CIS-TR 85-09
Fast Parallel Computation with Permutation Groups

Eugene M. Luks

Department of Computer and Information Science
University of Oregon

Pierre McKenzie

Département d'I.R.O.
Université de Montréal

CIS-TR 85-09
Fast Parallel Computation with Permutation Groups

Eugene M. Luks

Department of Computer and Information Science
University of Oregon

Pierre McKenzie

Département d'I.R.O.
Université de Montréal

Fast Parallel Computation with Permutation Groups[1]

Eugene M. Luks[2]

Department of Computer and Information Science
University of Oregon

Pierre McKenzie[3]

Département d'I.R.O.
Université de Montréal

Abstract. We develop fast parallel solutions to a number of basic problems involving solvable and nilpotent permutation groups. Testing solvability is in *NC*, and *RNC* includes, for solvable groups, finding order, testing membership, finding the derived series and finding a composition series. Additionally, for nilpotent groups, one can, in *RNC*, find the center, a central composition series, and point-wise stabilizers of sets. There are applications to graph isomorphism. In fact, we exhibit a class of vertex-colored graphs for which determining isomorphism is *NC*-equivalent to computing ranks of matrices over small fields. A useful tool is the observation that the problem of finding the smallest subspace containing a given set of vectors and closed under a given set of linear transformations (all over a small field) belongs to *RNC*.

1. Introduction and statement of results

We present several new fast parallel algorithms for dealing with permutation groups, along with implications for graph isomorphism. Until now little was known about the parallel complexity of non-Abelian permutation group problems (see [Mc84], [Re85]). Techniques developed for Abelian groups [McCo85], combining elementary number theory with properties of transitive Abelian groups, do not generalize. Among other things, we now place in *NC* or *RNC* fundamental questions about solvable and nilpotent groups that were not long known to be in *P*. Applications put graph isomorphism instances into *RNC*. The randomness (*R* in *RNC*) arises from the need for linear algebra over tiny fields F_p (where "tiny" means that the value of p is bounded by the length of the problem encoding). In fact, we show that linear algebra (e.g., finding ranks) over tiny fields is *NC*-equivalent to a subcase of graph isomorphism.

---

1) To appear in *Proceedings 26th Symposium on Foundations of Computer Science*, 1985.
2) Research supported by NSF Grant MCS-8403745.
3) Research supported by the Université de Montréal, by the Programme de Formation de Chercheurs et d'Action Concertée du Québec, and by the Natural Sciences and Engineering Research Council of Canada.

An essential tool for dealing with permutation groups that are (succinctly) represented only by generators (we assume all groups are so specified) is a membership test. The permutation group membership problem (GM) consists of determining whether a given permutation belongs to the generated group. Furst, Hopcroft and Luks showed that a variant of Sims' algorithm [Si70] for GM could be implemented in polynomial time [FuHoLu80a], but the only subcase known to be in *RNC* was that of Abelian groups [McCo83]. In [Mc84] the latter subcase was shown *NC*-equivalent to computing the rank of a matrix over a tiny field F_p.

Theorem 1.1. GM restricted to solvable permutation groups belongs to the complexity class *RNC*.

Theorem 1.1 yields, as special cases, *RNC* solutions to GM restricted to nilpotent groups, and thus to p-groups (positively answering a question from [McCo85]). We also show

Theorem 1.2. Computing the order of a solvable permutation group belongs to *RNC*.

The following result forms an important tool in several subsequent algorithms.

Theorem 1.3. Computing the normal closure $NCL_G(H)$ of a subgroup H of a solvable permutation group G belongs to *RNC*.

The normal closure $NCL_G(S)$ of a set S in a containing group G is the smallest subgroup containing S and normal in G. For example, this enables us to get at the structural underpinnings of the group:

Corollary 1.4. Computing the derived series and a composition series of a solvable permutation group belongs to *RNC*.

Note, though the derived series has poly-log length, a composition series need not.

Now consider the pointwise set stabilizer problem (POINTSET). Given a permutation group G and a subset of the points on which G acts, POINTSET consists of computing the largest subgroup of G fixing each point in the subset. (By contrast, the set stabilizer problem SET would permit mapping points in the subset to other points in the subset.) Theorem 1.3 is also instrumental in the proof of

Theorem 1.5. POINTSET for nilpotent groups belongs to *RNC*.

Pointwise set stabilizers play an early and important role in the development of fast sequential algorithms [FuHoLu80]. Though they arrive here at a much later stage (and, in fact, are not yet available for general groups), they are still of great value. For example, they are used in

Theorem 1.6. Let G be a permutation group in a class of groups X and let H be an *arbitrary* permutation group such that G normalizes H (i.e. H is normal in the group generated by G and H). Then computing the centralizer $C_G(H)$ *NC*-reduces to solving POINTSET for the class X.

Corollary 1.7. Let H be an *arbitrary* permutation group normalized by a nilpotent permutation group G. Computing $C_G(H)$ belongs to *RNC*.

The centralizer $C_G(H)$ consists of the elements of G that commute with all of H. A special case of theorem 1.6 involves finding centralizers of normal subgroups (when $H<G$) and, if $H=G$, we get another important structural result:

Corollary 1.8. The center of a nilpotent permutation group can be computed in *RNC*.

Even more structural information on a nilpotent group is attainable:

Theorem 1.9. A central series for a nilpotent group can be computed in *RNC*.

Proofs of the theorems stated thus far exploit the fact that much of linear algebra over tiny fields F_p can be performed in *RNC* ([BoGaHo82], [Ga84]). We establish the following tool:

Theorem 1.10. *RNC* contains the following problem: Given a subset A of F_p^d (p tiny), and a set T of linear transformations of F_p^d (described by matrices), find (a basis for) the smallest subspace that contains A and is closed under the action of T.

Given the persistence of $(R)NC$ in the above it is satisfying to know that

Theorem 1.11. Testing a permutation group for solvability belongs to *NC*.

Theorem 1.11 is, in fact, a consequence of the next theorem. Define a property as "hereditary" if whenever the property holds for a group G it holds for any subgroup and any quotient group of G, and whenever the property holds for both a normal subgroup N of G and for G/N it holds for G. Examples of hereditary properties include solvability, being a p-group, and having bounded non-Abelian composition factors.

