Journal of Computer and System Sciences **58**, 260–279 (1999)
Article ID jcss.1998.1614, available online at http://www.idealibrary.com on IDEAL®

Algorithms for Matrix Groups and the Tits Alternative

Robert Beals*

Department of Mathematics, University of Arizona, Tucson, Arizona 85712
E-mail: beals@math.arizona.edu

Received December 5, 1995; revised July 9, 1997

Tits has shown that a finitely generated matrix group either contains a nonabelian free group or has a solvable subgroup of finite index. We give a polynomial time algorithm for deciding which of these two conditions holds for a given finitely generated matrix group over an algebraic number field. Noting that many computational problems are undecidable for groups with nonabelian free subgroups, we investigate the complexity of problems relating to matrix groups with solvable subgroups of finite index. For such a group G, we are able in polynomial time to compute a homomorphism ϕ such that $\phi(G)$ is a finite matrix group and the kernel of ϕ is solvable. If, in addition, G has a nilpotent subgroup of finite index, we obtain much stronger results. For such groups, we show how to effectively compute an encoding of elements of G such that the encoding length of an element obtained as a product of length $\leqslant \ell$ over the generators is $O(\log \ell)$ times a polynomial in the input length. This result is the best possible; it has been shown by Tits and Wolf that if a finitely generated matrix group does not have a nilpotent subgroup of finite index, then the number of group elements expressible as words of length ℓ over the generators grows as c^{ℓ} for some constant $c > 1$ depending on G. For groups with abelian subgroups of finite index, we obtain a Las Vegas algorithm for several basic computational tasks, including membership testing and computing a presentation. This generalizes recent work of Beals and Babai, who give a Las Vegas algorithm for the case of finite groups, as well as recent work of Babai, Beals, Cai, Ivanyos, and Luks, who give a deterministic algorithm for the case of abelian groups. © 1999 Academic Press

[*] Research conducted while visiting IAS and DIMACS and supported in part by an NSF Mathematical Sciences Postdoctoral Fellowship and by the Alfred P. Sloan foundation. These results have been announced previously in *Proc. 36th IEEE FOCS, 1995*, pp. 593–602.

1. INTRODUCTION

1.1. Groups and Computation

For several decades, computers have been used for group theoretic investigations. The design and implementation of practical algorithms for group theory is the domain of computational group theory (cf. [Atk84]). Such algorithms have been developed for permutation groups [Si1] (cf. [Bu]), finitely presented groups (cf. [Si2]), and, more recently, finite matrix groups [NP, HR, H + 96a, H + 96b]. Many of these algorithms have been implemented in popular programming packages [Can, Sch +], which are used by hundreds of researchers in group theory, design theory, coding theory, and other areas. Methods for handling infinite matrix groups, particularly abelian, nilpotent, and solvable groups, are currently being developed by Ostheimer [Os].

The study of the asymptotic complexity of group theoretic algorithms is much newer than computational group theory, with the oldest papers dating back only 15–20 years [LSZ, LZ, Ba79, FHL]. As is the case with computational group theory, the area of permutation groups is particularly well studied [Ba91a, KLu, Lu93]. Polynomial time algorithms for several classes of finite matrix groups have been obtained in the last few years [Lu92, BBR, BB]. (Some important complexity results for these groups were obtained 10 years ago [BSz, Ba85].) For infinite matrix groups, polynomial time algorithms are known only for the class of abelian groups [KLi, Ge93a, Ge93b, CLZ, BBCIL].

This research has benefited theoretical computer science in several ways. One of the motivations for the invention of interactive proofs was Babai's investigation of complexity in finite matrix groups [Ba85]. Much of our understanding of the complexity of graph isomorphism comes from research in permutation group algorithms [Ba79, FHL, Lu82]. Numerous applications have been found in cryptography [BCY, BY, IY, M, MM89, MM92].

Asymptotic complexity theory has also benefited computational group theory. While “polynomial time” and “practical” are two different notions, often the structure used to design a polynomial time algorithm can be refined to yield a practical algorithm. For example, Luks's polynomial time algorithm for computing the composition factors of a permutation group [Lu87] introduced the basic approach used by several improved algorithms [BLS87, BLS88, BS]. An implementation of the [BS] algorithm is included with the group theory package GAP [Sch +].

This paper is an initial investigation into algorithms for infinite nonabelian matrix groups. We give a number of polynomial time algorithms for problems which were not previously known to be efficiently decidable. These algorithms are unlikely to be practical in their current form except for small dimensions. However, the results are new, even for fixed dimension. We hope that the approach introduced here will serve as a guide for future research, both theoretical and practical, in this area.

1.2. Matrix Groups: Problems and Obstacles

We consider the basic computational problem of testing membership in finitely generated matrix groups over number fields. Membership testing naturally reduces to two problems which are better suited for divide-and-conquer techniques: constructive membership testing, and computing a presentation. *Constructive membership testing* for a group G generated by a set S is the task of expressing a given g as a straight line program over S, or determining that $g \notin G$. A *presentation* for a group is a set of abstract generators and relations which define the group. These two problems are naturally occurring subproblems for many group computations, including (nonconstructive) membership testing (see, for example, [Lu92]).

In recent years, algorithms have been obtained for two important classes of groups: finite groups and abelian groups. For finite groups, Beals and Babai [BB] give Las Vegas algorithms for constructive membership and computing a presentation. Also, Babai, Beals, and Rockmore [BBR] have given a deterministic polynomial time algorithm to test finiteness. For abelian groups, Babai, Beals, Cai, Ivanyos, and Luks [BBCIL] give deterministic algorithms for both problems, building on the work of Cai, Lipton, and Zalcstein [CLZ] and Ge [Ge93a, Ge93b]. Ostheimer [Os] has developed similar methods independently.) We extend both of these lines of research, giving Las Vegas algorithms for groups with abelian subgroups of finite index. We obtain weaker results for two larger classes of groups: those with a nilpotent subgroup of finite index and those with a solvable subgroup of finite index.

We briefly recall the definitions of solvable and nilpotent groups. A group G is *solvable* if the *derived series* $G = G^0 \geqslant G^1 \geqslant \cdots$ reaches the identity in a finite number of steps, where G^{i+1} is the subgroup of G^i generated by all elements of the form $x^{-1}y^{-1}xy$ with $x, y \in G^i$. A group G is *nilpolent* if the *lower central series* $G = G_0 \geqslant G_1 \geqslant \cdots$ reaches the identity in a finite number of steps, where G_{i+1} is the group generated by elements of the form $x^{-1}y^{-1}xy$, where $x \in G$ and $y \in G_i$.

For general matrix groups, the membership problem is *undecidable*, even for 4×4 integer matrix groups [Mih]. Also, the problem of testing membership in a normal subgroup given by normal generators is undecidable for 2×2 matrix groups. These undecidability results stem from the presence of nonabelian free subgroups of $GL(n, \mathbf{Z})$. A group generated by a set S is *free* if no nonempty reduced word in the generators and their inverses represents the identity. (A word is reduced if it contains no substrings of the form gg^{-1} or $g^{-1}g$.) An important question, then, from the point of view of decidability, is which matrix groups contain nonabelian free subgroups? In a celebrated result, Tits [Ti] has shown that a finitely generated matrix group either has a nonabelian free subgroup or a solvable subgroup of finite index (the two conditions are mutually exclusive). This dichotomy is known as the "Tits alternative."

1.3. Main Results

We give an algorithm to decide the Tits alternative. While we solve it in polynomial time, we should point out that even the decidability of this problem is not immediate and it does not appear to have been known.

Recall that $GL(n, F)$ denotes the group of invertible $(n \times n)$ matrices over F, and $\leqslant$ stands for "subgroup."

Theorem 1.1. *Let F be a symbolically given algebraic number field, and let $G \leqslant GL(n, F)$ be a finitely generated group given by a list of generators. Then it* *is decidable in polynomial time whether or not G has a solvable subgroup of finite index.*

Combining this with the Tits alternative, we have

Corollary 1.2. *It is decidable in polynomial time whether or not $G \leqslant GL(n, F)$ has a nonabelian free subgroup.*

A group with a solvable subgroup of finite index necessarily has a solvable normal subgroup of finite index (a subgroup is *normal* if it occurs as the kernel $\ker(\phi)$ of a homomorphism ϕ). Such a group is called *solvable-by-finite*. More generally, if X and Y are classes of groups, then a group is X-by-Y if it has a normal subgroup from the class X such that the quotient belongs to Y. In this paper we shall be especially concerned with the following three classes of groups: solvable-by-finite, nilpotent-by-finite, and abelian-by-finite.

