Computing in Solvable Matrix Groups

Eugene M. Luks

CIS-TR-92-17
August 1992

Abstract

We announce methods for efficient management of solvable matrix groups over finite fields. We show that solvability and nilpotence can be tested in polynomial-time. Such efficiency seems unlikely for membership-testing, which subsumes the discrete-log problem. However, assuming that the primes in $|G|$ (other than the field characteristic) are polynomially-bounded, membership-testing and many other computational problems are in polynomial time. These problems include finding stabilizers of vectors and of subspaces and finding centralizers and intersections of subgroups. An application to solvable permutation groups puts the problem of finding normalizers of subgroups into polynomial time. Some of the results carry over directly to finite matrix groups over algebraic number fields; thus, testing solvability is in polynomial time, as is testing membership and finding Sylow subgroups.

Department of Computer and Information Science
University of Oregon
Eugene, OR 97403

to appear in
Proceedings 33rd IEEE Symposium on
Foundations of Computer Science, 1992

Computing in Solvable Matrix Groups

Eugene M. Luks*
Computer and Information Science
University of Oregon
Eugene, OR 97403

Abstract

We announce methods for efficient management of solvable matrix groups over finite fields. We show that solvability and nilpotence can be tested in polynomial-time. Such efficiency seems unlikely for membership-testing, which subsumes the discrete-log problem. However, assuming that the primes in $|G|$ (other than the field characteristic) are polynomially-bounded, membership-testing and many other computational problems are in polynomial time. These problems include finding stabilizers of vectors and of subspaces and finding centralizers and intersections of subgroups. An application to solvable permutation groups puts the problem of finding normalizers of subgroups into polynomial time. Some of the results carry over directly to finite matrix groups over algebraic number fields; thus, testing solvability is in polynomial time, as is testing membership and finding Sylow subgroups.

1 Introduction

Over the past 30 years, Computational Group Theory has matured into a fertile and valuable discipline (see [Le], [Ca], [At], [Sc]). Within it, the most developed subdomain has been that of permutation groups, the apparent reason being that a small set of generating permutations can designate a very large group. However, that observation would be useless without methods for dealing effectively with the groups that one can specify.

Such methods exist. Fundamental is a technique first proposed by Sims in the 1960's (see, e.g., [Si]). Suppose $G \le \mathrm{Sym}(\Omega)$, the group of all permutations of $\Omega = \{\omega_1, \dots, \omega_n\}$. For $1 \le i \le n$, let $G^{(i)}$ be the subgroup of G that fixes each of $\omega_1, \dots, \omega_{i-1}$ ($G^{(1)} = G$). Then $\{\omega_1, \dots, \omega_m\}$ is called a *base* for G if $G^{(m+1)} = 1$. The tower of subgroups

$$G = G^{(1)} \ge G^{(2)} \ge \cdots \ge G^{(m+1)} = 1$$

underlies almost all practical algorithms. However, to make use of it, one needs to obtain generators for every $G^{(i)}$. Fortunately, there is a classical method of Schreier (see [Ha]) that constructs generators for the subgroup $G^{(i+1)} \le G^{(i)}$ given generators for $G^{(i)}$ together with a complete set, T_i, of right coset representatives for $G^{(i+1)}$ in $G^{(i)}$.

To appreciate the effectiveness of the tower, note, for example, that testing membership in $G = G^{(1)}$ of a permutation s reduces to testing membership in $G^{(2)}$ of st^{-1}, where $t \in T_1$ is uniquely determined by $\omega_1^t = \omega_1^s$.

Critical to the implementations of this, and the myriad practical procedures that utilize the tower, is that the indices $|G^{(i)} : G^{(i+1)}|$ are "small" ($|G^{(i)} : G^{(i+1)}| \le n$); for example, Schreier generation demands consideration of the full T_i. This is also essential for *polynomial-time* computation in permutation groups. In fact, in 1980, Furst, Hopcroft and Luks [FHL] showed that a variant of Sims's method runs in polynomial time. That result inaugurated the now-substantial polynomial-time library for permutation groups (see [KL] for a summary).

In stark contrast to the permutation-group progress is the underdeveloped status of computation for matrix groups over finite fields. The matrix setting could be more desirable and more natural for representing and applying finite groups. Unfortunately, there have been almost no provably-efficient methods for managing matrix groups. Indeed, there are signs that even elementary problems may not have uniformly efficient procedures in this domain. Consider the basic problem of testing membership. Even for 1×1 matrices, this involves the discrete-log problem in finite fields, which seems unlikely to have a polynomial-time solution (see, e.g., [Bac]).

But, as we shall make clear, this discouraging observation does not close the door on all important instances of membership-testing. In any case, it does not fully account for the inefficiency of commonly implemented algorithms. A major bottleneck in the "standard" procedure for matrix-group membership is its direct utilization of Sims's algorithm ([Bu], [Ca]), in effect dealing with the full matrix group $GL(n, F)$ as a subgroup of the permutation group $\mathrm{Sym}(F^n)$. To be

*Research supported by NSF grant CCR-9013410

sure, one does not need to *list* this exponential-sized permutation domain, and there is still a base of reasonable size (a vector-space basis of F^n is a base for any subgroup of $GL(n, F)$). Nevertheless, there is an intrinsic blowup in the application of Sims's methods, for the indices $|G^{(i)} : G^{(i+1)}|$ can be exponential. This can happen even for elementary-abelian 2-groups represented over the 2-element field $GF(2)$. E.g., consider the subgroup $G < GL(n, 2)$ that fixes every vector in a subspace of dimension $n-1$; assuming ω_1 is any vector that is moved by G, we see that $|G^{(1)} : G^{(2)}| = 2^{n-1}$. This bottleneck is clearly unrelated to the discrete log obstruction, for the only prime in $|G|$ is 2.

We also demonstrate that not every important problem demands membership-testing. To be sure, membership-testing is so efficient in the permutation-group setting that one does not hesitate to call it. A case in point: polynomial-time testing of permutation-group solvability was touted in [FHL] as an immediate consequence of Sims's methods (membership-testing enables normal closures, therefore derived series). Nevertheless, we now show that solvability and nilpotence are polynomial-time testable for finite matrix groups (by which we mean something more than matrix groups over finite fields - see below). By comparison, previous methods for finite matrix groups put testing solvability and nilpotence in NP$\cap$co-NP ([BS]) or in random (Monte-Carlo) polynomial-time ([BCFLS]).

For issues that must involve membership-testing, one can strive for a reduction to discrete logs, and we could state some of our main results in that fashion. However, for the sake of ready exposition of the new techniques, it is more convenient to introduce an additional parameter into the timing. To wit, we develop a large library of solvable-matrix-group problems which can be solved in time that is polynomial in $(n+\log|F|+\mu(G))$, where $\mu(G)$ is the largest prime in $|G|$ other than the characteristic of F. This includes: membership-testing; finding $|G|$; finding the subgroup that stabilizes a vector or a subspace; finding subgroup intersections; finding centralizers; in fact, finding the structural "building blocks" of $|G|$.