Theorem 1.12. Testing a permutation group for a hereditary property *NC*-reduces to testing that property for a primitive permutation group.

Since it is known that a bound on the non-Abelian composition factors imposes a polynomial-bound on the size of a primitive permutation group [BaCaPa82], testing for small non-Abelian composition factors is also in *NC*. We observe that this class of groups arises in testing isomorphism of graphs of bounded valence [Lu82].

Define **N** to be the class of vertex-colored graphs for which the automorphism group within each color class is contained in a nilpotent, small (i.e. polynomial order) group that is computable in *NC*.

Examples. The automorphism group within each color class is nilpotent if, for example,

-it is a directed cycle (cyclic group)
-it is a connected trivalent graph with a distinguished edge (2-group)
-it is a p-ary tree with a cyclic orientation imposed on the children of each node (p-group).

Let m be the size of the color class. In the first example the group is transparent and of order m. In the second, it is computable in sequential time $O(m^3)$ and has size $2^{m/2}$ [GHLSW82], so we could allow m to be as large as $O(\log n)$. In the third, the conditions are satisfied with $m = O(\log_p n)$. Note that it is not essential that every color class be restricted. The containing group might be influenced by interconnections with other classes.

Theorem 1.13. If a graph is in N then generators for its automorphism group can be computed in *RNC*.

Instances of computability of automorphism groups typically facilitate isomorphism testing [Lu82]; for example, if one can compute the automorphism group of the disjoint union of two connected graphs then isomorphism is tested by seeing whether an automorphism switches the connected components (note, in the above examples, that the disjoint union of connected trivalent graphs with distinguished edges has a 2-group for automorphism group). Here, too, our automorphism group technique has isomorphism-testing implications, though, to broaden the applicability, we establish this connection in another way (remark 3.5). We can test isomorphism in *RNC* when the color classes are as described above. As we have indicated the *R* in *RNC* stems back to our exploitation of linear algebra in working with solvable and nilpotent groups. Thus, the following result is striking

Theorem 1.14. Computing the rank of a matrix over a tiny field is *NC*-equivalent to determining isomorphism in a certain *NC*-recognizable class of graphs.

This reduction is particularly interesting in light of the fact that the question of whether linear algebra over tiny fields belongs to *NC* (as opposed to *RNC*) remains open.

We conclude with comments on open problems.

2. Background and notation

We assume familiarity with the complexity classes *NC* [Pi79], *RNC* [Co83], and *P* (see [HoUl79]) generalized to include more than just decision problems (see [Co85]). Informally, we say that problem A *NC*-reduces to problem B if $B \in NC$ implies $A \in NC$ and $B \in RNC$ implies $A \in RNC$ (see [Co83] for a more precise formulation). A is *NC*-equivalent to B if A *NC*-reduces to B and B *NC*-reduces to A. We refer the reader to [Ha59] for definitions and basic facts about solvable groups, nilpotent groups, commutators, commutator subgroups, central series, derived series, and composition series. Our notation is mostly that of [Wi64].

We write $H \leq G$ when H is a subgroup of a group G. With S a set of group elements, $<S>$ represents the group generated by S. Let g,h belong to a group G. The commutator $[g,h]$ is defined as the element $g^{-1}h^{-1}gh$ and we write g^h for the conjugate of g by h, that is, $h^{-1}gh$. Group G "acts" on a set Ω if there is a homomorphism $\phi: G \longrightarrow \mathrm{Sym}(\Omega)$. In such a case we write α^g for the image of $\alpha \in \Omega$ under $g \in G$, and Γ^g for the set of images of elements of $\Gamma \subseteq \Omega$ under g. G_Γ is the setwise stabilizer of Γ, that is, the group of all elements in G which map Γ to itself. $\Gamma \subseteq \Omega$ is a G-orbit if, for each $\alpha,\beta \in \Gamma$, $\alpha^g = \beta$ for some $g \in G$. G acts transitively on $\Gamma \subseteq \Omega$ if Γ is a G-orbit. A G-block is a set $\Gamma \subseteq \Omega$ such that either $\Gamma^g \cap \Gamma = \emptyset$ or $\Gamma^g = \Gamma$ holds for each $g \in G$. If Γ is a G-block then G acts naturally on the G-block system $\{\Gamma^g | g \in G\}$. If G acted transitively on Ω, then a G-block system partitions Ω into G-blocks of equal size. We say that G acts primitively on Ω if Ω cannot be broken up into nontrivial (i.e. sizes $\neq 1$ or $|\Omega|$) G-blocks.

For our purposes, an algebra is a vector space equipped with an associative multiplication that distributes over linear combinations. For example, a set of matrices over F_p closed under matrix addition and under matrix multiplication is an algebra.

3. Proofs

Consider theorems 1.1 and 1.2 , denoting by G the given group. Sequential algorithms for GM proceed by first constructing a linear length tower of subgroups of G fixing progressively more points [FuHoLu80a]. The set of "strong generators" computed by these algorithms is the union of complete sets of coset representatives for each successive quotient space in this tower. Not only does this procedure not seem to parallelize, but even if a set of strong generators were given as input, one could only "sift" the test permutation through the underlying tower, one notch at a time, resulting in a linear time parallel solution at best.

Our solvable GM algorithm proceeds instead by constructing a subgroup tower of length $(\log n)^2$, where n is the size of the point set on which G acts, through which it is possible to "sift" group elements in *RNC*. For nilpotent groups, the tower of the last paragraph will be computable in *RNC* also, but only following the development of our pointwise set stabilizer algorithm below.

Definition (structure forest). A structure forest for a permutation group G is a forest on which G acts as automorphisms (fixing the roots of the individual trees), whose leaves form the permutation domain, and such that the stabilizer within G of any node ($\equiv$ set of subtended leaves) acts trivially or primitively on the children of this node. (When G is solvable, the Pálfy-Wolf bound on the order of primitive solvable groups ([Pa82], [Wo82]) ensures that the stabilizer of a node restricted to the children of this node has order polynomial in the number of children.)

