Preface

This issue deals with the question:

> Given a set Γ of permutations of n letters, determine the structure of $\langle\Gamma\rangle$, the subgroup of the full permutation group S_n that is generated by Γ.

Here, the word "determine" should be interpreted algorithmically. As is more often the case in the study of algorithms, a more precise formulation of the question is obtained by interchanging the quantifiers implicit in the problem statement. For example, rather than asking for all sets Γ whether there is a way to test if the group $\langle\Gamma\rangle$ is commutative, we mean:

> Is there an efficient algorithm that, given a finite set Γ of elements of S_n, decides whether the subgroup $\langle\Gamma\rangle$ of S_n is commutative?

The emphasis is on efficient algorithms. If the word "efficient" is removed here, an uninteresting problem remains: since all sets under consideration are finite, one could just list all elements of the group generated by Γ and check whether or not each pair of elements from the list commutes. This is an algorithm for testing commutativity of a group which is, in general, highly inefficient.

Roughly speaking there are two ways of dealing with the efficiency question. There is the practical method: many permutation group algorithms have been implemented over the last 25 years, and all sorts of statistics have been collected on how far one can go with a specific algorithm in a given amount of time and with given space, and so on. The other — more theoretical — way of expressing the efficiency of an algorithm is by providing an asymptotic upper bound for the number of operations (and the storage size) involved in a worst case computation using the algorithm. For permutation groups, the basic operation is to find the image of an element from $\{1,\ldots,n\}$ under a permutation. Multiplying two permutations takes $2n$ such basic operations. Checking whether $\langle\Gamma\rangle$ is commutative by means of the definition comes down to checking $xy = yx$ for all unordered pairs x, y from $\langle\Gamma\rangle$, which would require $|\langle\Gamma\rangle|^2$ multiplications. Since $\langle\Gamma\rangle = S_n$ may occur for small Γ (even of size 2), this algorithm is outrageously inefficient (the number of multiplications needed is exponential in n). However, $\langle\Gamma\rangle$ is commutative iff $xy = yx$ for all unordered pairs x, y from Γ, so, by restricting the commuting pairs test to elements from Γ, we obtain an algorithm needing only $|\Gamma|^2$ multiplications. This number is polynomial in the input size

which is expressed as a function of n and $|\Gamma|$. It is one of the standard measures of efficiency within computer science. Of course, similar remarks apply for space efficiency.

The polynomiality criterion for efficiency is still rather coarse. For a polynomial upper bound the first refinement is the degree of the polynomial in n. One might argue (and some have done so) that degree 3 is a realistic upper bound for feasibility beyond 'the first few cases'.

The commutativity example shows that, apart from the size n of the underlying set, the cardinality $|\Gamma|$ is a relevant number. Since Γ is only used to record the permutation group, one might ask whether, for each permutation group, a relative small set of generators can be found, for instance the "realistic bound" as a function of n. In fact one can do a lot better, see §2 of the paper by Bosma and Cannon in this issue.

Kantor's paper in this issue indicates several structural properties of the group $\langle\Gamma\rangle$ that can be computed in polynomial time and also some that very likely cannot. The paper by Bosma and Cannon provides a general introduction spiced with fundamental algorithms, more sophisticated versions of which are often implemented in the computer package CAYLEY — the standard package for computer group theory over the last few years. (Only recently, the package GAP has become a serious competitor.) CAYLEY is Cannon's brain child; Bosma has become actively involved in the building of a new version.

As noted above, not all algorithms can be made satisfactorily efficient. An old and practical way around this difficulty has been to resort to probabilistic methods. For instance, finding a Sylow p-subgroup can be done by iteratively finding a subgroup of order a power of p of which a given (smaller) subgroup is a normal subgroup. To start the induction, one has to find an element of order p. It is often practical to pick an element at random and check its order; if its order is a multiple of p, take a suitable power (and, if not, repeat; see §6 of the paper by Bosma and Cannon). By the way, finding Sylow p-subgroups is polynomial in n by deterministic methods (see Theorem 4.4 of Kantor's paper)!

Recently, the probabilistic methods have been given a theoretical foundation, and have been considerably improved. Cooperman and Finkelstein, who author the third paper in this issue, are two of the main contributors to these developments. They have taken up the challenge of implementing their algorithms and finding out whether or not the improvements are purely theoretical. Although the outcome of this enterprise is not yet clear, it relates to a new trend in which theory and practice are coming closer than before.

Altogether, the papers of this issue present an introduction to the state of the art in computational permutation group theory. During the last few years, the research in this field has been very productive, so much so that there are too many interesting results for them to be fully described in these limited number of pages. Valuable references for further reading are the special volume 12 of the J. Symbolic Computation devoted to computational group theory and the forthcoming proceedings of the 1992 DIMACS workshop on Groups and Computation.

The surge of computational group theory has been one reason for dedicating this issue to that topic. There is also a personal one. Being a user of several of the implemented algorithms, I found it exciting to learn about the ideas underlying my computations of Sylow subgroups, centralizers and the like. Since permutation groups play a role in quite a few researchers' lives, I expect the same will hold for other people. But even for those who are not actively involved in group theory, I am confident that the greater part of this CWI Quarterly issue will make enjoyable and beneficial reading.

Arjeh M. Cohen