Lest the reader be concerned about the non-input parameter $\mu(G)$ in the timing, we emphasize that these results are new even for groups in which $\mu(G)$ is absolutely bounded, in fact, even for p-groups where p is fixed. Furthermore, we indicate some natural and important applications in which $\mu(G)$ is guaranteed to be small.

For example, $\mu(G)$ is small in various applications to permutation group problems. Most notably, we now find normalizers of subgroups of solvable permutation groups in polynomial time. A previous polynomial-time method extended only to nilpotent permutation groups [KL]. Implemented procedures for permutation groups (e.g., in the group-theory systems Cayley [Ca] and GAP [Sc]) use backtracking methods that require exponential time in the worst case.

There are also applications to permutation groups that give new perspectives on problems that were previously known to be in polynomial time, but required much deeper methods. An example is finding the intersection of subgroups of a solvable quotient G/K of permutation groups (note, only G/K is hypothesized to be solvable, G need not be). It was announced in [KL] that this can be done in polynomial time. However, the method required Kantor's algorithms for Sylow subgroups ([Ka1], [Ka2]) which make essential use of consequences of the classification of finite simple groups (so they have a 15000 page proof). We can now offer a self-contained, elementary approach to this and other previously classification-dependent problems.

Another application is to finite groups $G < GL(n, F)$, where F is an algebraic number field. Babai, Beals and Rockmore [BBR] have shown how to embed such G in $GL(n, \mathbb{Z})$. But then the natural homomorphism $\mathbb{Z} \to \mathbb{Z}_p$, for any $p > 2$, faithfully embeds G in $GL(n, p)$ and $\mu(G) \leq n + 1$. Consequently, membership-testing is in polynomial time for finite solvable matrix groups over algebraic number fields. By earlier remarks, testing whether a finite matrix group is solvable or nilpotent is also shown to be in polynomial time; indeed, much deeper structural information is available (including composition series and Sylow subgroups).

Our approach to membership-testing involves a top-down decomposition of the group (quite unlike Sims's method, wherein almost no deep knowledge of the group, other than its order, is immediately needed or revealed). This is reminiscent of methods used for NC computation in permutation groups ([LM], [Lu2], [BLS]). The common approach assumes we already have a presentation $\langle X|\mathcal{R}\rangle$ of some quotient G/H of G, and the immediate goal is to get enough information about H to produce some non-trivial presentation of a quotient H/K, which, by standard means, then yields a presentation of G/K, etc. Using the Schreier method to get generators of H is out of the question for $|G:H|$ can be exponential. However, "sifting" the relations $\mathcal{R}$ produces "normal generators" of H, i.e., generators of a subgroup whose normal closure in G is H. One then needs to construct a manageable representation domain for some H/K, an operation that may have to bootstrap from the group structure itself.

In order to maintain determinism, we do not pass to irreducible actions, as finding these presently requires Las Vegas methods when F is large [Ró]. On the other hand, our procedures do exploit invariant subspaces when these are available. In fact, we offer a divide-and-conquer paradigm for finding important subgroups, such as subspace-stabilizers, which relies on both invariant subspaces and imprimitivity systems. However, unlike their analogues in permutation-group computation (orbits and imprimitivity blocks [Lu1]), these are not easy to locate.

We summarize some of the main results in §3, with samples of the methods in §§4-6. We propose issues for further research in §7.

Finally, we emphasize that our goal is polynomial-time computation. We leave for later investigation low-level complexity issues and practical implementation, worthy projects in which we invite participation.

2 Notation and Preliminaries

We refer to standard references (e.g., [Ha]) for terminology not recalled here. G always denotes a *finite* group.

We write $H \leq G$ if H is a subgroup of G and $H \trianglelefteq G$ if H is a normal subgroup, $H < G$ and $H \triangleleft G$ indicating strict inclusion. The *index* of H in G is $|G|/|H|$ and is denoted $|G:H|$. For $A \subseteq G$, $\langle A \rangle$ is the subgroup *generated by* A, i.e., the smallest subgroup containing A. The *order of* $g \in G$ is $|\langle g \rangle|$ and denoted $o(g)$.

For $g, h \in G$, $g^{-1}hg$ is the *conjugate of* h *by* g and is denoted h^g. If $A \subset G$, the *centralizer of* A *in* G is $C_G(A) = \{g \in G \mid \forall a \in A, a^g = a\}$; $C_G(G)$ is also called the *center* of G and is denoted $Z(G)$. The *normal closure* of H in G is the smallest normal subgroup of G that contains H and is denoted H^G. The *normalizer* of H in G is the largest subgroup of G in which H is normal, i.e. $\{g \in G \mid g^{-1}Hg = G\}$.

If $g, h \in G$, $[g,h] = g^{-1}h^{-1}gh$ (the *commutator* of g and h). If $S, T \subseteq G$, $[S,T]$ denotes the subgroup $\langle \{[s,t] \mid s \in S, t \in T\} \rangle$. The *derived* subgroup of G is $G' = [G,G]$ and the *derived series* is $G \geq G' \geq (G')' \geq \cdots$; G is said to be *solvable* if the derived series terminates with 1. The *lower central series* of G is defined recursively by: $L_1(G) = G$ and $L_{i+1}(G) = [G, L_i(G)]$; G is said to be *nilpotent* if the lower central series terminates with 1; it is *nilpotent of class* i if $L_i(G) > L_{i+1}(G) = 1$. Abelian groups are nilpotent and nilpotent groups are solvable; neither converse holds.

For a prime p, G is a *p-group* if $|G|$ is a power of p and G is a *p'-group* if $\gcd(|G|, p) = 1$. A group is nilpotent iff it is a direct product of p-groups. A *Sylow p-subgroup* of G is a p-subgroup of maximal order; a group is nilpotent iff it has a unique Sylow p-subgroup for each p.

A group is called *simple* if it has no proper normal subgroups. A composition series of G is a tower $G = H_1 \triangleright H_1 \triangleright \cdots \triangleright H_m = 1$ such that each quotient H_i/H_{i+1} is simple. Then G is solvable if the simple groups in some (hence all) composition series are cyclic.

The group of all permutations of the set Ω is denoted by $\mathrm{Sym}(\Omega)$. One says that G *acts on* Ω if there is a homomorphism $G \rightarrow \mathrm{Sym}(\Omega)$. If G acts on Ω, one writes ω^g, respectively, Δ^g, for the image of $\omega \in \Omega$, respectively, the image of $\Delta \subseteq \Omega$, under $g \in G$; then G is said to be *transitive* if $\Omega = \{\omega^g \mid g \in G\}$ for some (therefore, all) $\omega \in \Omega$. A subset $\Delta \subseteq \Omega$ is said to be *G-invariant* and G is said to *stabilize* Δ if $\Delta^g = \Delta$ for all $g \in G$. If G stabilizes Δ, we denote by G^Δ the image of G in $\mathrm{Sym}(\Delta)$. For any $\Delta \subset \Omega$, the *pointwise stabilizer of* Δ *in* G is $C_G(\Delta) = \{g \in G \mid \delta^g = \delta, \forall \delta \in \Delta\}$; note that this is consistent with the centralizer notation $C_G(H)$, with G acting via conjugacy. For $G < \mathrm{Sym}(\Omega)$, the set of *fixed points of* G is $\mathrm{Fix}(G) = \{\omega \in \Omega \mid \omega^g = \omega, \forall g \in G\}$.