Typical sequential methods for constructing such a forest (requiring at a "primitive" node, the subgroup fixing the blocks) lead either to "blow-ups" in the sizes of generating sets, or to sequential "sifts" through linear-height towers of groups. As had occurred independently to Reif [Re85], we can avoid these pitfalls, so that

Proposition 3.1. NC^3 contains the problem of computing a structure forest for an *arbitrary* permutation group G.

Proof. First we break up the point set into orbits [McCo83]. (Each orbit gives rise to a tree in the forest and we build each tree in parallel.) Now if G acts transitively on Ω and if $\Delta \subseteq \Gamma \subseteq \Omega$ is a G_Γ-block, then Δ is in fact a G-block. This suggests picking a nontrivial G-block of smallest size, say Γ (in NC^2 [Si67], [Mc84], [Re85]). The previous statement guarantees that not only G but also G_Γ then acts (transitively and) primitively on Γ. Hence Γ can be made a set of leaves with common parent. Images of Γ under G yield the other subtrees at the bottom level. The procedure is repeated with the parents so created (in effect with the G-action on the G-block system containing Γ). After $\log n$ iterations (hence NC^3) each tree is complete. □

Definition (power-commutator basis of a group). A power-commutator basis (PCB) of a group G is an ordered sequence $(b_1,\rho_1), \ldots, (b_m,\rho_m)$, $b_i \in G$, $\rho_i > 1$ an integer, $1 \leq i \leq m$,

such that

1) each $g \in G$ is uniquely expressible in the canonical form $b_1^{\epsilon_1} \cdots b_m^{\epsilon_m}$, $0 \leq \epsilon_i < \rho_i$, $0 \leq i \leq m$,
2) for each pair of integers i,j, $1 \leq i < j \leq m$, the canonical expression for the commutator $[b_j,b_i]$ satisfies $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_i = 0$, and
3) for each integer i, $1 \leq i \leq m$, the canonical expression for the element $b_i^{\rho_i}$ also satisfies $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_i = 0$.

Definition (PCB of a group relative to a normal subgroup). Given K a normal subgroup of a group G, a power-commutator basis for G relative to K (PCB of G *rel* K) is an ordered sequence $(b_1,\rho_1), \ldots, (b_m,\rho_m)$, $b_i \in G$, $\rho_i > 1$ an integer, $1 \leq i \leq m$, such that $\{(b_iK,\rho_i)\}_{1 \leq i \leq m}$ is a PCB for G/K. With respect to this PCB "sifting an element $g \in G$" means "computing the unique $h \in K$ such that the product gh^{-1} is expressible in the form $b_1^{\epsilon_1} \cdots b_m^{\epsilon_m}$, $0 \leq \epsilon_i < \rho_i$, $0 \leq i \leq m$." If the PCB is understood, we denote the induced function $G \longrightarrow K$ by *SIFT*; i.e., in the previous sentence, $h = SIFT(g)$.

Observe that if $\psi : G \longrightarrow H$ is a group epimorphism with $(b_i,\rho_i)_{1 \leq i \leq m}$ a PCB for H, then $(\psi^{-1}(b_i),\rho_i)_{1 \leq i \leq m}$ is a PCB for G *rel* $\mathrm{Ker}\psi$. Note also that if $(b_i,\rho_i)_{1 \leq i \leq m}$ is a PCB for a group, then for $i = 1, \ldots, m-1$, $\langle b_{i+1}, \ldots, b_m \rangle$ is a normal subgroup of $\langle b_i, \ldots, b_m \rangle$.

Computing PCBs is crucial to most of our algorithms, and a PCB for a group G exists if and only if G is a solvable group. Much of the usefulness of power-commutator bases stems from the following two propositions.

Proposition 3.2. Let K be a normal subgroup of G, and let $\{(b_i,\rho_i)\}_{1 \leq i \leq m}$ be a PCB for G *rel* K. Denote by S the set of images under *SIFT* of generators for G, of commutators $[b_j,b_i]$, $1 \leq i < j \leq m$, and of powers $b_i^{\rho_i}$, $1 \leq i \leq m$. Then $K = NCL_G(S)$.

Proof. That $NCL_G(S) \subseteq K$ follows by normality of K. So let $k \in K$. Since generators for G were sifted, k can be written as a product of PCB elements and of elements of S. Migrating occurrences of b_1 to the left (given that $b_1^{-1}sb_1 \in NCL_G(S)$ whenever $s \in NCL_G(S)$ and that $b_1^{-1}b_jb_1$ can be expressed without b_1 for $j > 1$) and reducing the resulting exponent of b_1 modulo ρ_1 (reexpressing $b_1^{\rho_1}$ as required), then repeating for $b_2, b_3, \cdots$, we can express k as

$$b_1^{\epsilon_1} b_2^{\epsilon_2} \cdots b_m^{\epsilon_m} \sigma \ , \ 0 \leq \epsilon_i < \rho_i \ , \ 1 \leq i \leq m,$$

with $\sigma \in NCL_G(S) \subseteq K$. It follows that $b_1^{\epsilon_1} \cdots b_m^{\epsilon_m} \in K$ and hence that $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_m = 0$ (by the uniqueness criterion in the definition of PCB for G *rel* K). Therefore $k \in NCL_G(S)$. Hence $K \subseteq NCL_G(S)$. □

Proposition 3.3. Let $K_1 \leq K_2 \leq G$ with K_1 and K_2 each normal in G. Then a PCB for G *rel* K_1 is obtained by appending a PCB for K_2 *rel* K_1 to a PCB for G *rel* K_2.

Proof. To show that $[b_j,b_i]$ is expressible appropriately when b_j belongs to the PCB for G *rel* K_2 and b_i to that for K_2 *rel* K_1, we appeal to normality of K_1. Other properties are straightforward to verify. □

Theorems 1.1 and 1.2 *(Proofs).* We compute a PCB, for the input (solvable) group G, for which it is possible to compute *SIFT* in *RNC*. Sifting then answers the membership question, and the order of G is $\rho_1\rho_2\cdots\rho_m$.