For solvable-by-finite groups, we obtain the following.

Theorem 1.3. *Let F be a symbolically given algebraic number field, and let $G \leqslant GL(n, F)$ be a finitely generated group given by a list of generators. Then in polynomial time, for some $m \leqslant n^2$, we can either compute a homomorphism ϕ: $G \to GL(m, F)$ such that $\phi(G)$ finite and* $\ker(\phi)$ *is solvable, or determine that G is not solvable-by-finite.*

By considering the image of ϕ, this reduces the problem of determining whether or not G is solvable to the finite case. Luks [Lu92] has given deterministic polynomial time algorithms for a host of problems relating to finite solvable matrix groups, including solvability testing. Thus, we have

Corollary 1.4. *Solvability of finitely generated matrix groups over number fields is decidable in polynomial time.*

We also provide, in Section 4, a Monte Carlo solvability test. This algorithm is fairly simple, but the error analysis is new and depends on the same structure theory which we develop for Theorems 1.1 and 1.3.

For groups which are nilpotent-by-finite, we obtain stronger results, reducing most problems to the unipotent case. A matrix is *unipotent* if its minimal polynomial is a power of $(x-1)$. A matrix group is unipotent if it consists of unipotent matrices. A unipotent matrix group is necessarily nilpotent.

Theorem 1.5. *Let F be a symbolically given algebraic number field, and let $G \leqslant GL(n, F)$ be a finitely generated group given by a list of generators. We can decide in polynomial time whether or not G is nilpotent-by-finite, and if so we can compute a homomorphism ϕ of G with the properties:*

1. $\ker(\phi)$ *is unipotent.*
2. $\phi(G)$ *is abelian-by-finite.*
3. *Constructive membership testing in $\phi(G)$ can be performed in Las Vegas polynomial time.*

In addition, normal generators for $\ker(\phi)$ *can be computed in Las Vegas polynomial time.*

For finitely generated abelian unipotent groups, [BBCIL] shows how to compute an isomorphism to a lattice. This, combined with known polynomial time algorithms for lattices (cf. [Schr]), gives

COROLLARY 1.6. *Let F be a symbolically given algebraic number field, let $G \leqslant GL(n, F)$ be a finitely generated group given by a list of generators, and let $X \in GL(n, F)$ be a given matrix. We can decide in polynomial time whether or not G is abelian-by-finite, and if so we can determine whether or not $X \in G$ in Las Vegas polynomial time.*

In working with matrix groups in characteristic 0, a problem even more basic than membership testing is that of estimating bit lengths of elements. This is nontrivial even for finite groups [Ba92, BBR]. In general, it is often desirable to represent elements of a group using an encoding other than the standard encoding (in which an element of $GL(n, F)$ is represented as an $n \times n$ array of elements of F). Let size(x) denote the bit length of x. For $G \leqslant GL(n, F)$ given by a generating set S and for technical reasons, we assume that the encoding of S includes a description of F, so size(S) is the input length.

Let G be a finitely generated group, generated by the finite set S. The *growth function* of G with respect to S is the function g_S, where $g_S(m)$ is the number of distinct elements of G expressible as words of length m in G. The asymptotic behavior of g_S does not depend on the choice of generating set S. We say that G has *polynomial growth* if $g_S(m) = O(m^c)$ for some constant c, and we say that G has *exponential growth* if $g_S(m) = \Omega(c^m)$ for some constant $c > 1$. Milnor and Wolf (cf. [Mil, Wo]) have shown that a solvable-by-finite group G has polynomial growth if G is nilpotent-by-finite and has exponential growth otherwise. Combined with the Tits alternative, this shows that a finitely generated matrix group G has polynomial growth if G is nilpotent-by-finite and has exponential growth otherwise.

Groups with exponential growth seem difficult to handle algorithmically. After performing t group operations (starting from the generators), it is possible to obtain a group element which is only expressible as an exponentially long (in t) word in the generators. If the group has exponential growth, then the number of elements expressible as exponentially long words is doubly exponential in t, so a typical such element requires exponential space just to encode.

We define a *slow-growth encoding* of a group G to be an encoding ζ of elements of G such that for any $g_1, ..., g_\ell \in G$, size$(\zeta(\prod_{i=1}^{\ell} g_i))$ is at most

$$O(\max_i \{\text{size}(\zeta(g_i))\} \cdot \log \ell).$$

A trivial counting argument shows that a finitely generated group has a slow-growth encoding only if it has polynomial growth. Conversely, a group G with polynomial growth has a slow-growth encoding assume that $G = \langle S \rangle$ is totally ordered, where S is finite, contains 1, and is closed under inverses. Then encode the ith smallest element of $S^j \backslash S^{j-1}$ by the ordered pair i, j (using any standard encoding for ordered pairs of natural numbers). Thus, the encoding length is polynomially related to the logarithm of the word length over S and is therefore slow-growth.

However, this encoding seems difficult to work with, in practice: how does one compute i and j?

We say that an encoding ζ of a group G is an *effective slow-growth encoding* if there is a supergroup $\mathscr{G}$ of G such that the following conditions all hold:

1. Membership of g in $\mathscr{G}$ is decidable in polynomial time.
2. For $g \in \mathscr{G}$, we can compute $\zeta(g)$ in polynomial time.
3. Given $\zeta(g_1)$ and $\zeta(g_2)$, we can compute $\zeta(g_1 g_2^{-1})$ in polynomial time.
4. ζ is a slow-growth encoding. That is, for any $g_1, ..., g_\ell \in \mathscr{G}$,

$$\operatorname{size}\left(\zeta\left(\prod_{i=1}^{\ell} g_i\right)\right) \leqslant O(\max_i \{\operatorname{size}(\zeta(g_i))\} \cdot \log \ell).$$

Note that the "polynomial" refered to in conditions 1–3, as well as the constant hiding behind the big-Oh in 4, all depend on the group G. That is, for a fixed G (and $\mathscr{G}$ and ζ) the complexity of the computational tasks in 1–3 is polynomial in the input length. We say that a family $(G_i, \zeta_i)_{i \in I}$ of groups G_i with effective slow-growth encodings ζ_i is *uniform* if, given G_i by a generating set S, complexity of these tasks is polynomial in size(S) and the input length, and the constant in 4 is $O(\operatorname{size}(S)^c)$ for some constant c.

When computing with an effective slow-growth encoding, if polynomially many group operations are performed, then the total time for these group operations is polynomially bounded. So it would be nice if effective slow-growth encodings existed for all groups with polynomial growth rate. We show that this is the case for matrix groups over number fields:

THEOREM 1.7. *Let F be a symbolically given algebraic number field, and let $G \leqslant GL(n, F)$ be a finitely generated group given by a list S of generators. We can determine in polynomial time whether the growth rate of G is polynomial or exponential. If G has polynomial growth, then in polynomial time we can find an effective slow-growth encoding ζ of G. These encodings are uniform for the family of nilpotent-by-finite matrix groups over F.*

Previously, it was known how to compute such a ζ for finite matrix groups [BBR] and for abelian matrix groups [BBCIL].

1.4. Methods

A matrix group $G \leqslant GL(V)$ is *primitive* if there is no direct sum decomposition $V_1 + \cdots + V_r$ of V such that G permutes the V_i. If such direct sum decompositions could be found for imprimitive groups, many computational problems would be reduced to the primitive case. Much is known about the structure of primitive solvable matrix groups [Su63, Su76]. However, no general imprimitivity algorithm is known (although Holt *et al.* [H+96b] have developed practical methods for matrix groups over finite fields).

A natural combinatorial approach to matrix group imprimitivity is suggested by Atkinson's algorithm for permutation group imprimitivity [Atk75]. For permutation groups, a system of imprimitivity can be found by considering the orbits of a representation of degree n^2. The analogous approach for matrix groups would consider the invariant subspaces of a representation of dimension n^2, but it is not clear that this works in all cases. Nevertheless, by using a novel application of finiteness testing [BBR], we obtain a method which is sufficiently general for our purposes.

We find enough of the imprimitivity structure to decide the Tits alternative, and with some further refinements, we obtain our results for nilpotent-by-finite groups as well. This allows a reduction of the abelian-by-finite case to the abelian and the finite cases separately, for which algorithms exist (cf. [BBCIL, Ge93a, Os, BBR, BB]).

