Proof Planning some Permutation Group Problems — Abstract —

Arjeh Cohen[1], Scott H. Murray[1], Martin Pollet[2], Volker Sorge[3]

[1] RIACA, Technische Universiteit Eindhoven,
The Netherlands, {amc$|$smurray}@win.tue.nl
[2] Fachbereich Informatik, Universität des Saarlandes,
Germany, pollet@ags.uni-sb.de
[3] School of Computer Science, University of Birmingham, UK,
V.Sorge@cs.bham.ac.uk

We describe the integration of permutation group algorithms from computer algebra with proof planning. We consider eight basic questions arising in computational permutation group theory, for which a computer algebra system provides solutions together with sets of certificates. The certificates are used to guarantee correctness by constructing a formal proof employing proof planning techniques.

The experiments were carried out by combining the computer algebra system GAP [2] and the proof planner of the Omega system [3]. We chose GAP, since it is particularly good in group theory and has a rich collection of permutation group algorithms. The choice of Omega was motivated by the fact that it enables the construction of proofs in a human-oriented reasoning style.

The results of our experiments can be summarised as follows: (i) We provide GAP functions which handle eight basic queries, ranging from "Is this permutation in that permutation group?" to "What is the order of this permutation group?" The functions provide certificates as well as solutions, enabling a user or an intelligent software system to provide a full proof of correctness of the solution. (ii) We provide proof planning constructs in Omega to prove that the answers given by GAP to these eight queries are correct; each query can essentially be modelled as a theorem and the proof planner can verify GAP's answers using the additional certificates to guide the planning process. Omega treats single queries independently from GAP, in a modular and easily extensible approach. As far as possible, we model a human-oriented reasoning style to give a human mathematician the necessary insights into the computed solutions. (iii) To this end we also implemented a collection of functions in GAP for turning answers and cerificates into natural language proofs using simple templates. Similarly, the proofs produced by Omega can be turned into natural language by means of the connected P.Rex system [1], which employs elaborate linguistic techniques. Notice that the natural language proof from GAP does not satisfy a correctness criterion, as it is implemented by humans and so a mistake in the proof could remain unnoticed.

The need to integrate automated proof assistants with computer algebra systems comes from two sources. On the computer algebra side, the production of certificates becomes necessary when systems are used via the internet as 'oracles.' In this setting, it is likely that users will not know where the answer is coming from and so will need to be convinced of its correctness. With a certificate, it is a relatively easy task to perform a verification. On the proof planning side, we wish to demonstrate that we meet the challenge of reconstructing a proof from exactly

the same mathematical data that a human would require. It would be unreasonable to expect more than that from a proof developing system. The fact that Omega can produce a proof from a certificate also ensures that the certificates have indeed supplied sufficient information.

In detail we concentrate on the following eight problems: Let G be a group generated by a set A of permutations of the points $\Omega := \{1, 2, \ldots, n\}$

1. **Membership:** show that a permutation g is an element of G
2. **Subgroup:** show that a group H is a subgroup of G
3. **Orbit:** determine an orbit containing a given point $x \in \Omega$, $xG = \{xg : g \in G\}$
4. **Schreier tree:** compute the Schreier tree rooted at x to determine a set of coset representatives for the orbit xG
5. **Stabiliser:** compute the stabilisers $G_x = \{g \in G : xg = x\}$.
6. **Base:** compute the base for G, i.e. a finite sequence $B = [x_1, \ldots, x_k]$ of distinct points in Ω such that $G_{x_1,x_2,\ldots,x_k} = 1$.
7. **Non-membership:** show that a permutation g is <u>not</u> an element of G
8. **Order:** compute the order of G given a base $B = [x_1, \ldots, x_k]$ as $|G| = \prod_{i=1}^{k} |x_i G^{(i-1)}|$.

We emphasise that this work extends the boundaries of what is feasible. Clearly, the eight permutation group queries we are dealing with can all be handled by simple enumeration. For instance, in order to decide if a given permutation g belongs to a permutation group G, you could just enumerate all elements of G and check whether g is one of these. However, G can be exponentially large as a function of n, the number of letters permuted by G, and so this soon becomes impractical. We use GAP's sophisticated group-theoretic algorithms for finding proofs in cases far beyond reach of such enumeration.

We use the idea of providing verified certificates for computation subproblems, which gives more mathematical insight into the solutions, i.e. we model a human reasoning approach rather than a machine-oriented one. Indeed the constructed proof plans follow the proof idea given by the GAP certificates, which are sufficient for a working mathematician to verify correctness. The results also show that our approach can be successfully applied to problems involving large and complex structures, as we can check mathematical data of a magnitude that is likely beyond the range of traditional theorem proving systems. The work can now serve as a basis to approach computation problems in graph theory, like showing two graphs are not isomorphic, in a similar fashion.

References

[1] Armin Fiedler. *P.rex*: An interactive proof explainer. In *Proceedings of IJCAR 2001*, *LNAI* 2083, pages 416–420. Springer Verlag, 2002.

[2] The GAP Group, Aachen, St Andrews. *GAP – Groups, Algorithms, and Programming, Version 4*, 1998. `http://www-gap.dcs.st-and.ac.uk/~gap`.

[3] J. Siekmann, C. Benzmüller, V. Brezhnev, L. Cheikhrouhou, A. Fiedler, A. Franke, H. Horacek, M. Kohlhase, A. Meier, E. Melis, M. Moschner, I. Normann, M. Pollet, V. Sorge, C. Ullrich, C.-P. Wirth, and J. Zimmer. Proof development with omega. In *Proceedings of CADE–18*, *LNAI* 2392, pages 143–148. Springer Verlag, 2002.