The standard vector space of n-tuples over a field F is denoted F^n. In any vector space, $\mathrm{Span}(A)$ is the linear span of the set A. For vector spaces V, W over F, $\mathrm{Hom}(V,W)$ is the space of linear transformations from V to W. The group of nonsingular linear transformations of V is denoted $GL(V)$. Note that $GL(V) < \mathrm{Sym}(V)$ and so the notation of the preceding paragraph applies to subgroups of $GL(V)$. A group $G < GL(V)$ is said to be *reducible* if there exists a proper G-invariant subspace of V. An *imprimitivity system* for $G < GL(V)$ is a collection, $\{W_1, \ldots, W_m\}$, of subspaces of V such that $V = W_1 \oplus \cdots \oplus W_m$ and G permutes the summands, acting transitively on that m-element collection (thus. $\dim(W_i) = \dim(V)/m$). The group of $n \times n$ matrices over the q-element field $GF(q)$ is denoted $GL(n,q)$. It is often convenient to indicate that $G < GL(V) = GL(n,q)$, where V is an n-dimensional vector space, so that we have all parameters available for the discussion.

A matrix is called *unipotent* if its only eigenvalue is 1; a matrix group is called *unipotent* if it consists of unipotent matrices. A matrix group is unipotent iff it is simultaneously triangulable with all main diagonals comprised of 1's. If the characteristic of F is p then a subgroup of $GL(n,F)$ is unipotent iff it is a p-group.

We call an abelian matrix group $A < GL(V)$ *uniform* if, for every integer m, $\mathrm{Fix}(A^m) = \{0\}$ or V, where $A^m = \{a^m \mid a \in A\}$.

Implicit to timing analyses is the fact that strictly decreasing sequences of subspaces (of F^n) or subgroups (of $GL(n,q)$) have polynomial length (bounded by n or $\log_2 |GL(n,q)| < n^2 \log_2 q$, respectively).

3 Main Results

Subgroups $G \leq GL(n,F)$ are input and output via sets of generators. Furthermore, we make the overall assumption that the input-generating-sets are not unreasonably large (e.g, for finite F, it is warranted to suppose that these are no larger than $\log|GL(n,F)| = O(n^2 \log|F|)$) so that we need not bother to incorporate their size in the asymptotic timings. We also assume reasonable encodings of finite F so that "polynomial-time," by itself, means polynomial in $(n + \log|F|)$.

Theorem 3.1 *Given $G \leq GL(n,F)$, where F is a finite field, one can test in polynomial time whether G is solvable and, if so, whether G is nilpotent.*

If G is solvable one can also find, for each prime $p \leq n^{\text{constant}}$, the p-part of $|G|$. If G is nilpotent, one can find its (unique!) Sylow p-subgroup for each such prime.

Theorem 3.2 *Given a solvable $G \leq GL(n,F)$, where F is a finite field, let μ denote the largest prime dividing $|G|$ other than the characteristic of F. The following problems can be solved in time that is polynomial in $(n + \log|F| + \mu)$.*

(1) *Find $|G|$.*

(2) *Given $x \in GL(n,F)$, test whether $x \in G$.*

(3) *Find a generator-relator presentation for G.*

(4) *Find a composition series in G.*

(5) *Given any vector $v \in F^n$, find the subgroup of G that fixes v.*

(6) *More generally, given any set A of vectors, find the subgroup of G that stabilizes the set A (assuming $|A|$ is not large, otherwise we can express the timing as a polynomial in $n + \log|F| + \mu + |A|$).*

(7) *Given any subspace $W \leq F^n$ (via a spanning set), find the subgroup of G that stabilizes W.*

(8) *Given any $x \in GL(n,F)$, find $C_G(x)$, the centralizer of x in G. Thus, given any (not necessarily solvable) $H \leq GL(n,F)$, find $C_G(H)$.*

(9) *Given $H, K \leq GL(n,F)$, find $H \cap K$.*

(10) *For any prime p, find a Sylow p-subgroup of G.*

An application of (7) leads to

Corollary 3.3 *Given permutation groups H, G, with $H \leq G$ and G solvable, one can find the normalizer of H in G in polynomial time.*

More generally, the normalizer of H in G can be found if $H/K \leq G/K$, where $K \trianglelefteq G$, assuming only that G/K is solvable.

Other applications to quotients of permutation groups include finding $C_A(B)$ and $A \cap B$ when $A, B < G/K$, with G/K solvable. These two are not new results for polynomial-time but previous methods [KL] (which are still more generally applicable) used consequences of the classification of finite simple groups.

Corollary 3.4 *Given $G < GL(n,F)$, where F is an algebraic number field, one can test whether G is a finite solvable or nilpotent group in polynomial time and, if so, test membership in G.*

One can also then find $|G|$, normalizers and centralizers of subgroups, intersections of subgroups, composition series and Sylow subgroups.

4 Membership-Testing and Basics

Throughout this section, we assume that we are dealing with solvable groups $G = \langle S \rangle < GL(n,q) = GL(V)$ for which $\mu(G)$ is bounded by a fixed polynomial in the size of the input. Alternately, the results can be rephrased, replacing "polynomial-time" by "polynomial in $|S| + n + \log q + \mu(G)$."

Assuming some reasonable encoding of the finite field F (e.g., as $GF(q)[x]/(f(x))$ for some irreducible f), it is elementary to convert our setting to matrix groups over the prime field $GF(q)$ (blowing up matrix sizes by a factor of degree(f)). Thus, we assume that $G \leq GL(n,q)$, where q is prime.

4.1 Overall Procedure

We construct, in a top-down fashion, a tower of normal (in G) subgroups

$$G = H_1 \triangleright H_2 \triangleright \cdots \triangleright H_m = 1$$

together with manageable representations $\phi_i : H_i \to M_i$ with $H_{i+1} = \text{kernel}(\phi_i)$.

By "manageable" we mean essentially that M_i is a type of group for which suitable machinery (testing membership, finding presentations) has already been developed. For example, if $M_i = \text{Sym}(\Omega)$ then we can appeal to standard permutation group machinery. The idea is further exploited by resolving all critical issues first with abelian groups (§4.4), after which M_i may be an abelian matrix group. With such representations, membership-testing, for example, becomes a *sifting* process (cf. [FHL]) through the tower: map a candidate x for membership in H_i by π_i to M_i (membership-failure could be reported in the call to the function π_i which may not be defined on x); find an $h \in H_i$ such that $\pi_i(h) = \pi_i(x)$; test membership of xh^{-1} in H_{i+1}.