2. PRELIMINARIES AND BACKGROUND

2.1. Building Blocks

We give some simple examples of effective slow-growth encodings, as well as some methods by which such encodings may be combined to create new ones. Babai, Beals, and Rockmore [BBR] have shown

THEOREM 2.1. *Let $G \leqslant GL(n, \mathbf{Q})$ be given by a finite list of generators. Then it is decidable in deterministic polynomial time whether or not G is finite. If G is finite, then in polynomial time we can compute a G-invariant n-dimensional lattice L and a G-invariant positive definite quadratic form B such that the group G of all elements of $GL(n, \mathbf{Q})$ which fix L and B, together with the standard encoding of rational matrices, is an effective slow-growth encoding.*

The generalization to finite subgroups of $GL(n, F)$ is obtained by considering F as a vector space over $\mathbf{Q}$.

The standard encoding is also sufficient for additive subgroups of a vector space.

PROPOSITION 2.2. *Let F be a symbolically given algebraic number field. Then the standard encoding of elements of F^n is a slow-growth encoding when restricted to any finitely generated subgroup.*

Unipotent groups are yet another example.

LEMMA 2.3. *Let A be an n-dimensional algebra with unit element* 1 *and radical R. Let $\mathcal{G}$ be the multiplicative group of elements of the form $1 + r$ with $r \in R$. Then the standard encoding of elements of $\mathcal{G}$ is slow-growth.*

Proof. By Friedl–Rónyai, the radical R is polynomial time computable, so $\mathcal{G}$ is polynomial time recognizable (condition 1). Conditions 2–3 are trivial. It remains to show that condition 4 holds. Consider a product $g = g_1 g_2 \cdots g_\ell$. Let $g_i = 1 + r_i$ for $1 \leqslant i \leqslant \ell$. Then g can be expressed as a sum of products of the r_i. Since any product of n or more of the r_i is zero, we only need to consider products of length less than n. The number of summands is thus $O(\ell^n)$, each of which has bit length at most $O(n^c \max_i \operatorname{size}(r_i))$. Applying Proposition 2.2 gives the desired bound on the bit length of g. ▮

We now describe several methods of combining encodings.

PROPOSITION 2.4. *Let $\mathcal{G}_1, \zeta_1$ and $\mathcal{G}_2, \zeta_2$ be groups with slow-growth encodings. For (g_1, g_2) in the direct product $\mathcal{G}_1 \times \mathcal{G}_2$, let $\zeta(g_1, g_2) = (\zeta_1(g_1), \zeta_2(g_2))$. Then $\mathcal{G}_1 \times \mathcal{G}_2, \zeta$ is a slow-growth encoding.*

Let G and H be groups, with G acting (on the left) on H as automorphisms. Then the *semidirect product* $H \rtimes G$ is defined as the set of all ordered pairs $\{(h, g) \mid h \in H,\ g \in G\}$, with multiplication defined by

$$(h_1, g_1)(h_2, g_2) = (h_1 \cdot g_1(h_2), g_1 g_2).$$

With this operation, $H \rtimes G$ is a group. It is not the case that, given slow-growth encodings for G and H, we can construct a slow-growth encoding for $H \rtimes G$, since $H \rtimes G$ might have exponential growth. However, we do have

PROPOSITION 2.5. *Let $\mathcal{G}_1, \zeta_1$ and $\mathcal{G}_2$, ζ_2 be groups with slow-growth encodings. Suppose that $\mathcal{G}_2$ acts on $\mathcal{G}_1$ as automorphisms such that the maximum bit length of $\zeta_1(g_2(g_1))$ for $g_2 \in \mathcal{G}_2$ is polynomially bounded by $\zeta_1(g_1)$ and the bit length of the generating set for $\mathcal{G}_2$. Then the semidirect product $\mathcal{G}_1 \rtimes \mathcal{G}_2$ with the encoding $\zeta(g_1, g_2) = (\zeta_1(g_1), \zeta_2(g_2))$ is a slow-growth encoding.*

This condition is very restrictive, but we give three important examples.

EXAMPLE 1. Suppose that $\mathcal{G}_1$ is a matrix group, and $\mathcal{G}_2$ acts linearly as a finite group on $\operatorname{Env}(\mathcal{G}_1)$. Then slow-growth encodings for $\mathcal{G}_1$ and $\mathcal{G}_2$ may be combined to obtain a slow-growth encoding for $G_1 \rtimes G_2$.

EXAMPLE 2. Suppose that $\mathcal{G}_1$ is a lattice, and $\mathcal{G}_2$ acts linearly as a finite group on $\mathcal{G}_1$. Then slow-growth encodings for $\mathcal{G}_1$ and $\mathcal{G}_2$ may be combined to obtain a slow-growth encoding for $G_1 \rtimes G_2$.

Let S_r denote the symmetric group of degree r.

EXAMPLE 3. Suppose that $\mathcal{G}_1$ is an iterated direct product $\mathcal{G}^r$, and $\mathcal{G}_2 = S_r$ acts by permuting the factors in the direct product. Suppose that ζ_1 is obtained from a slow-growth encoding of $\mathcal{G}$ by repeated applications of Proposition 2.4. Then slow-growth encodings for $\mathcal{G}_1$ and $\mathcal{G}_2$ may be combined to obtain a slow-growth encoding for $\mathcal{G}_1 \rtimes \mathcal{G}_2$.

In this last example, the group $\mathcal{G}_1 \rtimes \mathcal{G}_2$ is also called the *wreath product* of $\mathcal{G}$ by S_r, denoted $\mathcal{G} \wr S_r$.

2.2. Algebra Decomposition

Let F be a field, and let $G \leqslant GL(n, F)$. Then we denote by $\operatorname{Env}(G)$ the *enveloping algebra* of G, i.e., the set of F-linear combinations of elements of G. We denote by $M_n(F)$ the algebra of $n \times n$ matrices over F.

Let $\mathscr{A}$ be an algebra over the field F. The *radical* $\mathrm{Rad}(\mathscr{A})$ is the largest nilpotent ideal of $\mathscr{A}$. The quotient $\mathscr{A}/\mathrm{Rad}(\mathscr{A})$ is *semisimple* i.e., its radical is 0. If $\mathscr{A}$ and $\mathscr{B}$ are subalgebras of an algebra, then the *centralizer* of $\mathscr{A}$ in $\mathscr{B}$, denoted $C_{\mathscr{A}}(\mathscr{B})$, is the set $\{a \in \mathscr{A} \mid \forall b \in \mathscr{B} : ab = ba\}$. If $\mathscr{A} = \mathscr{B}$ this is the *center* of $\mathscr{A}$, denoted $Z(\mathscr{A})$. Centralizers can be found in polynomial time by solving linear equations.

There is a nice correspondence between the structure of $\mathrm{Env}(G)$ and the action of G on F^n. If $\mathrm{Env}(G)$ is the direct sum of ideals I_1 and I_2, then F^n is the direct sum of G-invariant subspaces I_1F^n and I_2F^n. In addition, the action of G is *completely reducible* (i.e., F^n is the direct sum of G-invariant subspaces on which G acts irreducibly) iff $\mathrm{Env}(G)$ is semisimple.

We require the result of Friedl and Rónyai [FR].

Theorem 2.16 [Fr]. *Let $\mathscr{A}$ be a subalgebra of $M_n(F)$, given by a list of generators. Then $\mathrm{Rad}(\mathscr{A})$ can be calculated in polynomial time, along with the minimal ideals of $\mathscr{A}/\mathrm{Rad}(\mathscr{A})$.*

We indicate a way in which the Friedl–Ronyai result can be used as a divide-and-conquer tool. Once the decomposition of $\mathrm{Env}(G)/\mathrm{Rad}(\mathrm{Env}(G))$ is known, a basis can easily be chosen for F^n such that the matrices $g \in G$ have the form

$$\begin{pmatrix} \eta_1(g) & * & \cdots & * \\ 0 & \eta_2(g) & \cdots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \eta_r(g) \end{pmatrix},$$

where each $\mathrm{Env}(\eta_i(G))$ is a simple algebra. Let η denote the representation:

$$\eta(g) = \begin{pmatrix} \eta_1(g) & 0 & \cdots & 0 \\ 0 & \eta_2(g) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \eta_r(g) \end{pmatrix}.$$

Then the kernel of η is solvable (it consists of unipotent matrices), so G is solvable-by-finite iff $\eta(G)$ is solvable-by-finite. We will make use of the homomorphism η throughout the paper.