Let F be the structure forest for G as computed per proposition 3.1 . Consider level i, $0\leq i\leq \log n$, as the level of all the nodes at distance i from a root in F (n is the degree of G). Denote the action of G on nodes at level $\leq i$ by ϕ_i. Note that the kernel of this action, $\mathrm{Ker}\phi_i$, fixes all nodes at height $\leq i$ and that $G/\mathrm{Ker}\phi_i$ may be viewed as the induced action on the forest obtained by pruning all trees to height i. These kernels form a $\log n$ height group tower of normal subgroups of G and we proceed, inductively, finding PCBs for the quotients $G/\mathrm{Ker}\phi_i$ (employing proposition 3.3). So suppose inductively that we have a PCB for G *rel* $\mathrm{Ker}\phi_k$, and that with respect to this PCB we can compute *SIFT* in *RNC*. We write K for $\mathrm{Ker}\phi_k$, and S for a known set (computed by sifting, proposition 3.2) for which $K = NCL_G(S)$. It suffices to show how to extend our PCB to a PCB for G *rel* $\mathrm{Ker}\phi_{k+1}$.

We first treat the case of nilpotent G (think p-group). At each level k node we compute generators for the subgroup of G stablizing that node; this may be done using Schreier's technique (see [Ha59]), the number of generators growing by a factor equal to the index of the subgroup in $G \leq$ the number of level k nodes. These subgroups act primitively, thus as cyclic groups of prime order, on the children of the corresponding node. We form the "vector space" L by taking the direct product of these primitive actions acting on all the children, more precisely a product of vector spaces of different characteristics. (There are other ways to obtain L; we follow this one with a view toward the generalization to the general solvable case.) Now, K acts on the level k+1 nodes as a subgroup of L. Writing ψ for this K-action, we claim that we can compute a PCB for K *rel* $\mathrm{Ker}\psi$ in *RNC*. To see this note that G acts by conjugation (hence as homomorphism, i.e., linear transformations) on the "vector space" L (homomorphisms must preserve parts of different characteristics). Recalling that $K = NCL_G(S)$, $\mathrm{Im}\psi$ is then the smallest subspace containing $\psi(S)$ and closed under the linear transformations induced by generators of G. We can therefore obtain a basis (hence a PCB) for the subspace $\mathrm{Im}\psi$ by theorem 1.10 and so, having kept track of inverse images throughout, a PCB for K *rel* $\mathrm{Ker}\psi$. This proves our claim. Now observing that $\mathrm{Ker}\psi$=$\mathrm{Ker}\phi_{k+1}$, we appeal to proposition 3.3 and produce the desired PCB for G *rel* $\mathrm{Ker}\phi_{k+1}$. We point out that one can sift through this PCB by sifting, in succession, through the two PCB's that form it; hence, the process remains in *RNC*.

The rest of the proof is devoted to the general solvable case. Here the step from K to $\mathrm{Ker}\phi_{k+1}$ is still too large ($K/\mathrm{Ker}\phi_{k+1}$ is not a vector space), so we need to refine this step of the normal series, inserting $O(\log n)$ groups whose successive factor groups can be viewed as vector spaces. We will be able to compute PCBs for successive quotients in sequence, and to use proposition 3.3 to paste these PCBs into a PCB for K *rel* $\mathrm{Ker}\phi_{k+1}$.

To describe the first stage in the refinement process, note that K on the level k+1 nodes acts as a subgroup of a direct product of primitive solvable groups. We compute, as in the nilpotent case, these primitive solvable groups, each acting on a set of children

of a level k node. Denoting by L the direct product of the small groups thus computed (hence L contains the image of the action of K on all level $k+1$ nodes), we compute a characteristic subgroup A of L such that L/A is a product of elementary Abelian groups (indeed there exists a prime p for which $T/<T^p,[T,T]>$ is nontrivial elementary Abelian whenever T is a nontrivial solvable group; but since we wish to preserve the conjugation action of G on L/A, the same "nontrivial" p must be chosen for each level k node in a G-orbit). Note that A, too, is a direct product of groups, one for each level k node (in fact, that's how we find it).

We claim that we can compute a PCB for the "vector space" $\mathrm{Im}\psi$, where

$$\psi : K \longrightarrow L/A,$$

which will yield a PCB for K *rel* $\mathrm{Ker}\psi$. For this we observe that G acts by conjugation on L, inducing actions on A and L/A. In other words conjugation by an element of G induces a linear transformation within the "vector space" L/A, and recalling that $K=NCL_G(S)$ we can use theorem 1.10 as before. (Computing the vectors and matrices assumed by theorem 1.10 requires a cyclic decomposition of L/A and the ability to express an arbitrary group element in terms of the basis; [McCo85] describes parallel algorithms for such problems.) This proves the claim, and we can further obtain a PCB for G *rel* $\mathrm{Ker}\psi$ by proposition 3.3 . Again, sifting through this PCB remains in *RNC*.

This completes the description of the first refinement stage. At this point we have gone from a PCB for G *rel* K to a PCB for G *rel* the subgroup $\mathrm{Ker}\psi$ of K. The stage is then repeated, with K replaced by $\mathrm{Ker}\psi$ and L replaced by A ($\mathrm{Ker}\psi \leq A$), until L becomes trivial (this occurs within $O(\log n)$ stages since at each stage the order of each non-trivial component of L is at least halved). At that point, $\mathrm{Ker}\psi$ fixes all level $k+1$ nodes; Hence $\mathrm{Ker}\psi = \mathrm{Ker}\phi_{k+1}$ and we have a PCB for G *rel* $\mathrm{Ker}\phi_{k+1}$. □

Theorem 1.3 *(Proof).* Writing $\mathbf{N}$ for $NCL_G(H)$, we compute a PCB for $\mathbf{N}$ by duplicating the strategy described in the proof of theorem 1.1 . What changes is the specification of the elements whose images under *SIFT*, given a PCB for $\mathbf{N}$ *rel* $(K \cap \mathbf{N})$, form a set S for which $K \cap \mathbf{N} = NCL_G(S)$ (K is the kernel which "shrinks" from G to 1 in $O(\log^2 n)$ stages). The proof of proposition 3.2 extends provided we now sift: the commutators and powers (as before) of PCB elements, and, in lieu of generators for G, the generators for H as well as each PCB element conjugated by each generator of G. The spanned "vector space" is closed up, as before under the action of G. □

