in Proc. 27th IEEE FOCS, 1986

Parallel Algorithms for Permutation Groups and Graph Isomorphism

Eugene M. Luks*

Department of Computer and Information Science
University of Oregon

Abstract

We develop parallel techniques for dealing with permutation group problems. These are most effective on the class of groups with bounded non-abelian composition factors. For this class, we place in NC problems such as membership testing, finding the center and composition factors, and, of particular significance, finding pointwise-set-stabilizers. The last has applications to instances of graph-isomorphism and we show that NC contains isomorphism-testing for vertex-colored graphs with bounded color multiplicity, a problem not long known to be in polynomial time.

1 Introduction

The last few years have seen substantial progress in polynomial-time algorithms for instances of the graph isomorphism question (e.g, [Ba1], [FHL1], [Lu1], [Mi1], [Mi2], [BGM], [GHLSW], [BL], [FSS], [BKL]). A conceptual breakthrough was Babai's demonstration [Ba1] of a random polynomial-time algorithm for testing isomorphism in the class of vertex-colored graphs with color multiplicity less than a prescribed bound b (we refer to this class as CG_b). Note that, even with $b = 2$, the number of vertex-color-preserving maps between n-vertex graphs, could be $2^{\lfloor n/2 \rfloor}$, and no procedure was previously demonstrated to be better than the brute force consideration of these. Babai's work formed the major inspiration for the development of polynomial-time tools for permutation groups by Furst, Hopcroft and Luks [FHL1], that, for instance, dispensed with the randomness in the isomorphism test for CG_b. Further advances have been both characterized and facilitated by extensions of these algebraic techniques.

It seems that those algebraic methods have been inherently sequential in nature. So the question arises as to whether any of these problems, recently placed in polynomial time, are susceptible to a parallel approach. We now answer this affirmatively. In particular, we put in NC the problem whose novel polynomial-time solution stimulated the above activity, namely, testing isomorphism in CG_b. Of course, one expects the machinery for graph isomorphism to remain algebraic. Nevertheless, the depth of requisite algebra is surprising. The isomorphism tests of [Ba1], [FHL1] rested on elementary group properties (for example, Lagrange's Theorem). To bring the problem into NC, however, we make essential use of the internal structure of primitive permutation groups.

For abelian permutation groups, McKenzie and Cook [Mc], [MC] seem to have resolved the main parallel complexity issues, placing in NC such central problems as: membership testing, finding set stabilizers, and determining the cyclic factors. However, their techniques are not extendible to wider group classes, for they rely both on cyclic decomposability, which is unique to abelian groups, and on the regularity of transitive abelian groups (so that the size of the group is only that of the permutation domain).

Fast parallel methods that began to treat non-abelian groups were announced by Luks and McKenzie in [LM]. Therein several basic problems were solved for the class of solvable groups, including testing membership and finding a composition series. The property of solvable groups that was exploited in [LM] was the existence of a polylog-length sequence of normal subgroups such that the quotients are products of vector spaces. Using this, divide-and-conquer algorithms were devised that had their base cases rooted in linear algebra problems, all of which were in turn reducible to the standard problem of matrix rank. Since the vector-space-quotient property characterizes solvable groups, it appeared that, as in the earlier abelian deadlock, the methods were not extendible. For the more restrictive class of nilpotent groups, it was shown that NC contains the problems

*Research supported by NSF Grants DCR-8403745 and DCR-8609491.

of finding the center, and pointwise stabilisers of subsets. The latter are a particularly useful tool in applications, including instances of graph isomorphism. Unfortunately, the technique for capturing these, exploiting their situation in a polylog-length normal series, does not apply to non-nilpotent groups.

(Note that [Mc],[MC],[LM] claim, for the most part, *random* NC results but the randomness stems only from the need for matrix ranks over small fields and these are now available in NC [Mu]).

Our new isomorphism-testing results require a wide extension of the class of groups that can be manipulated and analyzed with fast parallel algorithms. This includes the class, Γ_b, of groups whose non-abelian composition factors are subgroups of the symmetric group, S_b. The class properly contains solvable groups (which have *no* non-abelian composition factors) but is more specifically motivated by its natural occurrence in graph-isomorphism settings ([Lu1], [Mi1], [Mi2]). We demonstrate NC algorithms for determining the underlying structure of these groups, including order, composition factors, center. We also answer fundamental questions relating to their actions as permutation groups, including membership-testing and finding any *pointwise-set-stabilizer* (the subgroup that fixes all points in a target subset). The latter is directly applicable to isomorphism-testing in CG_b and broader graph classes.

Critical to our handling of non-solvable groups is a machinery that plays the linear algebra role. In this "non-abelian linear algebra" we consider the groups that arise as a natural generalization of the vector spaces in the solvable case. Thus, vector spaces over small finite fields, which are, as groups, direct products of small abelian simple groups, generalize to direct products of small not-necessarily-abelian simple groups ("small" connotes polynomial size). The analogues of the vector space *algorithms* are less obvious. While Gaussian elimination generalizes easily, the tools for *parallel* computation (notably, determinants) have no clear counterpart. Instead, we use properties unique to *non-abelian* simple groups to develop a suitable *non-abelian linear algebra* in NC. Basic problems including membership-testing and finding composition factors, for groups in Γ_b and wider classes, are reducible to abelian and non-abelian linear algebra questions. For the pointwise-set-stabiliser result, however, the divide-and-conquer of the generalised vector spaces does not capture the structure of the problem. We can reduce only to the case where the group acts primitively on each orbit. At that point, we prove the existence and NC-constructibility of a generalised-vector-space subgroup that acts independently and transitively in a large number of orbits. This subgroup can be used to stabilize, in parallel, the targeted points in those orbits.