An element e of an algebra is an *idempotent* if $e^2 = e$. The Friedl–Rónyai algorithm actually calculates for each ideal I of $\mathscr{A}/\mathrm{Rad}(\mathscr{A})$ an idempotent e in the center of $\mathscr{A}/\mathrm{Rad}(\mathscr{A})$ such that e is the multiplicative identity of I. For two distinct such ideals, I_1, I_2, we have $e_1e_2 \in I_1 \cap I_2 = \{0\}$, so $e_1e_2 = 0$.

The classical theorem of Wedderburn shows that $\mathscr{A}/\mathrm{Rad}(\mathscr{A})$ is the direct sum of simple algebras. These simple components are precisely the minimal ideals found by the Friedl–Rónyai algorithm. If $\mathscr{A}$ is the enveloping algebra of a nilpotent group, then something more can be said. We first quote the following result of Suprunenko [Su76, p. 223, Theorem 2].

Theorem 2.7. *Let F be an arbitrary field. The irreducible components of a nilpotent indecomposable subgroup of $GL(n, F)$ are pairwise equivalent.*

COROLLARY 2.8. *Let $G \leqslant GL(n, F)$ be nilpotent, and let $\mathscr{A} = \mathrm{Env}(G)$. Then there exist ideals $\mathscr{A}_1, ..., \mathscr{A}_r$ such that $\mathscr{A}$ is the direct sum of the $\mathscr{A}_i$ and each $\mathscr{A}_i/\mathrm{Rad}(\mathscr{A}_i)$ is simple.*

We quote two results of [BBCIL].

LEMMA 2.9. *Given a matrix $a \in M_n(F)$, we can in polynomial time find matrices a_d and a_n in the algebra generated by a, such that a_d is diagonalizable, (over an extension field), a_n is nilpotent, and $a = a_d + a_n$.*

COROLLARY 2.10. *Let $\mathscr{A}$ be a commutative algebra over the number field F, such that $K = \mathscr{A}/\mathrm{Rad}(\mathscr{A})$ is simple. Then in polynomial time, we can find a subalgebra of $\mathscr{A}$ isomorphic to K.*

Note that K is an extension field of F. This corollary will allow us to view F^n as a vector space (of smaller dimension) over K.

We quote below some well-known results on group algebras (cf. [We, Chap. 1]).

THEOREM 2.11. *Suppose that $H \leqslant G \leqslant GL(n, F)$, and $G : H$ is finite. Then if* $\mathrm{Env}(H)$ *is semisimple, so is* $\mathrm{Env}(G)$.

THEOREM 2.12 (Clifford). *Suppose that $G \leqslant GL(n, F)$ with* $\mathrm{Env}(G)$ *semisimple. Then for any normal subgroup N of G,* $\mathrm{Env}(N)$ *is semisimple.*

3. STRUCTURE OF SOLVABLE-BY-FINITE GROUPS

In this section we consider in some detail what it means for a group to have a solvable subgroup of finite index. The results presented here are nonconstructive, although algorithmic versions will be presented in later sections.

Suppose that the group $G \leqslant GL(n, F)$ has a solvable subgroup H of finite index. Then the intersection of the conjugates of H is a normal subgroup N of finite index, so G is solvable-by-finite.

A matrix group is *triangularizable* if, over some extension field K of F, all elements of the group may be simultaneously put in upper triangular form. A theorem of Mal'cev (cf. [We, Theorem 3.6]) states that a solvable matrix group has a triangularizable normal subgroup of finite index. Let T be a triangularizable normal subgroup of N of finite index. Then the intersection of the G-conjugates of T is a triangularizable normal subgroup M of G of finite index in G. Without loss of generality we suppose that $M = N$.

Now suppose that $\mathrm{Env}(G)$ is semisimple. Then $\mathrm{Env}(N)$ is semisimple as well by Clifford's theorem (Theorem 2.12). Therefore N is abelian (it is diagonalizable over an extension field). We suppose that for any positive integer ℓ, the groups N and $N^\ell = \{x^\ell \mid x \in N\}$ have the same enveloping algebra (otherwise replace N by N^ℓ; this can only happen finitely many times before we arrive at a subgroup with the desired property).

Let $\theta \colon G \to GL(M_n(F))$ denote the conjugation action of G on $M_n(F)$. It clear that $\mathrm{Env}(\theta(N))$ is semisimple, so by Theorem 2.11, $\mathscr{A} = \mathrm{Env}(\theta(G))$ is semisimple as well. Let $\mathscr{A}_1, ..., \mathscr{A}_s$ be the decomposition of $\mathscr{A}$ into a direct sum of minimal ideals, and let $W_i = \mathscr{A}_i M_n(F)$ (the W_i are subspaces of $M_n(F)$, but not necessarily subalgebras). Let $\mathscr{B}$ be the direct sum of those W_i such that the restriction of $\theta(G)$ to W_i is finite. Note that $\mathscr{B}$ is a subalgebra of $M_n(F)$. Let $\theta_{\mathscr{B}}$ be the conjugation action of G on $\mathscr{B}$, and note that $\theta_{\mathscr{B}}(G)$ is finite.

Lemma 3.1. *Let $\mathscr{B}$ be as above. Then $\mathscr{B} = C_{M_n(F)}(\mathrm{Env}(N))$, and $\mathrm{Env}(N) = C_{M_n(F)}(\mathscr{B})$.*

Proof. We first show that $C_{M_n(F)}(\mathrm{Env}(N)) \subseteq \mathscr{B}$. Suppose a centralizes $\mathrm{Env}(N)$. We wish to show that a has finitely many G-conjugates. Since N has finite index in G, this is equivalent to showing that a has finitely many N-conjugates. This is immediate, since $a^N = \{a\}$.

Next, we show that $\mathscr{B} \subseteq C_{M_n(F)}(N)$. Suppose that a does not centralize N. We wish to show that $a \notin \mathscr{B}$; i.e., a has infinitely many N-conjugates. Suppose to the contrary that $|a^N| = m$ for some finite m. Let $\ell = m!$, and note that a centralizes N^ℓ. Therefore, $\mathrm{Env}(N) \neq \mathrm{Env}(N^\ell)$, contradicting our choice of N.

Now we need to show that $C_{M_N(F)}(\mathscr{B}) = \mathrm{Env}(N)$. Let $\mathscr{C} = C_{M_n(F)}(\mathscr{B})$. Since $\mathrm{Env}(N) \subseteq \mathscr{B}$, $\mathscr{C} \subseteq C_{M_n(F)}(\mathrm{Env}(N)) = \mathscr{B}$. Therefore it suffices to show that $\mathrm{Env}(N) = Z(\mathscr{B})$. Let $E_1, ..., E_r$ denote the minimal ideals of $\mathrm{Env}(N)$. Let $V_i = E_i F^n$. Then $V_1 + \cdots + V_r$ is a direct sum decomposition of F^n into N-invariant subspaces. These subspaces are $\mathscr{B}$-invariant as well, but even more can be said. Each V_i may be viewed as a vector space over the field E_i. For any $a \in \mathscr{B}$, the action of a on V_i is linear over E_i, since a centralizes E_i. Conversely, suppose $a \in M_n(F)$ leaves the V_i invariant and acts as E_i-linear transformations on each V_i. Then a centralizes N and so belongs to $\mathscr{B}$. So $Z(\mathscr{B})$ commutes with all E_i-linear transformations on V_i. Since the center of a full matrix algebra is the set of scalar matrices, we have $Z(\mathscr{B}) = \mathrm{Env}(N)$ as desired. ▮

So far we have assumed that $\mathrm{Env}(G)$ is semisimple. We now replace that with the weaker assumption that N is nilpotent. As in the semisimple case, by Theorem 2.7, the minimals ideals of $\mathrm{Env}(N)/\mathrm{Rad}(\mathrm{Env}(N))$ yield a direct sum decomposition $V_1 + \cdots + V_r$ of F^n into N-invariant subspaces. The group G permutes the V_i. We consider the case that this action is transitive. Let G_1, denote the subgroup $\{g \in G \mid gV_1 = V_1\}$. Then G is a subgroup of the wreath product $G_1 \wr S_r$. We now consider the $r=1$ case (i.e., we focus our attention on G_1). In this case $\mathrm{Env}(N)/\mathrm{Rad}(\mathrm{Env}(N))$ is a field E of finite degree d over F. By [Su76, p. 228, Theorem 3], there is an embedding of E in $M_n(F)$ such that N acts as E-linear transformations (the embedding of E in $M_n(F)$ gives a natural way of interpreting F^n as an n/d-dimensional vector space over E). Indeed, the same reference shows that N acts as a subgroup of the direct product of the multiplicative group of E and a unipotent subgroup of $\mathrm{End}_E(F^n)$. Since N is normal in G, we have that G acts by conjugation as automorphisms of E which fix F. The kernel H of this action has index at most d in G. Note that H acts as E-linear transformations, so we may consider H to be a subgroup of $GL(n/d, E)$.