Corollary 1.4 *(Proof).* The length of the derived series is $O(\log^2 n)$, and successive commutator subgroups are obtained from theorem 1.3 using the fact that $[G,G] = NCL_G\{[g,h] \mid g,h \in S\}$ whenever S is a generating set for G. Now a composition series for G is formed by the subgroups $<\{b_j\}_{j \leq i}>$ for $1 \leq i \leq m$ in the PCB computed in the proof of theorem 1.1 , since the ρ_is are prime integers. □

Theorem 1.5 *(Proof).* This proof bears a superficial resemblance to that of theorem 1.1 . Initially we mark nodes, in the structure forest F for G, which subtend leaves to be fixed. From generators for the group G_k fixing the marked nodes at level k (available inductively), we compute generators for G_{k+1}, again by looking at the induced

action on the level k,k+1 trees extracted from F (but this time only those with marked roots, noting that G_k fixes these roots and that an unmarked root cannot have marked descendants). Iterating this process eventually yields generators for the group fixing exactly the marked leaves.

To describe how to compute generators for G_{k+1} from generators for G_k, write K for G_k and L for the image of K on the aforementioned trees (as in the nilpotent case of theorem 1.1 L is a direct product of vector spaces). We compute (componentwise and in *NC*) the largest subgroup H of L that fixes the marked level k+1 nodes (so that H includes the image of G_{k+1}). Then we consider the quotient space L/H (this assumes a single characteristic for L; extending to the case of a product of spaces of different characteristics is no problem) and we compute a basis for $\mathrm{Im}\psi$, where

$$\psi : K \longrightarrow L/H,$$

(in *RNC*, using the rank algorithm as in [McCo85]) in order to deduce a PCB for K *rel* $\mathrm{Ker}\psi$. But $\mathrm{Ker}\psi$ is precisely the set of elements in K which fix the marked level k+1 nodes, that is, $\mathrm{Ker}\psi = G_{k+1}$. So a generating set for G_{k+1} is obtained from the PCB for K *rel* $\mathrm{Ker}\psi$ by applying proposition 3.2 and theorem 1.3 . □

Theorem 1.8 *(Proof).* The technique is a parallelized version of an algorithm in [Lu85]. Write C for $C_G(H)$ and Ω for the relevant point set. We form, for each generator h of H, the set

$$\Gamma_h = \{(x,x^h) \mid x \in \Omega\} \subseteq \Omega \times \Omega.$$

Observing that $g \in G$ commutes with a generator h of H iff g (on $\Omega \times \Omega$) stabilizes Γ_h, imagine $\Omega \times \Omega$ colored (in *NC*) in such a way that two points share a color iff they belong to exactly the same sets Γ_h. We claim that by refining the color partition until each color class becomes a G-block (working on each G-orbit in parallel and successively seeking any nontrivial intersection of color classes with an image, under generators of G, of the smallest color class and using the intersections to refine colorings), we maintain the property that $g \in G$ preserves each color class iff $g \in C$. To conclude from this refinement we consider the action of G on the set made up of Ω together with the colored classes as additional points and we obtain C as the pointwise set stabilizer of the additional points.

To prove our claim, note that G normalizing H implies that C is normal in G. Hence if C preserves color class Γ then, for any $g \in G$, $\Gamma^{gC} = \Gamma^{g(g^{-1}Cg)} = \Gamma^g$ is preserved by C also. In other words we lose no element of C if we insist on preserving not only each original color class Γ but Γ^g for each $g \in G$ as well. That is, we lose no element of C if we refine until each class is a G-block. □

Theorem 1.10 *(Proof).* First we obtain a basis σ for the matrix algebra with identity, τ, generated by T, and second we compute a basis for $B = Span(\sigma A) \subseteq F_p^d$. B is a subspace of F_p^d containing A and closed under T because B is closed under $Span(\sigma)$ and the latter includes the identity transformation as well as each matrix in T. B is the smallest such subspace because linear closure under a set of linear transformations implies closure under the span of this set.

The first step is performed in stages. Write τ_i for the subspace of of τ spanned by all products of i matrices in T (with the identity transformation thrown into T). Stage j computes a basis for the subspace τ_{2^j} by forming the product of each pair of basis matrices for $\tau_{2^{j-1}}$, and then by computing in RNC a basis for the new span using the techniques in [BoGaHo82]. $2\log d$ stages suffice because the dimension of τ is at most d^2, $\tau_i \subseteq \tau_{i+1}$ for each i, and $\tau_i = \tau$ whenever $\tau_i = \tau_{i+1}$. The second step is in RNC by [BoGaHo82]. □

We postponed the demonstration of theorem 1.9 so as to capitalize on some ideas in the last proof.

Theorem 1.9 *(Proof).* Suppose $G = \langle S \rangle$. We may assume that G is a p-group, for general nilpotent G can be factored as a direct product of p-groups [Mc84] and it is an easy matter to reassemble central series of the factors into one for G. We have seen in the proof of Theorem 1.1 that a normal series

$$G = G^{(0)} \geq G^{(1)} \geq \cdots \geq G^{(s)} = 1$$

can be constructed in which each quotient $G^{(k)}/G^{(k+1)}$, denoted below by V, is an elementary abelian p-group (vector space over F_p). Furthermore, we have a convenient representation domain in which to work with V in the $k,k+1$ slice of the structure forest. We need to show that, for each k, we can insert G-normal subgroups

$$G^{(k)} = H^{(0)} \geq H^{(1)} \geq \cdots \geq H^{(m)} = G^{(k+1)}$$

so that $[G,H^{(j)}] \leq H^{(j+1)}$. Equivalently, we need to insert G-invariant subspaces

$$V = V^{(0)} \geq V^{(1)} \geq \cdots \geq V^{(m)} = 0$$

so that, for $g \in G$, $v \in V^{(j)}$, $v - v^g$ is in $V^{(j+1)}$ (note that we switch to the additive notation in V in viewing $[g,v]$). Each g in G induces a linear transformation t_g of V where $t_g(v) = v - v^g$. Let $T = \{t_s \mid s \in S\}$. We compute, as in the proof of Theorem 1.10 , τ_i, the linear span of the set of all products of i elements from T. This time, however, we need τ_i for all $i = 1,2, \ldots, dim(V)$. Clearly, these can be computed in parallel once we have determined the τ_{2^j} (note that τ_{a+b} is spanned by the products of basis elements of τ_a and τ_b). We claim that we may take $V^{(j)} = \tau_j(V)$. It is immediate that $V^{(j+1)} = SpanT(V^{(j)})$, so that $V^{(j)} \geq V^{(j+1)}$ follows inductively from $V \geq V^{(1)}$. To see that $V^{(j)}$ is G-invariant, it suffices to note that it is invariant under S, but, for $s \in S$, $v \in V^{(j)}$,