"Top-down" in the tower construction has an algebraic as well as an algorithmic connotation. That is, the group that we shall have constructed at any stage is G/H_i. We shall not actually have generators for any H_i until the entire procedure has run. Furthermore, though we have indicated that we construct domains in which to study the quotients H_i/H_{i+1}, we do not necessarily have faithful representations (e.g., as permutation groups or matrix groups) of the full quotients G/H_i. However, there is a concrete way in which we do have G/H_i in hand, namely via a generator-relator presentation $\langle X_i|\mathcal{R}_i\rangle$ (the computational aspect of this is formalized in §4.2).

One particular advantage of the representation of quotient groups via presentations is that the relations provide, in a natural manner, precisely the needed elements of H_i with which to carry on. Corresponding to the relations R_i is a set $T_i \subset H_i$ such that $H_i = \langle T_i\rangle^G$ and the constructive nature of the presentation (§4.2) gives access to T_i.

The core of the method then is, using only T_i, which still may not fully generate H_i, to construct the map $\pi_i : H_i \to M_i$. More precisely, π_i is a function that would be defined on any elements of H_i as they are later encountered. The group M_i will also be acted on by G and the map π_i will be a G-map, that is $\pi_i(h^g) = (\pi_i(h))^g$ for all $h \in H_i$. Thus, in M_i, full generators for, and a presentation of $\langle \pi_i(T_i)\rangle^G$ can be constructed. All of this can be pulled back to $H_i/\text{kernel}(\pi_i)$.

Since π_i is a G-map, $\text{kernel}(\pi_i) \trianglelefteq G$. We let $H_{i+1} = \text{kernel}(\pi_i)$ and use the presentations of G/H_i and H_i/H_{i+1} to form a presentation of G/H_{i+1}, thus stepping down the tower.

When the tower is exhausted, we have in hand a data structure for membership-testing. But the procedure has other important corollaries. An easy example is that of finding $|G|$. Assuming that, for the "manageable" M_i, finding orders of subgroups is in polynomial-time (this will be the case) then $|G| = \prod_i |\pi(H_i)|$. Having access to effective presentations in a class of groups has other applications, including finding the center of a group (§4.6). In fact, we need centers in p-groups within the procedure for solvable groups. Thus, we first deal with nilpotent groups in §4.5.

4.2 Presentations

Our algorithmic formulation of presentations $\langle X|\mathcal{R}\rangle$ for quotients G/H must allow the passage back and forth between words in the free group $\mathcal{F}(X)$ and elements of G/H, which are viewed as liftings in G. Given that ability and generators for G, one can construct $T \subseteq H$ such that $H = \langle T\rangle^G$. One can also build a presentation for G/K given presentations for G/H and H/K. We formalize these notions in this subsection.

For a set X, $\mathcal{F}(X)$ denotes the free group on X. Thus, given any may $\phi : X \to G$, there is a unique homomorphism $\hat{\phi} : \mathcal{F}(X) \to G$ extending ϕ.

Let $H \trianglelefteq G$. A *constructive presentation of G mod H* (or, if $H = 1$, a *constructive presentation of* G) is a 4-tuple $\Pi = (X, \phi, \psi, \mathcal{R})$ in which

X is a set; $\phi : X \to G$; $\psi : G \to \mathcal{F}(X)$; $\mathcal{R} \subset \mathcal{F}(X)$ such that

$g\big(\hat{\phi}\psi(g)\big)^{-1} \in H$ for all $g \in G$;

$\hat{\phi}^{-1}(H) = \langle \mathcal{R}\rangle^{\mathcal{F}(X)}$.

For computational purposes, it is assumed that $\Pi = (X, \phi, \psi, \mathcal{R})$ is input or output by

(i) Specifying $\phi(X)$ and $\mathcal{R}$.

(ii) Giving a procedure for determining $\psi(g)$, for any $g \in G$.

Remarks. (1) At the point of construction of ψ we do not necessarily have complete knowledge of the group G, perhaps not even a full generating set. It is only assumed that one can determine the images under $\hat{\phi}, \psi$ given elements of the respective domains.

(2) The procedure in (ii) does not need to output the full word. It would suffice to indicate a straight-line program (each step designating a product or inverse of predecessors) from X. Similarly, the words in $\mathcal{R}$ can be given by straight-line programs.

Lemma 4.1 *Given $\Pi = (X, \phi, \psi, \mathcal{R})$ as above, $\langle X|\mathcal{R}\rangle$ is a generator-relator presentation of the group G/H, with mutually-inverse isomorphisms $F(X)/\langle\mathcal{R}\rangle^{F(X)} \rightleftarrows G/H$ naturally induced by $\hat{\phi}, \psi$.*

For $g \in G$, we set $\text{Sift}_\Pi(g) = g\big(\hat{\phi}\psi(g)\big)^{-1}$.

Lemma 4.2 *If $G = \langle\phi(X)\rangle$ then $H = \langle\hat{\phi}(\mathcal{R})\rangle^G$. More generally, if $G = \langle S\rangle$, then $H = \langle\hat{\phi}(\mathcal{R}) \cup \text{Sift}_\Pi(S)\rangle^G$.*

The following lemma recalls, in our setting, a recipe for gluing together presentations for G/H, H/K.

Lemma 4.3 *Suppose that $\Pi = (X, \phi, \psi, \mathcal{R})$ is a constructive presentation of G mod H and $\Pi' = (X', \phi', \psi', \mathcal{R}')$ is a constructive presentation of H mod K where $K \trianglelefteq G$. Then $\Pi'' = (X'', \phi'', \psi'', \mathcal{R}'')$ is a constructive presentation of G mod K where*

$X'' = X \dot{\cup} X'$ *(disjoint union);*

$$\phi''(x) = \begin{cases} \phi(x), & \textit{for } x \in X \\ \phi'(x'), & \textit{for } x' \in X'; \end{cases}$$

$\psi''(g) = \psi'\big(\text{Sift}_\Pi(g)\big)\psi(g)$ *for* $g \in G$;

$$\mathcal{R}'' = \{r\big(\psi'\hat{\phi}(r)\big)^{-1} \mid r \in \mathcal{R}\} \cup \mathcal{R}'$$
$$\cup \{\big((x')^x\big)^{-1}\psi'\big(\phi'(x')^{\phi(x)}\big) \mid x \in X, x' \in X'\}.$$

If $G = \langle\phi(X)\rangle$ then $G = \langle\phi''(X'')\rangle$. Thus, if we start the tower method with a constructive presentation of

G mod G for which $G = \langle\phi(X)\rangle$, the condition persists through each G mod H_i.

4.3 Use of Invariant Subspaces

We assume that we have $H = \langle T\rangle^G$, with $H \neq 1$ (iff $T \not\subseteq \{1\}$) and that we seek some tractable representation $\pi : H \rightarrow M$. At various points in our procedures, we come into possession of a proper G-invariant subspace $W < V$. We can use that either to find π directly or to recurse to a "smaller" problem.

We have induced actions $\rho_1 : G \rightarrow GL(W)$, $\rho_2 : G \rightarrow GL(V/W)$. If, in either of these actions, H does not act trivially then we can recursively consider the problem for $\rho_i(G), \rho_i(T)$, which returns a representation $\pi' : \rho_i(H) \rightarrow M$, whereupon we set $\pi := \pi' \circ \rho_i$.