We summarize this in our main structure theorem for nilpotent-by-finite groups.

Theorem 3.2. *Suppose $G \leqslant GL(n, F)$ is nilpotent-by-finite. Then G has a nomal subgroup N of finite index such that:*

1. $\mathrm{Env}(N)$ *is a direct sum of ideals* $\mathscr{B}_1, ..., \mathscr{B}_r$, *such that each* $\mathscr{B}_i/\mathrm{Rad}(\mathscr{B}_i)$ *is a field* E_i.

2. F^n *is a direct sum of N-invariant subspaces* $V_1, ..., V_r$ *where* $V_i = \mathscr{B}_i F^n$. *These are permuted by G.*

3. E_i *can be embedded in* $\mathrm{End}_F(V_i)$ *such that N acts as* E_i*-linear transformations on* V_i.

4. *For any* $1 \leqslant i \leqslant r$, *the subgroup* $G_i = \{g \in G \mid gV_i = V_i\}$ *has a subgroup* H_i *of index dividing the degree of* E_i *over F such that* H_i *acts as* E_i*-linear transformations on* V_i.

The subgroup N is a subgroup of the direct product of a unipotent normal subgroup U of G and the multiplicative groups of the fields E_i.

Later, we will give a polynomial time algorithm to find the V_i. This will reduce many problems for nilpotent-by-finite G to the case $r = d = 1$ (so $G = H$, and N is isomorphic to a subgroup of a direct product of F^* and a unipotent group).

4. MONTE CARLO SOLVABILITY TESTING

We will see that the results of the previous section, although nonconstructive, are useful in the analysis of a natural Monte Carlo solvability test. This algorithm was suggested to the author by E. M. Luks and has not previously been analysed. For simplicity we consider only the case of rational matrix groups.

4.1. Straight Line Programs over Finitely Generated Groups

Let G be a finitely generated group, generated by the finite set S. A *straight line program*, or SLP, of length r over S is a sequence $g_1, ..., g_r$ such that for all $1 \leqslant i \leqslant r$, either $g_i \in S$ or $g_i = xy^{\ell}$, where $\ell \in \{-1, 1\}$ and $x, y \in S \cup \{g_1, ..., g_{i-1}\}$. We say that g_r is *reachable by an SLP of length r over S*. Also, if $X \subseteq \{g_1, ..., g_r\}$, we say that X is reachable by an SLP of length r over S. Note that if X is reachable by an SLP of length r_1 over S, and Y is reachable by an SLP of length r_2 over $X \cup S$, then Y is reachable by an SLP of length $r_1 + r_2$ over S. Observe that for $1 \leqslant i \leqslant r$, the element g_i is expressible as a word of length at most 2^{i-1} over the elements of S and their inverses. This will be used to calculate upper bounds on the bit lengths of elements reached by straight line programs.

Lemma 4.1. *Let* $\mathscr{A}$ *be an algebra of dimension d over a field F, and let S be a finite subset of* $\mathscr{A}^*$ *which generates* $\mathscr{A}$ *as an F-algebra. Then there is a straight line program* $g_1, ..., g_r$ *of length* $r = d$ *over S such that* $\{g_1, ..., g_r\}$ *is a basis for* $\mathscr{A}$ *over F. Such a straight line program can be found in polynomial time.*

Proof. Take an arbitrary $g_1 \in S$. For $i \geqslant 1$ suppose that $g_1, ..., g_i$ have been specified, and let V_i denote $\mathrm{Span}(\{g_1, ..., g_i\})$. It suffices to show that if $V_i \neq \mathscr{A}$, then we may choose g_{i+1} so that $V_i \neq V_{i+1}$. This is clear if V_i is a proper subspace of Span(S), so suppose $\mathrm{Span(S)} \subseteq V_i$. Then $\mathscr{A}$ is the smallest subspace containing V_i and closed under multiplication, so for some $x, y \in \{g_1, ..., g_i\}$, the element

$g_{i+1} = xy \notin V_i$, as desired. The elements x, y can clearly be selected in polynomial time. ▮

LEMMA 4.2. *Let $G = (S)$ for some finite set S and let $K = \langle T^G \rangle$ for some finite set $T \subseteq G$. Suppose that $\phi: K \to H$ is a homomorphism with $|H|$ finite. Then there is a straight line program $g_1, ..., g_r$ of length $r \leqslant 2 \log_2 |H|$ such that $\phi(K) = \langle \phi(\{g_1, ..., g_r\} \cap K) \rangle$.*

Proof. For simplicity assume that $H = \phi(K)$. Note that G acts as automorphisms on H and that H is generated by the G-images of the elements of $\phi(T)$. Let $A = \langle \phi(T) \rangle \leqslant H$, and let $t = |T|$. Assume without loss of generality that for any proper subset X of T, $\langle \phi(X) \rangle \neq A$. Then $t = |T| \leqslant \log_2 |A|$.

We are now ready to specify the straight line program. Let $g_1, ..., g_t$ be the elements of T. We proceed by induction. Suppose $g_1, ..., g_i$ have been specified. Let $A_i = \langle \phi(\{g_1, ..., g_i\} \cap K \rangle$. We assume by induction that $|A_i| \geqslant 2^{i/2}$ (the base case $i = r$ holds since $t \leqslant \log_2 |A|$). Either $A = H$ and we are done, or A is not closed under the action of G, in which case there is some $x \in \{g_1, ..., g_i\} \cap K$ and some $y \in §$ such that $\phi(yxy^{-1}) \notin A_i$. We set $g_{i+1} = xy^{-1}$ and $g_{i+2} = yg_{i+1} = yxy^{-1} \in K$. Therefore, $|A_{i+2}| \geqslant 2\,|A_i|$, and the induction step is complete. Eventually, preimages of a generating set for H are reached, using a straight line program of length $\leqslant 2 \log_2 |H|$ as desired. We note also that the "black-box" normal closure algorithm of Babai, Cooperman Finkelstein, Luks, and Seress [BCFLS] will produce, in Monte Carlo polynomial time, a straight-line program of length $O(\log |H|)$ with the desired property. ▮

An almost identical proof yields

LEMMA 4.3. *Let $G = \langle S \rangle$ for some finite set S and let $K = \langle T^G \rangle$ for some finite set $T \subseteq G$. Suppose that $\phi: K \to H$ is a homomorphism with $|H|$ infinite. Then for any bound b, there is a straight line program $g_1, ..., g_r$ of length $r \leqslant 2 \log_2 b$ such that $|\langle \phi(\{g_1, ..., g_r\} \cap K) \rangle| \geqslant b$.*

4.2. *The Algorithm*

Let $G \leqslant GL(n, \mathbf{Q})$ be given by a finite generating set S. We give a Monte Carlo polynomial time algorithm to determine whether or not G is solvable. The algorithm makes only one-sided errors; if the algorithm reports that G is not solvable, then it does so with certainty.

The algorithm is quite simple (see Fig. 1). Choose a random prime number p (not dividing the denominators of the entries of the elements of S or their inverses) with bit length b, and look at the homomorphic image K of G obtained by reducing modulo p. Solvability of K may be tested in deterministic polynomial time by Luks's algorithm [Lu92]. Suppose G is solvable. Then K will be found to be solvable, and the algorithm will output "G is (with high probability) solvable." Suppose now that G is not solvable. We will show for a suitable polynomially bounded choice of b that with high probability K will not be solvable. If K is found not to be solvable, then the algorithm outputs "G is (certainly) not solvable."

```
IsSolvable($G$)
$G = \langle S \rangle \leq GL(n, \mathbf{Q})$;
{
    $D = 1$;
    for $A \in S$ do
        $D = D \cdot$ numerator(det($A$));
        for $i, j \in \{1..n\}$ do
            $D = D \cdot$ denominator($A_{ij}$);
        od;
    od;
    $p =$ a random $b$-bit prime not dividing $D$;
    $R = \{\}$;
    for $A \in S$ do
    $R = R \cup \{A \bmod p\}$;
    od;
    if $K = \langle R \rangle \leq GL(n, p)$ is solvable then
        output "$G$ is probably solvable";
    else
        output "$G$ is certainly not solvable";
    fi;
}
```

FIG. 1. Pseudocode for the Monte Carlo solvability test.