$$v^s = v - t_s(v) \in V^{(j)} + V^{(j+1)} \leq V^{(j)}.$$

This equation also gives the congruence

$$v^g \equiv v \bmod (V^{(j+1)})$$

for all g in a generating set, and so the congruence holds for all $g \in G$, whence $v - v^g \in V^{(j+1)}$. Finally, we need to show that, if $m = dim(V)$, $V^{(m)} = 0$. For this, recall that the nilpotency of G implies that there is an M so that, for all $h,h_1,h_2, \ldots, h_M \in G$,

$$[h_M \cdots [h_2,[h_1,h]]\ldots] = 1.$$

But this implies that $\tau_M = 0$. Knowing, then, that the sequence $V^{(0)} \geq V^{(1)} \geq V^{(2)} \ldots$

will reach 0 eventually, we must simply conclude that this happens within m steps. For this, observe that once equality $V^{(0)}=V^{(i+1)}$ happens, then the sequence is stable (by induction) thereafter. But the sequence of dimensions $m=dim\,V^{(0)}, dim\,V^{(1)}, dim\,V^{(2)}, \ldots$ can strictly decrease at most m times. □

We remark that the proof may be extended to produce a central composition series by inserting, if necessary, arbitrary intermediate spaces so that dimensions go down by 1 in each step.

Theorems 1.11 and 1.12 *(Proofs).* Theorem 1.11 follows from theorem 1.12 once we observe that solvability is a hereditary property and that the Pálfy-Wolf bound reduces testing solvability of a primitive solvable group to testing solvability of a group having order polynomial in its degree (hence in *NC* by brute force).

Theorem 1.12 generalizes a technique used in [Mc84] to test nilpotency in *NC*. Each transitive constituent of the group is tested for the hereditary property P in parallel. We compute a structure tree for the transitive group G (proposition 3.1). With S the set of children of the root, P holds for G iff P holds for both the G-action on S and for the stabilizer within G of any one node in S restricted to the leaves subtended. To see this note that the (restricted) stabilizers of each node in S are images of conjugate subgroups in G whose direct product contains the kernel of the G-action. Now G acts on S, and from a generating set of size r for G we can compute by Schreier's method (see [Ha59]) a generating set of size $r|S|$ for the stabilizer of some node $\alpha\in S$. Applying this argument recursively to the stabilizer acting on the children of α, eventually we reach the bottom of the structure tree with a generating set bounded in size by r times the number of leaves in the tree (i.e. r times the degree of G). □

Remark 3.4. The basic divide-and-conquer technique used in the last proof has other applications. Suppose we consider the question of whether a prime p divides the order of G, a possibly easier problem than computing the order. This time we observe that property holds for G iff it holds for at least one transitive constituent, and it holds for a transitive group iff it holds either for the (primitive) group acting on a set of maximal blocks or for the subgroup fixing one block, in its action within that block. Consider, for example, groups with bounded non-abelian composition factors. It is known [BaCaPa82] that primitive groups in this class have polynomially bounded order. Thus testing whether p divides the order is in *NC*. Note, though, that we do not know how to find the order.

Theorem 1.13 *(Proof).* We compute generators for $Aut(X)$, X in **N**, by reduction to POINTSET for nilpotent groups, as follows. Write G for the direct product of the automorphism groups in the color classes of X. By definition of **N**, G restricted to a pair of color classes is a direct product K of two *NC*-computable polynomial-size nilpotent groups. Then by brute force and in *NC* we can compute the subgroup H of K mapping edges to edges, and we can describe the (right) action of K as permutations of cosets of H. Computing $Aut(X)$ then is an instance of POINTSET for nilpotent groups if we extend the action of G to the union of all such cosets of all groups H (one group H and corresponding K for each pair of color classes) and if we take the trivial cosets H as the points to be fixed. We conclude by theorem 1.5 . □

Remark 3.5. New graph automorphism algorithms often lead to new graph isomorphism tests. A typical procedure is to apply the automorphism group construction to the disjoint union of connected graphs to be tested, then to check whether some element switches the components. Thus, theorem 1.13 applies if the union of the graphs belong to N. Unfortunately, that limits the applicability since the disjoint union, for example, of two directed cycles does not have a nilpotent automorphism group (except if the cycles have size 2^c). It is possible to get around this difficulty in a way that retains isomorphism testing for the graph examples to which we have been able to apply theorem 1.13 (e.g., the illustrations given for N). We outline the idea here, promising full details at a later date. We imitate the algorithm for $Aut(X)$ in theorem 1.13 to compute $Iso(X,Y)$, the set of all isomorphisms from X to Y directly (an analogous approach is exploited in [GHLSW82]). We assume this time that

(1) we know the small nilpotent automorphism group in each color class of X.

(2) we know a single isomorphism in each color class of X to corresponding color class of Y.

Forming G as in the proof of theorem 1.13 , and gluing the isomorphisms together we form a set Gf that contains $Iso(X,Y)$. The set Gf is, strictly speaking, not a coset of G though it behaves like one and $Iso(X,Y)$, if not empty, is a subcoset (with corresponding subgroup $Aut(X)$). A reduction similar to that in theorem 1.13 leads this time to the POINTSET-TRANSPORTER problem (PST): we are given G acting on A, an injection $f:A \longrightarrow B$, and sequences $a_1, \ldots, a_m$ and $b_1, \ldots, b_m$ in A, B respectively; the problem is to determine if there are elements of Gf that map a_i to b_i for $i=1, \ldots, m$ and, if so, output the "subcoset" of these. We can solve PST for nilpotent G in a manner analogous to POINTSET. A forest over $f(A)$ is copied, via f, from the structure forest on A. In addition to markings in the forest over A, edges (a_i,b_i) are drawn and these lift to corresponding edges between parents, etc. (if this procedure does not produce a target for some parent or if two parents have a common target, the answer is "empty"). Again, we proceed down the tree, cutting the coset Gf to a pointset-transporter at successive levels. When we get to level k, G is the pointwise stabilizer of the marked nodes and we view its "vector space action" on the children of these nodes. We need now elements of Gf that map each marked child, c, to its partner, d. This is equivalent to finding the subcoset of G mapping each c to $f^{-1}(d)$. This again translates to a linear algebra problem, though not a homogeneous one. If not empty, the answer is a subcoset, Hg, of G and we lift the subspace interpretation of H back to G by the same sifting and normal closure procedure as before.