Suppose then that the actions of H on W and V/W are both trivial. There is then a natural nontrivial homomorphism π from H to (the additive group of) $\mathrm{Hom}(V, W)$ wherein $\pi(h)(v) = v^h - v$ for $v \in V$ and $h \in H$. There is also a natural action of G on $\mathrm{Hom}(V, W)$ such that $f^g(v) = \left(f(v^{g^{-1}})\right)^g$ for $f \in \mathrm{Hom}(V, W), v \in V, g \in G$. With this action π is then a G-map.

To determine $\pi(H) = \langle\pi(T)\rangle^G$ we need only observe that this is the smallest subspace of $\mathrm{Hom}(V, W)$ containing $\pi(T)$ and closed under the action of S. (Since q is prime, $\pi(H)$ is a vector space over q.)

It is an easy matter to write a constructive presentation for $\pi(H)$. Membership-testing in $\pi(H)$ is done via linear algebra.

Remark. Here, as in other uses of homomorphisms, the presence of a constructive membership test in $\pi(H)$ enables us to lift elements of $\pi(H)$ back to H. This is needed, for example, to lift the presentation of $\pi(H)$ to a presentation for H mod kernel(π).

We point out one useful source of invariant subspaces. For any $U \subseteq V$, $W = \mathrm{Fix}(\langle U\rangle^G)$ is G-invariant. Furthermore, given $G = \langle S\rangle$, we can find W as follows: $W := \mathrm{Fix}(U)$; **while** $\exists s \in S : W^s \neq W$ **do** $W := W \cap W^s$.

One application arises when $\langle U\rangle^G$ is unipotent, for a unipotent group has a nonzero fixed point.

We are also able to apply the idea to U for which $A = \langle U\rangle^G$ is abelian but not uniform. In such case, we can find m such that $0 < \mathrm{Fix}(A^m) < V$.

4.4 Abelian Groups

We consider the case of abelian $G = \langle S\rangle < GL(V)$.

The hypothesis on $\mu(G)$ enables us to determine $o(g)$ for any $g \in G$. (Besides q the primes that can occur in $|G|$ are less than $\mu(G)$; and we have an upper bound on the power to which they can occur, e.g., $\log_2 |GL(n, q)| < n^2 \log_2 q$). Hence, by forming suitable powers, we can express $g \in G$ as a product of elements of prime-power order. Doing this for each element of S and gathering the factors corresponding to each prime p dividing $|G|$, we get generators for the Sylow subgroups of G. Since G is the direct product of these, it suffices to show how to manage abelian p-groups. (For membership-testing, one first factors the candidate element.)

If G is q-group then it is unipotent and the reduction of §4.3 handles everything.

We assume then that G is a p-group, with $p \neq q$.

Recall that, at a generic step in the tower, we have $H = \langle T\rangle^G$. But, as G is abelian, this implies $H = \langle T\rangle$.

It suffices to give a procedure that constructs either a proper G-invariant subspace (thus leading into §4.3) or proof that $H = \langle h\rangle$, together with a method for expressing any $k \in H$ as a power of h. (A constructive presentation $\Pi = (\{x\}, \phi, \psi, \mathcal{R})$ could satisfy $\phi(x) = h$, $\psi(h) = x$, $\mathcal{R} = \{x^{o(h)}\}$.)

Lemma 4.4 *Let $G = \langle S\rangle < GL(V)$ be an abelian p-group with $p \neq q$ and $H = \langle T\rangle \leq Z(G)$. Then, in polynomial time, one can locate either a proper G-invariant subspace or prove that H is cyclic and locate a generator.*

Proof: It suffices to describe a polynomial-time procedure which, given $h, k \in H$, produces either a proper G-invariant subspace or else $z \in H$ such that $\langle h, k\rangle = \langle z\rangle$.

Without loss of generality $o(h) \geq o(k) \geq p$. Let $h_1 = h^{o(h)/p}, k_1 = k^{o(k)/p}$. If $\mathrm{Fix}(h_1) \neq 0$ then it is a proper G-invariant subspace. Else, recall that the abelian group $\langle h_1, k_1\rangle$ necessarily acts as a cyclic group on an $\langle h_1, k_1\rangle$-irreducible subspace V_0, and so k_1 necessarily acts on V_0 as h_1^r for some $0 \leq r < p$. Find r such that $\mathrm{Fix}(h_1^{-r}k_1) \neq 0$. If $\mathrm{Fix}(h_1^{-r}k_1) < V$ then it is a proper G-invariant subspace, else $k_1 = h_1^r$, in which case set $k' = h^{-ro(h)/o(k)}k$. Recurse for (h, k') (observing that $\langle h, k\rangle = \langle h, k'\rangle$ but $o(k') \leq o(k)/p$). ∎

If $H = \langle h\rangle$ the procedure in the above proof can be used to express any $k \in H$ as a power of h.

4.5 Nilpotent Groups

Knowing now that abelian groups are manageable, we turn to the case of nilpotent groups.

So, we assume we have nilpotent $G = \langle S\rangle < GL(V)$ and $H = \langle T\rangle^G$.

We first observe that we can test whether H is abelian (it is not sufficient for the elements of T to commute). One method involves the established membership-test of §4.4: since H is abelian, we can find generators for it (test whether $\langle T\rangle$ is closed under conjugacy by elements of S and, if not, find some new conjugate to add that increases $\langle T\rangle$). However, we offer a more generally-applicable test of commutativity of H.

In the following theorem, we do not assume either solvability of G or restrictions on $\mu(G)$.

Theorem 4.5 *Given $G = \langle S\rangle < GL(V)$ and $T \subset GL(V)$. Let $H = \langle T\rangle^G$. In polynomial-time, one can find (a basis of)* Span(H)*, in fact, a basis consisting of elements of H.*

Proof: Span(H) is the smallest subspace of Hom(V, V) that contains T and is closed under multiplication and under conjugation by elements of S. In performing these closures by multiplying and conjugating basis elements from each stage, it is easy to keep basis elements at all stages in H. ∎

Observe that $H = \langle T\rangle^G$ is abelian iff Span(H) is commutative iff a basis of Span(H) commutes. Hence this is polynomial-time testable. Furthermore, if H is not abelian, we find witnesses to the fact, namely, $h_1, h_2 \in H$ such that $[h_1, h_2] \neq 1$.

Thus, if H is not abelian, we can find $h_0 \in H \setminus Z(H)$. As each round puts us in a smaller term of the lower central series of G, the following procedure runs in polynomial-time: start with $h_0 \in H \setminus Z(H)$; **while** $\exists s \in S : [h_0, s] \notin Z(H)$ **do** replace h_0 by such $[h_0, s]$. Note that we test whether $[h_0, s]$ is in $Z(H)$ by seeing whether it commutes with Span(H).

When we exit from the above while-loop we have $h_0 \in H \setminus Z(H)$ but $[h_0, S] \subseteq Z(H)$ and the latter implies $[h_0, G] \subseteq Z(H)$.