Suppose that $G = \langle S \rangle$ is not solvable. We wish to show that the algorithm has only a very small probability of outputting "solvable." First we show that the set of primes p for which this is occurs is finite. A theorem of Zassenhaus [Za] states that there exists a number ℓ, depending only on n, such that all solvable matrix groups in dimension n *over any field* have derived length at most ℓ. Since G is not solvable, there is an element $g \in G^{(\ell)} \backslash 1$. The group K is solvable only if g is congruent to the identity matrix modulo p, which only happens for a finite set of primes p.

In order to estimate the error probability, we need to fine-tune this analysis. By Huppert [Hu], we may take $\ell = O(\log n)$. The most precise estimate, $\ell \leqslant 5 \log_9(n-2) + O(1)$, is due to Newman (cf. [We, Theorem 3. 10]). We will show that there exists an element $g \in G^{(\ell)} \backslash 1$ which is expressible as a straight line program over S with polynomially bounded length. First, we prove

Lemma 4.4. *Let $G, H \leqslant GL(n, \mathscr{C})$ with $G = \langle S \rangle$ and let $H = \langle T^G \rangle$ for some finite set $T \subseteq G$. Suppose H is not solvable. Then there exists a subset $T' \subseteq H$ expressible as a straight line program over $S \cup T$ of length $O(n^4)$ such that $\langle T' \rangle$ is not solvable.*

Proof. It suffices to consider the case that $\operatorname{Env}(G)$ is semisimple. We will treat T' as a variable that ranges over subsets of H. Initialize T' to be T. Let L denote the subgroup generated by T'.

Suppose first that $\operatorname{Env}(L)$ is not invariant under conjugation by G. By Lemma 4.1, we may add some elements of L to T' so that T' spans $\operatorname{Env}(L)$, using a straight line program of length $O(1)$ per increase in the dimension of $\operatorname{Span}(T')$. Then there is an $s \in S$, $t \in T'$, such that $t^s \notin \operatorname{Env}(L)$. We add t^s to T' (it is calculated by a straight line program of length 2) and repeat the process until we arrive at an L

such that $\mathrm{Env}(L)$ is invariant under conjugation by G. In other words, $\mathrm{Env}(L) = \mathrm{Env}(H)$. The total straight line program cost of this is $O(n^2)$.

Let $\theta\colon GL(n, \mathscr{C}) \to GL(M_n(\mathscr{C}))$ denote the conjugation action of $GL(n, \mathscr{C})$ on $M_n(\mathscr{C})$. By applying the reasoning of the previous paragraph to $\theta(L)$, we see that at a straight line program of cost $O(n^4)$, we arrive at an L such that $\mathrm{Env}(\theta(L)) = \mathrm{Env}(\theta(H))$.

Now consider the irreducible constituents $\theta_1, ..., \theta_r$ of $\theta(H)$. Let d_i be the degree of θ_i, so $\sum \theta_i = n^2$. For each $1 \leqslant i \leqslant r$, we want to have either $\theta_i(L) = \theta_i(H)$ or $\theta_i(L)$ is infinite. Suppose that $\theta_i(H)$ is finite. Then, by Lemma 4.2 and using Newman's bound [New] of $(2d_i)!$ on the order of finite subgroups of $GL(d_i, \mathbf{Q})$, a straight line program of length $O(d_i \log d_i)$ will reach a $T' \subseteq H$ such that $\theta_i(L) = \theta_i(H)$. Now suppose that $\theta_i(H)$ is infinite. Then, by Lemma 4.3 there is a straight line program of length $O(d_i \log d_i)$ which reaches a $T' \subseteq H$ such that the resulting L violates Newman's bound and is, therefore, infinite. The total cost of performing this step for all i is $O(n^2 \log n)$.

Now we claim that L is not solvable. Suppose to the contrary that L is solvable. Since $\mathrm{Env}(L) = \mathrm{Env}(H)$ is semisimple by Theorem 2.12, L has an abelian normal subgroup N of finite index. The algebra $\mathscr{B} = C_{M_n(\mathbf{Q})}(\mathrm{Env}(N))$ is $\theta(L)$-invariant and therefore, $\theta(H)$-invariant (since $\mathrm{Env}(\theta(L)) = \mathrm{Env}(\theta(H))$. Let $\theta_{\mathscr{B}}$ denote the conjugation action of H on $\mathscr{B}$. Since $\theta_{\mathscr{B}}(L)$ is finite, we have that $\theta_{\mathscr{B}}(H) = \theta_{\mathscr{B}}(L)$. By Lemma 3.1, $\mathrm{Env}(N)$ is the full centralizer in $M_n(\mathbf{Q})$ of $\mathscr{B}$, so the kernel in H of $\theta_{\mathscr{B}}$ is $H \cap \mathrm{Env}(N)$, which is abelian. However, $\theta_{\mathscr{B}}(H) = \theta_{\mathscr{B}}(L)$ is solvable, so H is solvable, a contradiction. ▮

Induction on ℓ gives, as an immediate consequence,

Lemma 4.5. *Suppose the finitely generated group $G \leqslant GL(n, \mathbf{Q})$ is not solvable. Then for any positive integer ℓ, there is a nonidentity element g of $G^{(\ell)}$ reachable by a straight line program of length $O(\ell n^4)$ over the generators of G.*

Theorem 4.6. *Let $G \leqslant GL(n, F)$ be given by a generating set S, where F is a symbolically given algebraic number field. Then in Monte Carlo polynomial time, it can be decided whether or not G is solvable.*

Proof. By treating F as an algebra over $\mathbf{Q}$, we may restrict our attention to rational matrix groups. We analyse the algorithm given above. Clearly, if G is solvable then the algorithm always gives the correct answer, so suppose that G is not solvable. Let $\ell = O(\log n)$ be an upper bound on the derived lengths of solvable matrix groups of degree n. By the above lemma, there is a nonidentity element g of $G^{(\ell)}$ reachable by a straight line program of length $O(\ell n^4)$ over S, from which it follows that g is expressible as a word of length $2^{O(\ell n^4)}$ over the elements of S and their inverses.

The algorithm will give the correct answer in the case that g is not congruent to the identity matrix modulo p, so consider the entries of $g - I$. These are rational numbers, not all zero, the numerators of which have bit lengths bounded by $2^{O(\ell n^4)}$ times a polynomial in the input length. Therefore, we may set b to be $O(\ell n^4 \log \mathrm{size}(S))$, and we will achieve that with high probability a randomly selected b-bit prime p will yield a correct computation.

We note that the element g and the straight line program reaching it are never actually calculated. ∎

5. ALGORITHM FOR THE TITS ALTERNATIVE

We are now ready to prove Theorems 1.1 and 1.3. By using the Friedl–Rónyai algorithm [FR], it suffices to consider the case that $\mathrm{Env}(G)$ is semisimple (otherwise, let η be the homomorphism described after Theorem 2.6, and replace G by $\eta(G)$). In other words, we are trying to determine if a completely reducible group is abelian-by-finite.

The algorithm that we present here will attempt to find the structure described in Section 3. If at any point this fails, then we know that the group G is not solvable-by-finite. We give the pseudocode for this algorithm in Fig. 2.

Let $\theta: G \to GL(M_n(F))$ denote the conjugation action of G on $M_n(F)$ (we are viewing $M_n(F)$ as a vector space). We compute $\mathscr{A} = \mathrm{Env}(\theta(G))$, and verify that $\mathscr{A}$ is semisimple (if not, then G is not solvable-by-finite). If $\mathscr{A}$ is semisimple, then we find its minimal ideals $\mathscr{A}_1, ..., \mathscr{A}_s$. Let $W_i = \mathscr{A}_i M_n(F)$, and let θ_i be the conjugation action of G on W_i. Using the finiteness algorithm of Babai, Beals, and Rockmore [BBR], we determine the set of i such that $\theta_i(G)$ is a finite group. Let $\mathscr{B}$ be the sum of the W_i such that $\theta_i(G)$ is finite. Note that $\mathscr{B}$ is a subalgebra of $M_n(F)$ (the

```
IsSolvableByFinite($G$)
$G = \langle S \rangle \leq GL(n, F)$;
{
    $\zeta$ = DiagonalBlockHomomorphism($G$);
    $G = \zeta(G)$;
    $\theta$ = conjugation of $G$ on $M_n(F)$;
    $\mathcal{A} = \mathrm{Env}(\theta(G))$;
    if not IsSemisimple($\mathcal{A}$) then
        output "$G$ is not solvable-by-finite";
        quit;
    fi;
    $\mathcal{A}_1, \ldots, \mathcal{A}_s$ = MinimalIdeals($\mathcal{A}$);
    $\mathcal{B} = 0$;
    for $i \in \{1, \ldots, s\}$ do
        $W_i = \mathcal{A}_i M_n(F)$;
        $\theta_i$ = conjugation of $G$ on $W_i$;
        if IsFinite($\theta_i(G)$) then
            $\mathcal{B} = \mathcal{B} + W_i$;
        fi;
    od;
    $\mathcal{C} = Z(\mathcal{B})$;
    if $\mathcal{C} = C_{M_n(F)}(\mathcal{B})$ then
        output "$G$ is solvable-by-finite";
    else
        output "$G$ is not solvable-by-finite";
    fi;
}
```

FIG. 2. Pseudocode for the solvable-by-finite test.