Theorem 1.14 *(Proof).* We show that the question of whether $v \in \mathrm{Span}(v_1, \ldots, v_m)$, given $v, v_1, \ldots, v_m \in F_p^d$, is *NC*-reducible to an instance of graph NONisomorphism. The resulting graphs lie in a class covered by remark 3.5 above and so isomorphism is testable in *RNC*. (Actually, we could get around using that remark by observing that the linear algebra problem is reconstructible in *NC* from the specific graphs).

First we build a graph Y for which $F_p^d = Aut(Y)$ (we are considering only the additive group of the space). For this take d disjoint directed p-cycles $A_0, \ldots, A_{d-1}$, each p-

cycle colored uniformly, d distinct colors. We claim (see next paragraph) that for any $w \in F_p^d$ it is possible to augment Y, without yet changing the automorphism group, so that it contains a colored p-cycle $C_w=(c_0, \ldots, c_{p-1})$, with the following property: any element $x \in F_p^d$ maps c_i to $c_{i+x\cdot w}$, (where $x\cdot w$ denotes dot product of vectors and all arithmetic is mod p). If C_w is then colored, but non uniformly, the automorphisms of the resulting graph are orthogonal to w. Including such augmentations for $v_1, \ldots, v_m$ produces a graph whose automorphism group is the orthogonal complement of $\mathrm{Span}(v_1, \ldots, v_m)$. Noting that the fundamental question can be restated "Is this orthogonal complement orthogonal to v?" we augment further to include a p-cycle reflecting the dot product with v. The question then is whether the points in this (as yet uncolored) p-cycle are fixed by all automorphisms. To turn this into a graph isomorphism question, we make two copies of the last graph. In each we now color the critical p-cycle C_v, p–1 points red, one point blue (red and blue have not been used before). But we make sure different points are individualized in the two copies. The resulting two graphs are NON-isomorphic iff the points of C_v are fixed by all graph automorphisms. It doesn't matter which two different points were colored blue, since if any automorphism acted non-trivially on C_v, some power of it would map one of these points to the other.

It remains to prove the claim concerned with augmenting Y to contain the p-cycle C_w, $w \in F_p^d$. In the following, new cycles added to Y are always colored with new colors. Let $A = (a_0, \ldots, a_{p-1})$ and $B = (b_0, \ldots, b_{p-1})$ be cycles and let $r,s \in \mathbf{Z}_p$ be nonzero. A basic building block in the construction of C_w is yet another new cycle U, called the "(r,s)-sum of A and B", having the property that an automorphism of the new graph that maps a_0 to a_i and b_0 to b_j necessarily maps u_0 to u_{ri+sj}: color a "square grid" $d_{i,j}$ $(0 \le i,j \le p-1)$, and join $d_{i,j}$ to each of a_i, b_j, c_{ri+sj} ; then U satisfies the (r,s)-sum property. Finally to construct C_w (assuming all entries of w nonzero and d a power of 2; simple modifications take care of other cases), we construct the (w_{2i}, w_{2i+1})-sum of A_{2i} and A_{2i+1}, $0 \le i < d/2$ in parallel, then the pairwise (1,1)-sums of the results, then the pairwise (1,1)-sums of those, and so on, until only one cycle remains. (Actually, we can construct all of these cycles at the same time). This one cycle is C_w. □

Remarks 1. The construction involves directed graphs. Standard procedures can be used to convert the isomorphism question to one for undirected graphs.
2. The augmentation with C_w can be done so that additional color classes have size 1, p, or p^2. In these, it is not necessary to include any internal (to the class) edges at all. However, to arrive at an easily recognizable subclass of N, we can add (superfluous but non-interfering) edges so that all non-trivial color classes have groups that are trivial or cyclic of order p or the direct product of two cyclic groups of order p.

4. Comments, questions, and regrets.

We delineate some of the frontiers between problems we have now established to be in *RNC* and others about which little is known.

The tractability of the problems of finding order (ORDER), pointwise set stabilizer (POINTSET), and set stabilizer (SET) are of particular interest. We have solved the

first for solvable groups and the second for nilpotent groups. (We are embarrassed to confess that we had claimed a solution to POINTSET for solvable groups that seemed to generalize Theorem 1.5 in the spirit of the nilpotent-to-solvable generalization in Theorem 1.1 ; there now seems evidence that no generalization along such lines can exist). The third problem, SET, has been solved in *RNC* for the still more restrictive class of abelian groups ([Mc84], [McCo85]). The battle lines are then clear, how about ORDER for general groups? or, for a next step beyond solvable, groups with bounded non-abelian composition factors [Lu82]? POINTSET for solvable groups? SET for nilpotent groups? We hesitate to recommend SET for general groups, since that problem is not known to be in *P*, indeed, if it were then graph isomorphism would be in *P* [Lu82]. On the other hand SET is in *P* even for solvable groups [Lu82]. By way of further motivation, we mention that the first author has shown that trivalent graph isomorphism *NC*-reduces to SET for 2-groups (the original reduction in [Lu82] is *not* an *NC* reduction). Even POINTSET for solvable groups would broaden the graph isomorphism applications noted herein; for example, since S_4 is solvable, testing isomorphism of vertex-colored graphs with ≤ 4 vertices in each color class would be in *RNC* (though, remarkably, the 5-vertex case would remain open).