These conditions on h_0 imply that the map $\pi : H \rightarrow Z(H)$ such that $\pi(h) = [h_0, h]$ is nontrivial homomorphism and a G-map. As the image group is of a manageable class (abelian), we take one step down the tower.

4.6 Finding Kernels, Centers

Before proceeding to the more general class of solvable groups, we indicate an application of the procedure.

Assume that we have a class $\mathcal{C}$ of groups such that for matrix groups G in $\mathcal{C}$, the problems of membership-testing and finding-constructive-presentations are in polynomial-time. We assume that the $\mathcal{C}$ is closed under taking of subgroups and homomorphic images.

Given $G = \langle S\rangle < GL(V) = GL(n, q)$ with $G \in \mathcal{C}$ and $T \subset GL(V)$, we can test whether $T \leq G$ and, if so, we can find $\langle T\rangle^G$ (adding conjugates until the subgroup is normal). Normal closures have numerous applications but we are particularly interested in the following.

Suppose we are also given a representation $\phi : G \rightarrow GL(W)$ (say by specifying $\phi(S)$), where W is a vector space over $GF(q)$. Then we can determine a constructive presentation for $\phi(G)$ and therefore for G mod kernel(ϕ). Using Lemma 4.2, we derive $T \subset G$ such that kernel$(\phi) = \langle T\rangle^G$, whence we find generators for the normal closure kernel(ϕ).

We can then apply this result to obtain $C_G(H)$ for $H = \langle T\rangle \trianglelefteq G$ since $C_G(H)$ is the kernel of the induced action (by conjugacy) of G on Span(H).

In particular, finding $Z(G)$ is in polynomial time.

Remark. We shall ultimately be able to find $C_G(T)$ for solvable $G < GL(V)$ (with polynomially bounded $\mu(G)$) and *arbitrary* $T \subset GL(V)$ (see §6.4)

4.7 Solvable Groups

We turn finally to solvable groups. Again, we have $G = \langle S\rangle$ and $H = \langle T\rangle^G$. Note that, if H is abelian, we are done by §4.4.

There is one problem reduction that is worth singling out first. Suppose that $1 < A < GL(V)$ with A abelian and normalized by G. Then, if q divides $|A|$, A has a unipotent part whose fixed points yield a proper G-invariant subspace. If, on the other hand, A is a q'-group then dim(Span$(A)) \leq n$ (since A would be diagonalizable over an algebraic extension of $GF(q)$). Thus, if we also had $1 < C_H(A) < H$, we could recursively consider the naturally-induced representations, via conjugacy, of G, H on Span(A) (a "smaller" problem as the image of H is isomorphic to $H/C_H(A)$).

It is easy to find abelian $A < H$ with $A^G = A$: let $1 \neq h \in H$; while $\langle h\rangle^G$ is nonabelian do replace h by a nontrivial element of $[\langle h\rangle^G, \langle h\rangle^G]$. The number of loop iterations is bounded by the length of the derived series of H, so that we emerge with an abelian $A = \langle h\rangle^G$. Unfortunately, this may not suffice for a problem reduction as above, for it could be that A is a q'-group and is centralized by H $(C_H(A) = H)$.

Consider, instead, the induced conjugacy-action of G on Span(H) and find $U \subset H$ such that $A = \langle U\rangle^G$ *acts as a nontrivial abelian group on* Span(H). Thus, $A \not\leq Z(H)$ but $[A, A] \leq Z(H)$.

If A is itself abelian, then it serves the aforementioned purpose as it is not centralized by H. If A is nonabelian then it is class-2 nilpotent, for $[A, A] \leq Z(A)$, so we can find full generators. Then, for some prime p, the Sylow p-subgroup, B, of A is a class-2 nilpotent p-group. (We can find B by extracting the p-parts of the generators of A.) We may assume $p \neq q$, else Fix(B) is a proper G-invariant subspace. As B is nilpotent, we may assume that we have a full set of generators. As an invariant of a loop to follow, we extract the following relevant data:

(*) B is a class-2 nilpotent p-group, normalized by G, and $1 < C_H(B) < H$.

Before proceeding with the algorithm, we make some observations about uniform (abelian) groups.

Lemma 4.6 *Suppose $Z < GL(V)$ is uniform. Let $\{W_i\}_{i \in I}$ be the maximal Z-invariant subspaces on which Z acts as a cyclic group. Then $V = \bigoplus_{i \in I} W_i$.*

To find these unique W_i when Z is a p-group: by repeated application of Lemma 4.4, find W for which Z^W is cyclic; find $K = C_Z(W)$ (§4.6) and set $W_1 = \text{Fix}(K)$; decompose $V = W_1 \oplus V'$ with a Z-invariant V' (since Z is a q'-group such decomposition exists - finding it is linear algebra); recursively decompose V.

Lemma 4.7 *Suppose $B < GL(V)$ is class-2 nilpotent and $Z(B)$ is cyclic and uniform. Then $|B : Z(B)| \leq n^2$.*

In fact, one shows that distinct coset representatives are linearly independent.

(LOOP) We pick up with B satisfying $(*)$.

We may assume $[H, Z(B)] = 1$, else we can utilize the induced actions of G, H on $\text{Span}(Z(B))$ ($Z(B)$ is computable, §4.6).

We may assume $Z(B)$ is uniform, else $\text{Fix}(Z(B)^{p^r})$ is a proper G-invariant subspace for some r.

With $Z = Z(B)$, decompose $V = W_1 \oplus \cdots \oplus W_r$ as in Lemma 4.6 (so $Z(B)^{W_i}$ is cyclic). Since $W_i = \text{Fix}(C_{Z(B)}(W_i))$ and $[H, C_{Z(B)}(W_i)] = 1$, W_i is H-invariant.

If, for any i, the inclusion $Z(B)^{W_i} \leq Z(B^{W_i})$ is strict: replace B by the strictly larger $B^{W_1} \times \cdots \times B^{W_r}$ acting naturally on $W_1 \oplus \cdots \oplus W_r$ and goto (LOOP) (noting that $(*)$ still holds as $C_H(\text{new}\, B) = C_H(\text{old}\, B)$).

Thus, we may assume that all $Z(B^{W_i})$ are cyclic and uniform, so that $|B^{W_i} : Z(B^{W_i})| \leq \dim(W_i)^2$. Set $\Omega = B^{W_1}/Z(B^{W_1}) \dot\cup \cdots \dot\cup B^{W_r}/Z(B^{W_r})$ (disjoint union), so $|\Omega| \leq n^2$. There is a natural action $\pi : G \to \text{Sym}(\Omega)$. If $\pi(H) \neq 1$, then we have our nontrivial representation in which to test membership and find a presentation.

Otherwise H acts trivially on Ω which implies $[H, B] \subset Z(B)$ and so H acts on B as an abelian p-group. Consider the induced conjugacy-action $\pi : G \to GL(\text{Span}(B))$. Then $\pi(H)$ is a nontrivial (since $C_H(B) < H$) abelian p-group and we appeal to §4.4.