W_i are merely subspaces). By Lemma 3.1, if G is solvable-by-finite then $\mathscr{B}$ is the centralizer in $M_n(F)$ of $\operatorname{Env}(N)$, for some abelian normal subgroup N of finite index in G. If this is the case, we verify it as follows.

By solving linear equations, we find the center $\mathscr{C}$ of $\mathscr{B}$, and verify that $\mathscr{C}$ is the centralizer in $M_n(F)$ of $\mathscr{B}$. This is the case if G is solvable-by-finite (cf. Lemma 3.1). Conversely, suppose that $\mathscr{C}$ is the full centralizer in $M_n(F)$ of $\mathscr{B}$. G acts by conjugation as a finite group on $\mathscr{B}$, so $G \cap C_{M_n(F)}(\mathscr{B})$ has finite index in G. But $C_{M_n(F)}(\mathscr{B}) = \mathscr{C}$ is commutative, so G is abelian-by-finite. This completes the proof of Theorem 1.1. ▮

To prove Theorem 1.3, we let ϕ be the conjugation action of G on $\mathscr{B}$. ▮

6. THE NILPOTENT-BY-FINITE CASE

We now give the proofs of Theorems 1.5 and 1.7. A general divide-and-conquer strategy of the algorithm is to reduce computational problems to "simpler" groups using direct products and semidirect products. For example, suppose that a direct sum $F^n = V_1 + V_2$ of G-invariant subspaces is found. Let G_i denote the action of G on V_i. Then G is a subgroup of $G_1 \times G_2$, and we may work with the G_i separately. The supergroup $\mathscr{G}$ that we define will be the direct product $\mathscr{G}_1 \times \mathscr{G}_2$.

We begin by applying the algorithm of the previous Section to $\eta(G)$. Let $f_1, ..., f_r$ be the central idempotents corresponding to the decomposition of $\mathscr{C}$. We wish to "lift" these central idempotents of $\operatorname{Env}(N)/\operatorname{Rad}(\operatorname{Env}(N))$ to central idempotents of $\operatorname{Env}(N)$. Let $G_i \leqslant G$ consist of all elements g such that $\eta(g)$ fixes V_i. Schreier generators (cf. [Si1, FHL]) for G_i can be found in polynomial time. Note that η extends naturally to an algebra homomorphism from $\operatorname{Env}(G)$ to $\operatorname{Env}(\eta(G))$, which we will also denote η. To lift f_i, find an $e_i \in Z(\operatorname{Env}(G_i))$ such that $\eta(e_i) = f_i$ (this reduces to solving a system of linear equations). By Lemma 2.9, we may assume that e_i is diagonalizable. By Corollary 2.8, this will succeed unless the group is not nilpotent-by-finite.

Assume that G is nilpotent-by-finite and that $e_1, ..., e_r$ have been lifted. Let $V_i = e_i F^n$. The V_i are permuted by G, and G is easily represented as a subdirect product of subgroups of wreath products. Standard techniques [Si1, FHL], together with Propositions 2.4 and 2.5, now allow us to reduce our computational problems to the case $r = 1$.

We assume now that $r = 1$, so $\mathscr{C}$ is a field E of degree d over F for some d dividing n. The action of G on $\mathscr{C}$ corresponds to F-automorphisms of $\mathscr{C}$ (i.e., automorphisms which fix F). Therefore the centralizer in G of $\mathscr{C}$ has index at most d. Again, techniques of [Si1, FHL] allow us to reduce to the case that G acts trivially on $\mathscr{C}$.

We now assume that G acts trivially on $\mathscr{C}$, and we wish to embed $E = \mathscr{C}$ in the center of $\operatorname{Env}(G)$. Such an embedding exists iff G is nilpotent-by-finite. By Corollary 2.10, we can find such an embedding or show that none exists in polynomial time.

Once the field E has been embedded in the center of $\operatorname{Env}(G)$, we may effectively view G as a subgroup of $GL(n/d, E)$. Thus, we have reduced to the case that $E = F$.

We are now in the case that $G \leqslant GL(n, F)$ has a normal subgroup N of finite index such that N is a subdirect product of F^* and a unipotent group U. By applying the Friedl–Rónyai algorithm to $\operatorname{Env}(G)$, we are able to factor elements of G as products of the form $g = \eta(g)\,\mu(g)$, where $\eta(G)$ acts by conjugation as a finite group on $M_n(F)$ and the group $\bar{U}$ generated by the $\eta(G)$-conjugates of $\mu(G)$ is unipotent (this group may be bigger than U). Let θ denote the conjugation action of $\eta(G)$ on $M_n(F)$, and let π denote a homomorphism of $\eta(G)$ to $GL(kn, p)$ obtained by converting $\eta(G)$ to a rational matrix group of degree kn (k is the degree of F) and reducing the entries modulo p for some odd prime p. By [New, Theorem IX.7], π will act faithfully on any subgroup H of G such that $\eta(H)$ is finite.

We claim that G is isomorphic to the semidirect product $(\mu(G) \rtimes \theta(G)) \times \pi(G) \times \det(G)$. It is sufficient to show that for $g \in \ker(\pi) \cap \ker(\theta) \cap \ker(\det)$, $\eta(g) = 1$. If $\theta(g) = 1$, then $\eta(g)$ is a scalar matrix λI for some $\lambda \in F$. For such a g, if $\det(g) = 1$, then $\lambda^n = 1$, so π acts faithfully on $\langle g \rangle$. If, in addition $\pi(g) = 1$, then $\eta(g) = 1$ as desired.

To prove Theorem 1.7, by Proposition 2.5 it suffices to give slow-growth encodings separately for (supergroups of) $\mu(G)$, $\theta(G)$, $\pi(G)$, and $\det(G)$. For the first three, the standard encoding suffices (for $\theta(G)$ this is Theorem 2.1, for $\mu(G)$ this is Lemma 2.3). For $\det(G) \leqslant F^*$, we use Ge's multiplicative encoding [Ge93a]. ▌

To prove Theorem 1.5, let $\phi(g) = (\theta(g), \pi(g), \det(g))$. The group $\theta(G)$ is managed by [BB], and the group $\det(G)$ is handled with [Ge93a]. The group $\pi(\ker(\theta) \cap \ker(\det))$ has order $\leqslant n$, and poses no problem. The [Ge93a, BB] algorithms give a polynomial length straight line program to compute normal generators for $\ker(\phi)$. By using the slow growth encoding of Theorem 1.7, such a straight line program can be evaluated in polynomial time. ▌

REFERENCES

[Atk75] M. D. Atkinson, An algorithm for finding the blocks of a permutation group, *Math. Comp.* **29** (1975), 911–913.

[Atk84] M. D. Atkinson (Ed.), "Computational Group Theory," Proceedings of the London Mathematical Society Symposium on Computational Group Theory, Academic Press, London, 1984.

[Ba79] L. Babai, Monte-Carlo algorithms in graph isomorphism testing, unpublished manuscript, 1979.

[Ba85] L. Babai, Trading group theory for randomness, *in* "Proc 17th ACM STOC, Providence, RI, 1985," pp. 421–429.

[Ba91a] L. Babai, Computational complexity in finite groups, *in* "Proc. International Congress of Mathematics, Kyoto, 1990," pp. 1479–1489, Springer-Verlag, Hong Kong, 1991.

[Ba92] L. Babai, Deciding finiteness of matrix groups in Las Vegas polynomial time, *in* "Proc. 3rd ACM-SIAM SODA (Symp. on Discrete Algorithmss), 1992," pp. 33–40.