Incidentally, in light of the narrowing instances of solutions to ORDER, POINTSET, SET, it is worth remarking that ORDER *NC*-reduces to POINTSET and POINTSET to SET. In any class of groups, ORDER is *NC*-reducible to POINTSET as follows: POINTSET enables one, in parallel, to produce the subgroups, G_i, $i=1, \ldots, n$, which fix the first i points of the set (in any predetermined order); but it is always possible to compute the index $[G_i, G_{i+1}]$, for that is the size of the orbit of the $(i+1)$st point under G_i; the product of these indices is the order of G. In any class of groups, POINTSET is *NC*-reducible to SET: to stabilize, pointwise, the subset $\{a_1, \ldots, a_m\}$ of A, look at the naturally induced action of G on $A \times A$ and find the set stabilizer of $\{(a_1, a_2), (a_2, a_3), \ldots, (a_{m-1}, a_m)\}$.

Beyond ORDER for general groups, one should ask whether the group's building blocks, the composition factors (cf. corollary 1.4), are obtainable in *NC* or *RNC*; it is known they are attainable in polynomial-time [Lu85]. Kantor [Ka85] has shown that Sylow p-subgroups can be found in polynomial time; can this be done in *RNC*? can it be done even for solvable groups in *RNC*? For general groups, can one even find an element of order p, given that p divides $|G|$. A possible easier question than ORDER for general groups might be that of testing, for a prime p, whether p does divide $|G|$ (see remark 3.4).

We remark, finally, on the very fundamental question of permutation group membership, GM. (It is well known, by the way, that GM reduces to ORDER: compare order of group with that obtained when given element is added to generators.) Is GM complete for *P* (as conjectured in [McCo83]) and so unlikely to belong to *NC* or *RNC*? On the one hand, our work suggests that a reduction showing completeness of GM would involve nonsolvable groups, and on the other hand, the following problem emerges as the first obstruction to further progress on GM. Let K be a fixed permutation group on a set A, and consider testing membership in a subgroup, given by generators, of

$K \times K \times \cdots \times K$, the direct product of n K's, which is acting naturally on a disjoint union of n A's. By our results, this critical problem is in RNC if $|K| \leq 59$ (whence solvable). But if K is a simple group of order 60, we have no idea how to proceed.

References

[Ba79]

Babai, L., *Monte Carlo Algorithms in Graph Isomorphism Testing* , Tech. Rep. 79-10, Dép. Math. et Stat., Univ. de Montréal, 1979.

[BaCaPa82]

Babai, L., Cameron, P.J., Pálfy, P., On the order of primitive groups with restricted nonabelian composition factors, *J. Algebra.* 79, 1982, pp 161-168.

[BoGaHo82]

Borodin, A., von zur Gathen, J., Hopcroft, J., Fast Parallel Matrix and GCD Computations, *Information and Control* 52 (1982), pp 241-256.

[Co83]

Cook, S.A., *The Classification of Problems which have Fast Parallel Algorithms* , Proceedings of the 1983 International FCT Conference, Lecture Notes in Computer Science 158, Springer-Verlag, pp 78-93.

[Co85]

Cook, S.A., *A Taxonomy of Problems with Fast Parallel Algorithms* , preprint, 1985.

[FuHoLu80a]

Furst, M., Hopcroft, J., Luks, E., Polynomial Time Algorithms for Permutation Groups, *Proc 21st IEEE FOCS* (1980) , pp 36-41.

[FuHoLu80b]

Furst, M., Hopcroft, J., Luks, E., *A subexponential algorithm for trivalent graph isomorphism* , Tech. Rep. 80-426, Dept. of Computer Science, Cornell Univ., 1980.

[Ga84]

von zur Gathen, J., *Lecture Notes for CSC2408S (Parallel Arithmetic Complexity)* , Dept. of C.S., Univ. of Toronto, Jan 1984.

[GHLSW82]

Galil, Z., Hoffmann, C. M., Luks, E. M., Schnorr, C. P., Weber, A., An $O(n^3 \log n)$ deterministic and an $O(n^3)$ probabilistic isomorphism test for trivalent graphs, *Proc 23rd IEEE FOCS* (1982) , pp 118-125.

[HoUl79]

Hopcroft, J.E., Ullman, J.D., *Introduction to Automata Theory, Languages, and Computation* , Addison-Wesley, 1979.

[Ka85]
Kantor, W.M., *Sylow p*-subgroups of permutation groups in polynomial time, submitted for publication, 1985.

[Lu82]
Luks, E.M., Isomorphism of Graphs of Bounded Valence Can Be Tested in Polynomial Time, *JCSS,* 25, 1982, pp. 42-65.

[Lu85]
Luks, E.M., *Computing composition factors of a permutation group in polynomial time,* Dept. of Computer and Information Sciences Tech. Rep. 85-07, Univ. of Oregon, 1985.

[Mc84]
McKenzie, P., *Parallel Complexity and Permutation Groups* , Doctoral Thesis, Dept. of Computer Science, Univ. of Toronto, 1984.

[McCo83]
McKenzie, P., Cook, S.A., The Parallel Complexity of the Abelian Permutation Group Membership Problem, *Proc. 24 th IEEE FOCS* , 1983, pp 154-161.

[McCo85]
McKenzie, P., Cook, S.A., The Parallel Complexity of Abelian Permutation Group Problems, submitted for publication, 1985.

[Pa82]
Pálfy, P., A Polynomial Bound on the Orders of Primitive Solvable Groups, *J. Algebra.* July 1982, pp 127-137.

[Pi79]
Pippenger, N., On simultaneous resource bounds, *Proc 20th IEEE FOCS* (1979) , pp 307-311.

[Re85]
Reif, J., *Probabilistic Algorithms in Group Theory* , TR-01-85 , Aiken Computation Laboratory, Harvard Univ. , November 1984.

[Si70] Sims, C.C., Computational methods in the study of permutation groups, in *Computational Problems in Abstract Algebra* , ed. J. Leech, Pergamon Press, 1970, pp 169-183.

[Wi64]
Wielandt, H., *Finite Permutation Groups* , Academic Press, 1964.

[Wo82]
Wolf, T.R., Solvable and nilpotent subgroups of GL(n,q^n), *Can. J. Math.* , v.34, 1982, pp 1097-1111.