5 Testing solvability and nilpotence

We focus on the limited task of testing nilpotence and solvability, for which we can discard the §4 restrictions on $\mu(G)$. However, the developments of §4 remain relevant.

As before, we may assume q is prime.

Testing nilpotence is almost immediate. The critical observation is that, for $q \neq p > n$, the Sylow p-subgroups of $GL(n, q)$ are necessarily abelian. Let Δ be the set of primes $\leq n$ together with q. Then we can express $|GL(n, q)| = ab$ where a only involves primes in Δ and b involves no primes in Δ.

Given a nilpotence candidate $G = \langle S \rangle$, let $S_a = \{s^b \mid s \in S\}$, $S_b = \{s^a \mid s \in S\}$. Then G is nilpotent iff: $\langle S_a \rangle$ is nilpotent and $[S, S_b] = 1$. Furthermore, if $\langle S_a \rangle$ is nilpotent then it involves only primes in Δ so that $\mu(\langle S_a \rangle) \leq n$.

For general solvability, we show, if G is solvable, that we are still able to "construct" a normal tower

$$G = H_1 \rhd H_2 \rhd \cdots \rhd H_m$$

together with manageable representations $\pi_i : H_i \to M_i$, with M_i known solvable, for $i < m$. But, in general, we shall not have $H_m = 1$. Nevertheless, H_m will be of a special form for which solvability is readily testable. In a sense the groups in the tower will not be completely constructed since we may never have full generators, though we shall have generators for H_i/H_m.

Suppose the tower is constructed down to H, i.e., we have a presentation for G/H, which is known to be solvable, and we can express $H = \langle T \rangle^G$.

We consider a class of groups H for which solvability is directly testable, i.e., without finding presentations, etc. We say H is *$\overline{q}$-triangulable* if the transformations in H can be simultaneously triangulated over the algebraic closure of $GF(q)$. Clearly $\overline{q}$-triangulable groups are solvable. Given T we can test whether $H = \langle T \rangle^G$ is $\overline{q}$-triangulable as follows: if H is abelian then output "yes" else find $1 \neq u \in [H, H]$; if $U = \langle u \rangle^G$ is not a q-group (unipotent) then output "no" else let $W := \text{Fix}(U)$; recursively test the action of H on W and V/W; if both calls respond "yes" then output "yes" else output "no".

Thus, with each new H, we perform the above test. If the answer is "yes", we announce G is solvable. Else we follow the procedure of §4.7. If we arrive at an invariant subspace W, we recurse on W if the action of H on W is not $\overline{q}$-triangulable else we recurse on V/W.

We must observe also that if the procedure goes through without finding an invariant subspace, then this round will succeed just as before for no "large" primes will appear in the induced representations. Indeed, we note that: in a representation $\pi : G \to GL(\text{Span}(A))$ for an abelian q'-group A, $\mu(\pi(G)) \leq n$; and the p corresponding to the class-2 nilpotent p-group B cannot exceed n.

6 Other Computational Problems

We have space herein to make only cursory comments about methods for the remaining problems in Theorem 3.2. Again we hypothesize that $\mu(G)$ is polynomially bounded.

6.1 An Algorithmic Paradigm

We indicate a divide-and-conquer paradigm for dealing with solvable matrix groups. Critical to this is

Theorem 6.1 *Given $G = \langle S \rangle \subset GL(n,q) = GL(V)$, with G solvable. In polynomial-time one can find at least one of the following.*

1. *A proper G-invariant subspace $W < V$.*
2. *$H \leq G$ and an imprimitivity system $V = W_1 \oplus \cdots \oplus W_m$ for H with $|G:H| < m^{c_0}$ (c_0 is an easily computable constant).*
3. *$A \leq G$ such that A is abelian and $|G:A| \leq \max(12, n)$.*

The proof involves some deeper (than §4) use of the structure of solvable matrix groups and will appear in a full version of this paper.

Theorem 6.1 offers an analogue to the divide-and-conquer (orbits, imprimitivity blocks) for solvable permutation groups that solves such problems as finding-set-stabilizers, finding-intersections and finding-centralizers ([Lu1]). It also incorporates the permutation-group machinery (in case 2).

The typical situation to which it applies involves a problem that requires computing a subcoset of G. In particular, the problem should have the following characteristics.

(i) Given a G-invariant subspace $W < V$, the problem can be solved in a polynomial number of steps together with recursive calls to induced problems on W and on V/W.

(ii) The problem is in polynomial time for abelian groups.

The general procedure in cases 2 and 3 involves writing G as union of cosets of the respective subgroups, splitting into subproblems accordingly. The case 2 situation works much like the method of [Lu1,§3], finding a minimal block system in the action on $\{W_1, \ldots, W_m\}$ then splitting the group into cosets of the stabilizer of the blocks. The polynomial timing ultimately depends on the polynomial bound on the size of primitive solvable permutation groups [Pá]. (The constant c_0 is related to that same bound.)

In the next two subsections we indicate two applications of this paradigm.

6.2 Vector Stabilizer

We address Theorem 3.2(5). To accommodate recursive calls to subspaces and decomposition into cosets, we consider a generalization.

A function $f : G \rightarrow V$ is called a *crossed homomorphism* if $f(xy) = f(x)^y + f(y)$ for $x, y \in G$. It is assumed that a procedure is given for computing f. (Although, given a constructive membership-test in G, it suffices to define f on the generators. We can also verify that such a specification defines a crossed homomorphism but that is not essential to the present application.) Consider the problem:

Given: f as above and $v \in V$.
Find: $\{g \in G \mid f(g) = v\}$.

In particular, $C_G(v)$ is the solution to $v^g - v = 0$ and $g \mapsto v^g - v$ is a crossed homomorphism.

Observe that the solution to $f(g) = v$ is either $\emptyset$ or a right coset of the subgroup $\{g \in G \mid f(g) = 0\}$.

Suppose that $W < V$ is G-invariant. Then, recursively working in V/W (i.e., modulo W) we solve $\overline{f(g)} = \overline{v}$. If nonempty the solution is Hy. Since $f(H) \subseteq W$ and $f(hy) = v$ iff $f(h) = (v - f(y))^{y^{-1}}$, we recursively solve the resulting problem in W. (When solving the "local" problem in some W, we make sure, in the resulting coset Hz, that H contains $C_G(W)$, the kernel of the action).

Suppose G is abelian. We first verify that $v \in W = \mathrm{Span}(f(G)) = (\mathrm{Span}(f(S))^G$, while computing a basis of W consisting of elements of the form $f(x)$, with $x \in G$. Since G is abelian, $f(g) = v$ implies $f(x)^g = f(x) + v^x - v$ for all $x \in G$. It is a constructive membership to find such a g (if one exists) and then the set of all solutions to the latter equation is Hg where $H = C_G(W)$. Finally, for $h \in H$, $f(hg) = v$ iff $f(h) = v - f(g)$, and solving this is linear algebra (as H acts trivially on W).

