Journal of Symbolic Computation ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Symbolic Computation

www.elsevier.com/locate/jsc

Efficient characteristic refinements for finite groups

Joshua Maglione

Department of Mathematics, Colorado State University, Fort Collins, CO 80523, USA

ARTICLE INFO

Article history:
Received 18 February 2016
Accepted 4 July 2016
Available online xxxx

Keywords:
p-Groups
Isomorphism
Lie algebras
Filters

ABSTRACT

Filters were introduced by J.B. Wilson in 2013 to generalize work of Lazard with associated graded Lie rings. It holds promise in improving isomorphism tests, but the formulas introduced then were impractical for computation. Here, we provide an efficient algorithm for these formulas, and we demonstrate their usefulness on several examples of *p*-groups.

1. Introduction

Isomorphism between two finite groups becomes easier when we use isomorphism invariant subgroups (i.e. characteristic subgroups) to constrain the number of possibilities. With this in mind, Fitting uncovered several characteristic subgroups to later be used to determine isomorphism between groups (Fitting, 1938), see the accompanying bibliography in Cannon and Holt (2003). However, in the case of *p*-groups, these characteristic subgroups are usually the whole group or the trivial group. As seen in Eick et al. (2002), the inclusion of just one new characteristic subgroup can greatly improve performance.

New sources for computable characteristic subgroups of *p*-groups were uncovered in Wilson (2013, 2015). In addition, it was shown that the inclusion of new characteristic subgroups induced more subgroups and gave formulas to automate this process of refining. However, the formulas required an exponential amount of computation. In this paper, we prove that we can do this in polynomial time and we provide an implementation for Magma. Indeed, even for groups of order 3^{100}, we are able to refine a typical characteristic series by about ten-fold in just a few minutes; see Fig. 1 on page 9.

E-mail address: maglione@math.colostate.edu.

http://dx.doi.org/10.1016/j.jsc.2016.07.007
0747-7171/

A *filter* for a group G is a function $\phi : M \to 2^G$ from a commutative monoid $M = \langle M, +, 0, \preceq \rangle$ into the normal subgroups of G satisfying the following: for all $s, t \in M$

$$[\phi_s, \phi_t] \leq \phi_{s+t} \qquad \& \qquad s \preceq t \Longrightarrow \phi_t \leq \phi_s.$$

Wilson proves (Wilson, 2013, Theorem 3.1) that each filter has an associated Lie ring:

$$L(\phi) = \bigoplus_{s \in M - \{0\}} \phi_s / \langle \phi_{s+t} \mid t \in M - \{0\} \rangle . \tag{1}$$

The use of monoids M is essential as it allows for somewhat arbitrary refinements some of which are discussed in Section 4. We prove the following theorem.

Theorem 1. *Suppose $\phi : \mathbb{N}^d \to 2^G$ is a filter where $\preceq$ is the lexicographical order. If $H \triangleleft G$ and there exists $s \in \mathbb{N}^d$ such that*

$$\left\langle \phi_{s+t} \mid t \in \mathbb{N}^d - \{0\} \right\rangle < H < \phi_s,$$

then there exists a polynomial-time algorithm that refines ϕ to contain H in its image.

The result is smaller homogeneous components, faster automorphism computations, and an easier explanation of structure. Indeed, in Maglione (2015a), it was shown that even well-studied unipotent classical groups admit surprises such as characteristic filters whose factors are at most of order p^2. Together with Eick et al. (2002), this then reduces automorphism questions to $\mathrm{GL}(2, p)$ instead of $\mathrm{GL}(d, p)$. This and further uses in Maglione (2015b), Wilson (2015) make it desirable to compute with filters efficiently.

In addition to providing a computational framework for filters in Section 3, we refine several filters for common examples of p-groups in Section 5. We look to large examples in the literature and we also consider a sample of 2,000 sections (i.e. quotients of subgroups) of the Sylow 3-subgroups of classical groups of Lie type. We find that the larger the group, the more new structure we find, and because of the repetitive nature, often one discovery leads to more discoveries. All of our computations were run in Magma V.21-5 (Bosma et al., 1997) on a computer with Intel Xeon W3565 microprocessors at 3.20 GHz.

2. Preliminaries

We denote the set of nonnegative integers by $\mathbb{N}$, and the set of all subsets of a set G by 2^G. For groups and rings, we follow notation found in Gorenstein (1980). For $g, h \in G$, we set

$$[g, h] = g^{-1} g^h = g^{-1} h^{-1} g h;$$

for $X, Y \subseteq G$, we set

$$[X, Y] = \langle [x, y] : x \in X, y \in Y \rangle.$$

We let $\mathbb{Z}_p$ denote the group $\mathbb{Z}/p\mathbb{Z}$.

For a p-group G, we consider two recursively defined series: the lower central series and the exponent-p central series. The lower central series starts with $\gamma_1(G) = G$ and $\gamma_{i+1}(G) = [\gamma_i(G), G]$, and the exponent-$p$ central series begins with $\eta_1(G) = G$ and $\eta_{i+1}(G) = [\eta_i(G), G]\eta_i(G)^p$. The class ($p$-class) of G is the number of nontrivial terms in the lower central series (exponent-p central series).

2.1. Complexity

An algorithm runs in *polynomial time* if the number of operations it uses is bounded by a polynomial of the input length. At least one mark of efficiency is polynomial time, but we include run times from experiments as well.

Please cite this article in press as: Maglione, J. Efficient characteristic refinements for finite groups. J. Symb. Comput. (2016), http://dx.doi.org/10.1016/j.jsc.2016.07.007