[BBCIL] L. Babai, R. Beals, J.-y. Cai, G. Ivanyos, and E. M. Luks, Multiplicative equations over commuting matrices, *in* "Proc. 7th ACM-SIAM SODA, 1996," pp. 498–507.

[BBR] L. Babai, R. Beals, and D. Rockmore, Deciding finiteness of matrix groups in deterministic polynomial time, *Israel J. Math.*, to appear.

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

[BLS87] L. Babai, E. M. Luks, and Á. Seress, Permutation groups in *NC*, *in* "Proc. 19th ACM STOC, 1987," pp. 409–420.

[BLS88] L. Babai, E. Luks, and Á. Seress, Fast management of permutation groups, *in* "Proc. 28th IEEE FOCS, 1988," pp. 272–282 .

[BSz] L. Babai, and E. Szemerédi, On the complexity of matrix group problems, *in* "Proc. 25th IEEE FOCS, 1984," pp. 229–240.

[BB] R. Beals and L. Babai, Las Vegas algorithms for matrix groups, *in* "Proc. 34th IEEE FOCS, 1993," pp. 427–436.

[BS] R. Beals and Á. Seress, Structure forest and composition factors in nearly linear time for small base groups, *in* "Proc. 24th ACM STOC, 1992," pp. 116–125.

[BCY] G. Brassard, C. Crépeau, and M. Yung, Everything in *NP* can be argued in *perfect* zero-knowledge in a *bounded* number of rounds, *in* "Proc. of the 16th ICALP, 1989," pp. 123–136, Springer-Verlag, Berlin.

[BY] G. Brassard, and M. Yung, One-Way Group Actions, *in* "Advances in Cryptology: CRYPTO '90 proceedings, 1991," Lecture Notes in Computer Science, Vol. 537, pp. 94–107, Springer-Verlag, Berlin.

[Bu] G. Butler, "Fundamental Algorithms for Permutation Groups," Lecture Notes in Computer Science, Vol. 559, Springer-Verlag, Berlin, 1991.

[CLZ] J.-y. Cai, R. J. Lipton, and Y. Zalcstein, The complexity of the membership problem for 2-generated commutative semigroups of rational matrices, *in* "Proc. 35th IEEE FOCS (1994)," pp. 135–142.

[Can] J. J. Cannon, An Introduction to the Group Theory Language, Cayley, Computational Group Theory, *in* "Proceedings of the London Mathematical Society Symposium on Computational Group Theory" (M. D. Atkinson, Ed.), pp. 145–184, Academic Press, London, 1984.

[FHL] M. L. Furst, J. Hopcroft, and E. M. Luks, Polynomial-time algorithms for permutation groups, *in* "21st IEEE FOCS, 1980," pp. 36–41.

[FR] K. Friedl and L. Rónyai, Polynomial time solutions of some problems in abstract algebra, *in* "Proc. 17th ACM STOC (1985)," pp. 153–162.

[Ga] F. R. Gantmacher, "Matrix Theory," Chelsea Publishing, New York, 1959.

[Ge93a] G. Ge, "Algorithms Related to Multiplicative Representations of Algebraic Numbers," Ph.D. thesis, Math. Dept., U.C. Berkeley, 1993.

[Ge93b] G. Ge, Testing equalities of multiplicative representations in polynomial time, *in* "Proc. 34th IEEE FOCS (1993)," pp. 422–426.

[HR] D. F. Holt and S. Rees, Testing modules for irreducibility, *J. Austral. Math. Soc. Ser. A* **57** (1994).

[H+96a] D. F. Holt, C. R. Leedham-Green, E. A. O'Brien, and S. Rees, Computing matrix group decompositions with respect to a normal subgroup, *J. Algebra* **184** (1996), 818.

[H+96b] D. F. Holt, C. R. Leedham-Green, E. A. O'Brien, and S. Rees, Testing matrix groups for primitivity, *J. Algebra* **184** (1996), 795.

[Hu] B. Huppert, Linear auflösbare Gruppen, *Math. Z.* **67** (1957), 479–518.

[IY] R. Impagliazzo and M. Yung, Direct minimum-knowledge computations, *in* "Advances in Cryptology: CRYPTO '87 Proceedings," Lecture Notes in Computer Science, Vol. 293, pp. 40–51, Springer-Verlag, Berlin, 1988.

[KLi] R. Kannan and R. Lipton, Polynomial-time algorithms for the orbit problem, *J. Assoc. Comput. Mach.* **33**, No. 4 (1986), 808–821.

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

[LSZ] R. Lipton, L. Snyder, and Y. Zalcstein, Complexity of the word and isomorphism problems for finite groups, *in* "Proc. Conf. Info. Sciences and Systems, 1976," John Hopkins University, pp. 33–35.

[LZ] R. Lipton and Y. Zalcstein, Word problems solvable in logspace, *J. Assoc. Comput. Mach.* **24** (1977), 522–526.

[Lu82] E. M. Luks, Isomorphism of graphs of bounded valence can be tested in polynomial time, *J. Comput. System Sci.* **25** (1982), 42–65.

[Lu87] E. M. Luks, Computing the composition factors of a permutation group in polynomial time, *Combinatorica* **7** (1987), 87–99.

[Lu92] E. M. Luks, Computing in solvable groups, *in* "Proc. 33rd IEEE FOCS, 1992," pp. 111–120.

[Lu93] E. M. Luks, Permutation groups and polynomial-time computation, *in* "Groups and Computation," pp. 139–175, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 11, Amer. Math. Soc., Providence, RI, 1993.

[M] S. Magliveras, A crypotosystem from logarithmic signatures of finite groups, *in* "Proc. 29th Midwest Symposium on Circuits and Systems (1986)," pp. 972–975.

[MM89] S. Magliveras and N. Memon, Properties of Cryptosystem PGM, *in* "Advances in Cryptology—CRYPTO '89," Lecture Notes in Computer Science, Vol. 435, Springer-Verlag, Berlin, 1989, 447–460.

[MM92] S. Magliveras and N. Memon, Algebraic properties of cryptosystem PGM, *J. Cryptology* **5** (1992), 167–183.

[Mih] K. A. Mihaĭlova, The occurrence problem for a direct product of groups, *Dokl. Akad. Nauk* **119** (1958), 1103–1105. [in Russian]

[Mil] J. Milnor, Growth of finitely generated solvable groups, *J. Differential Geometry* **2** (1968), 447–449.

[NP] P. M. Neumann and Cheryl E. Praeger, A recognition algorithm for the special linear groups, *Proc. London Math. Soc.* **65** (1992), 555–603.

[New] M. Newman, "Integral Matrices," Pure and Applied Mathematics, Vol. 45, Academic Press, New York, 1972.

[Os] G. Ostheimer, Ph.D. thesis, Mathematics Dept., Rutgers University, 1996.

[Sch+] M. Schönert *et al.*, "GAP—Groups, Algorithms, and Programming," Lehrstuhl D für Mathematik, Rheinisch Westfälische Technische Hochschule, Aachen, Germany, 1992.

[Schr] A. Schrijver, "Theory of Linear and Integer Programming," Wiley, New York, 1986.

[Si1] C. C. Sims, Computation with permutation groups, *in* "Proc. Second Symposium on Symbolic and Algebraic Manipulation" (S. R. Petrick, Ed.), pp. 23–28, ACM, New York, 1971.

[Si2] C. C. Sims, "Computation with finitely presented groups," Cambridge Univerity Press, New York, 1994.

[Su63] D. Suprunenko, "Soluble and Nilpotent Linear Graphs," Transl. Math. Monographs, Vol. 9, Amer. Math. Soc., Rhode Island, 1963.

[Su76] D. Suprunenko, "Matrix Groups," Transl. of Math. Monographs, Vol. 45, Amer. Math. Soc., Rhode Island, 1976.

[Ti] J. Tits, Free subgroups in linear groups, *J. Algebra* **20** (1972), 250–270.

[We] B. A. F. Wehrfritz, "Infinite Linear Groups," Ergeb. Math. Grenzgeb., Vol. 76, Springer-Verlag, Berlin, 1973.

[Wo] J. A. Wolf, Growth of finitely generated solvable groups and curvature of Riemannian manifolds, *J. Differential Geometry* **2** (1968), 421–446.

[Za] H. Zassenhaus, Beweis eines Satzes über diskrete Gruppen, *Abh. Math. Sem. Univ. Hamburg* **12** (1938), 289–312.