6.3 Subspace Stabilizer

The algorithm for Theorem 3.2(7) also uses the paradigm. For example, we point out how invariant subspaces can be utilized. First of all we need the following generalization of the problem:

Given: Subspaces $W_1, W_2 < V$.
Find: $\{g \in G \mid W_1^g = W_2\}$.

Suppose that $W < V$ is G-invariant. Recursively let $Hy = \{g \in G \mid (W_1 \cap W)^g = (W_2 \cap W)\}$ (unless the subanswer is $\emptyset$). Then H stabilizes $W_1 \cap W$. We seek $\{h \in H \mid W_1^h = W_3\}$ where $W_3 = W_2^{y^{-1}}$. Recursively working in V/W (i.e., modulo W) we solve $\overline{W_1}^h = \overline{W_3}$ for Jz (unless $\emptyset$). We still seek $\{j \in J \mid W_1^j = W_4\}$ where $W_4 = W_3^{z^{-1}}$, but we know that J stabilizes both $A = W_1 + W = W_4 + W$ and $B = W_1 \cap W = W_4 \cap W$. For $i = 1, 4$, let $E_i \in$Hom(A/B,A/B) be the projection map onto W_i/B ($A/B = W_i/B \oplus W/B$). Then for $j \in J$, $W_1^j = W_4$ iff $E_1^j = E_4$. The latter is solved in §6.2 (viewing the J action on the space $\mathrm{Hom}(A/B, A/B)$).

We omit the solution in the abelian G case (it can be reduced to dealing with idempotents as in the end of last paragraph).

6.4 Miscellany

We briefly comment on some other problems in §3.

It is easy to keep track of a composition series (Theorem 3.2(4)) in running the basic algorithm of §4.

The method for Theorem 3.2(6) goes beyond the paradigm of §6.1 as an additional divide-and-conquer is used on the set. It is analogous to Miller's generalization [Mil] of set-stabilizer to hypergraph stabilizer.

For Theorem 3.2(8), $C_G(x)$ is the stabilizer of the "vector" x in $\mathrm{Hom}(V,V)$.

To find $H \cap K$ (Theorem 3.2(9)), observe that, in the action of $H \times K$ on $V \oplus V$, the centralizer of the linear transformation mapping $(v,w) \mapsto (w,v)$ is $\{(h,h) \mid h \in H \cap K\}$.

Finding Sylow subgroups (Theorem 3.2(10)) follows the basic methods of Kantor [Ka1] (see also [Ma]), given the machinery developed in §4.

Corollary 3.3 involves reduction to subspace stabilizer problems and constructive Frattini arguments ([KL], [Ma]).

7 Open Questions

The ideas of §4 can be reorganized around reductions to finding-irreducible-subspaces and finding-discrete-logs (i.e., solving $\alpha^x = \beta$ in finite fields). We ask whether such a reduction can be made for other classes of matrix groups. For arbitrary matrix groups?

From another point of view, we ask whether membership-testing in matrix groups is polynomial in input+$\mu(G)$. Considering the result of [BBR], this would put membership-testing for finite matrix groups over algebraic number fields into polynomial time. (Finiteness of G is essential. Babai [Bab] has observed that, as a consequence of a result of Mihailova [Mih], testing membership in 4×4 integral matrix groups is undecidable.)

Acknowledgements

The author is indebted to W.M. Kantor and C.R.B. Wright for helpful discussions on these topics.

References

[At] M.D. Atkinson, ed., Computational Group Theory, Academic Press, 1984.

[Bab] L. Babai, *Trading group theory for randomness,* Proc. 17th ACM STOC, 1985, 421–429.

[Bac] E. Bach, *Number-theoretic algorithms,* in Annual Review of Computer Science, v.4 1989-1990, Annual Reviews Inc., 119–172.

[BBR] L. Babai, R. Beals, and D. Rockmore *Deciding finiteness of matrix groups in deterministic polynomial time,* Tech Rep, U. Chicago, 1992.

[BCFSL] L. Babai, G. Cooperman, L. Finkelstein, E. Luks and Á. Seress *Fast Monte Carlo algorithms for permutation groups,* 23rd ACM STOC, 1991, 90–100.

[BLS] L. Babai, E.M. Luks and Á. Seress, *Permutation groups in NC,* Proc. 19th ACM STOC, 1987, 409–420.

[BS] L. Babai and E. Szemerédi, *On the complexity of matrix group problems,* Proc. 24th IEEE FOCS, 1984, 229–240.

[Bu] G. Butler, *The Schreier algorithm for matrix groups,* in Proc. 1976 ACM Symp. on Symbolic and Algebraic Comp., 167–170.

[Ca] J.J. Cannon, *An introduction to the group theory language Cayley,* in Computational Group Theory (ed. M.D. Atkinson), Academic Press 1984, 145–183.

[FHL] M.L. Furst, J. Hopcroft and E.M. Luks, *Polynomial time algorithms for permutation groups,* Proc. 21th IEEE FOCS, 1980, 36–41.

[Ha] M. Hall, Jr., The Theory of Groups, Macmillan, New York 1959.

[Ka1] W.M. Kantor, *Sylow's theorem in polynomial time,* J. Comp. Syst. Sci. 30 (1985) 359-394.

[Ka2] W.M. Kantor, *Finding Sylow normalizers in polynomial time* J. Algorithms 11 (1990), 523–563.

[KL] W.M. Kantor and E.M. Luks, *Computing in quotient groups,* 22nd ACM STOC, 1990, 524–534.

[Le] J. Leech, ed., Computational Problems in Abstract Algebra, Pergamon, NY, 1970.

[Lu1] E.M. Luks, *Isomorphism of graphs of bounded valence can be tested in polynomial time,* J. Comp. Syst. Sci. 25 (1982), 42–65.

[Lu2] E.M. Luks, *Parallel algorithms for permutation groups and graph isomorphism,* Proc. 27th IEEE FOCS 1986, 292–302.

[LM] E.M. Luks and P. McKenzie, *Parallel algorithms for solvable permutation groups,* J. Comp. Syst. Sci. 37 (1988), 39–62.

[Ma] P.D. Mark, *Sylow's theorem and parallel computation,* Doctoral dissertation, CIS, U.Oregon,1992.

[Mih] K.A. Mihailova, *The occurrence problem for direct products of group,* Mat. Sb. (N.S.) 70(112) (1966), 241–251.

[Mil] G.L. Miller, *Isomorphism of graphs which are pairwise k-separable,* Inform. and Control 56 (1983), 21–33.

[Pá] P. Pálfy, *A polynomial bound on the orders of primitive solvable groups,* J. Algebra, 78 (1982), 127–137.

[Ró] L. Rónyai, *Computing the structure of finite algebras,* J. Symbolic Computation 9 (1990), 355–373.

[Sc] M. Schönert et al, GAP: Groups, Algorithms and Programming, Aachen, 1992.

[Si] C.C. Sims, *Some group-theoretic algorithms,* Springer Lect. Notes in Math. 697 (1978), 108–124.

[Su] D.A. Suprunenko, Matrix Groups, AMS, 1976.