were based on group theoretic algorithms. Graph isomorphism is polynomial for graphs of bounded valence. Their work made the complexity of group theoretic problems fashionable, see Furst, Hopcroft, and Luks(1979), Even and Goldreich(1981), Babai, Kantor, and Luks(1983), Babai and Szemeredi(1984).

Magliveras, Oberg, and Surkan(1987) use "logarithmic signatures" of permutation groups - essentially a base and strong generating set - as a means of encrypting data, and as a highly effective random number generator.

Bibliographical Remarks

ASCIONE, J.A., HAVAS, G., and LEEDHAM-GREEN, C.R. (1977): A computer-aided classification of certain groups of prime power order, *Bull. Aust. Math. Soc.,* **17,** pp. 257-274, 317-320, microfiche supplement.

ATKINSON, M.D. (ed.) (1984): *Computational Group Theory* (Proc. LMS Symp. on Computational Group Theory, Durham, July 30-August 9, 1982), Academic Press, London.

BABAI, L. (1979): Monte Carlo algorithms in graph isomorphism testing, manuscript.

BABAI, L., KANTOR, W., and LUKS, E. (1983): Computational complexity and the classification of finite simple groups, *Proc. 24th IEEE Foundations of Computer Science,* pp.162-171.

BABAI, L., and SZEMEREDI, R. (1984): On the complexity of matrix group problems I, *Proc. 25th IEEE Foundations of Computer Science,* pp.229-240.

BROWN, H., BÜLOW, R., NEUBÜSER, J., WONDRATSCHEK, H., and ZASSENHAUS, H. (1978): *Crystallographic Groups of Four-Dimensional Space,* Wiley, New York.

BUCHBERGER, B., COLLINS, G.E., and LOOS, R. (1982): *Computer Algebra : Symbolic and Algebraic Computation,* Springer-Verlag, Wien.

BUTLER, G. (1986): Divide-and-conquer in computational group theory, **SYMSAC '86** (Proceedings of the 1986 ACM Symposium on Symbolic and Algebraic Computation, Waterloo, July 21-23, 1986), ACM, New York, pp.59-64.

CANNON, J.J. (1969): Computers in group theory: A survey, *Commun. ACM,* **12,** pp.3-12.

CANNON, J.J. (1984): An introduction to the group theory language, Cayley, in Atkinson(1984), pp.145-183.

CANNON, J.J., and RICHARDSON, J.S. (1984/5): Cayley - Teaching group theory by computer, *SIGSAM Bull.,* **18/19,** pp.15-18.

EVEN, S., and GOLDREICH, O. (1981): The minimum-length generator sequence problem is NP-hard, *J. Algorithms,* **2,** pp.311-313.

FELSCH, V. (1978): A bibliography on the use of computers in group theory and related topics: algorithms, implementations, and applications, *SIGSAM Bull.,* **12,** pp.23-86.

FURST, M., HOPCROFT, J.E., and LUKS, E. (1980): Polynomial-time algorithms for permutation groups, *Proc. 21st IEEE Foundations of Computer Science,* pp.36-41.

GORENSTEIN, D. (1985): *The enormous theorem,* Scientific American **253**, 6, pp.92-103.

HALL, Jr, M. (1959): *The Theory of Groups,* Macmillan, New York.

HAVAS, G., and KOVACS, L.G. (1984): Distinguishing eleven crossing knots, in Atkinson(1984), pp.367-373.

HOFFMAN, C.M. (1982): *Group Theoretic Algorithms and Graph Isomorphism,* Lecture Notes in Computer Science, **136,** Springer-Verlag, Berlin.

KOCAY, W.L. (1984): Abstract data types and graph isomorphism, *J. Combinatorics, Information, and Systems Science,* **9,** pp.247-259.

LAUE, R., NEUBÜSER, J., and SCHOENWAELDER, U. (1984): Algorithms for finite soluble groups and the SOGOS system, in Atkinson(1984), pp.105-135.

LEON, J.S. (1979): An algorithm for computing the automorphism group of a Hadamard matrix, *J. Comb. Theory (A),* **27,** pp.289-306.

LEON, J.S. (1982): Computing automorphism groups of error correcting codes, *IEEE Trans. Infor. Theory,* **IT-28,** pp.496-511.

LEON, J.S. (1984a): Computing automorphism groups of combinatorial objects, in Atkinson(1984), pp.321-335.

LEON, J.S. (1984b): CAMAC2: A portable system for combinatorial and algebraic computation, *EUROSAM '84* (Proc. Internat. Symp. Symbolic and Algebraic Computation, Cambridge, July 9-11, 1984), J. Fitch (ed.), *Lecture Notes in Computer Science,* **174,** Springer-Verlag, Berlin, pp.213-224.

LEON, J.S., and SIMS, C.C. (1977): The existence and uniqueness of a simple group generated by (3,4)-transpositions, *Bull. Amer. Math. Soc.,* **83,** pp.1039-1040.

LUKS, E. (1980): Isomorphism of graphs of bounded valence can be tested in polynomial time, *Proc. 21st IEEE Foundations of Computer Science,* pp.42-49.

MAGLIVERAS, S.S., and LEAVITT, D.W. (1984): Simple 6-(33,8,36) designs from $P\Gamma L_2(32)$, in Atkinson(1984), pp.337-352.

MAGLIVERAS, S.S., OBERG, B.A., and SURKAN, A.J. (1987): A new random number generator from permutation groups, *Rend. Sem. Mat. Fis. Milano,* **54,** pp. 203-223.

MAGNUS, W., KARRASS, A., and SOLITAR, D. (1966): *Combinatorial Group Theory: Presentations of Groups in Terms of Generators and Relations,* Wiley Interscience, New York.

MCKAY, B.D. (1981): Practical graph isomorphism, (Proc. Tenth Manitoba Conf. on Numerical Math. and Computing, Winnipeg, vol. 1), *Congr. Numer.,* **30,** pp. 45-87.

NEUBÜSER, J. (1982): An elementary introduction to coset table methods in computational group theory, *Groups - St Andrews 1981* (Proc. internat. conf., St Andrews, July 25-August 8, 1981), London Mathematical Society Lecture Note Series, **71,** C.M. Campbell and E.F. Robertson (eds), Cambridge University Press, Cambridge, pp.1-45.

NEUBÜSER, J., PAHLINGS, H., and PLESKEN, W. (1984): CAS; Design and use of a system for the handling of characters of finite groups, in Atkinson(1984), pp.195-247.

NORTON, S.P. (1980): The construction of J_4, *Proc. Symp. Pure Math.,* **37,** pp.271-277.

WIELANDT, H. (1964): *Finite Permutation Groups,* Academic Press, New York.