Recall that Γ_b is, essentially, the broadest class of groups for which there is known to be a polynomial-time solution to the *set stabilizer* problem (finding the subgroup stabilizing a subset as a whole) [Lu1]. This problem instance achieved notoriety when valence-$(b+1)$ graph isomorphism was shown to be polynomial-time-reducible to it. Those familiar with the reduction will recall that it, too, appeared inherently sequential. Nevertheless, we now show that it can be replaced by an NC reduction. Though set-stabilizer is not yet available in parallel, this reduction, together with pointwise-set-stabilizers, yields an NC algorithm to test isomorphism in a subclass of the class of bounded valence graphs; this case requires methods of [Lu1] to establish even sequential polynomial time.

2 Definitions and preliminaries

We assume familiarity with the complexity class NC ([Pi], [Co]), informally, the class of problems solvable in polylog ($= \log^{constant} n$) time using a polynomial number of processors. We refer to any standard text, e.g., [Ha], for basic facts about groups.

We write $H \leq G$ if H is a subgroup of G and $H \triangleleft G$ if H is a normal subgroup of G. For $H \leq G$ the *normal closure* of H in G is the smallest normal subgroup containing H, the *centralizer* of H in G is the set of elements in G that commute with all elements in H. A group is called *simple* if it has no normal subgroups. If $\mathcal{T}$ is a collection of isomorphism types of simple groups, a $\mathcal{T}$*-semisimple* group is a direct product of groups of these types, we write T*-semisimple* if there is just one group T in $\mathcal{T}$ and *semisimple* if the class does not require explication. The *composition factors* of a group G are obtained by taking the quotient groups G_i/G_{i+1} in any series

$$1 = G_r \triangleleft \cdots \triangleleft G_1 \triangleleft G_0 = G$$

where these quotients are simple. The *socle* of a group G is the subgroup generated by all minimal normal subgroups and is denoted $Soc(G)$.

The group of all permutations of an n-element set A is denoted $Sym(A)$, or S_n if the specific set is not essential. We say that a group G *acts on* A if there is a homomorphism $G \rightarrow Sym(A)$, then for $a \in A$, $\gamma \in G$, we let a^γ denote the image of a under the permutation induced by γ and the *orbit* of a is $\{a^\gamma : \gamma \in G\}$. The permutation group induced on a single orbit is called a *constituent* of G. We say that G is *transitive* on A if there is only one orbit. If G is a transitive subgroup of $Sym(A)$, we say

G is *regular* if, for any $a \in A$, only the identity of G fixes a. If G is transitive and $D \subset A$, D is called a *block* (for G) if for all $\gamma \in G$, either $D^\gamma = D$ or $D^\gamma \cap D = \emptyset$, and G is called *primitive* if there are no blocks that are proper subsets of A. If D is a block then the set of images of D is called a *block system* and an action of G is induced on the block system; the block system is *minimal* if that action is primitive. In algorithms, permutation groups will always be input and output via a set of generators.

A standard tool for permutation group computation is a *strong generating set* [Si, Section 4]. As generalized in [FHL1], an SGS for G presumes any tower of subgroups

$$1 = G_r \leq \cdots \leq G_1 \leq G_0 = G.$$

An SGS is then the union of systems B_i of left coset representatives for G_i mod G_{i+1}. Hence, any $\alpha \in G$ has a unique representation $\alpha = \beta_0\beta_1 \cdots \beta_{r-1}$ with $\beta_i \in B_i$. Clearly, if we have an NC-construction of an SGS, we would know $|G|$. Note, though, that we would not necessarily have a membership test, for "sifting" (that is, factoring; see [FHL1]) would appear to take $r-1$ steps, which could be linear in the size of the underlying set. Thus, we shall insist that SGS's are *effective*, in the sense that they come along with an NC-procedure for determining the unique factorization of elements of G. It is useful to observe that an SGS for G/N, pulled back to G, appended to an SGS for N, gives an SGS for G.

3 Brief statement of results

We give NC-algorithms for testing isomorphism in a significant class of graphs. This includes the class of vertex-colored graphs with bounded color multiplicity. Moreover, the color multiplicity bound can actually be allowed to grow to $O(\log n)$ (n is the total number of vertices) provided the color-valences are kept bounded. (for colors C_i, C_j, the $i-j$ color valence is the maximum number of C_i-neighbors of a C_j vertex). It is worth noting that the algorithm of [Ba], [FHL1] would require sequential time $O(n^{\log\log n})$ for this extended class (though it can be dispatched in polynomial time by methods of [Lu1]).

It is also shown that isomorphism-testing of valence-$(b+1)$ graphs is NC-reducible to finding set stabilizers in groups in Γ_b. The instance of bounded-color-valence graph isomorphism mentioned in the last paragraph makes use of a modification of this reduction.

Machinery for the above includes new parallel algorithms for dealing with permutation groups that are presented only by generators, focussing on but not restricted to, groups in the class Γ_b. We show that the following problems are in NC for G in Γ_b. (Sections 5,6.)

(1) Find the order of G

(2) Test whether a given permutation belongs to G

(3) Find the normal closure of a given subgroup.

(4) Find all the composition factors of G.

(5) Find the centralizer in G of a given normal subgroup. In particular, find the center of G.

(6) Find the pointwise-set-stabilizer of a given subset of the permutation domain.

Note that (1)-(4) have been solved previously only for solvable groups, (5) and (6) only for nilpotent groups [LM]. To break out of these restricted classes, it is necessary to develop analogues of the tools of linear algebra, in which vector spaces are replaced by direct products of not-necessarily-abelian simple groups (see Section 4).

We point out that the Γ_b restriction ensures that the primitive groups left by divide-and-conquer are manageable. There are other situations in which we can count on this. For example, when the orbit sizes are polylog (whence we introduce, on each in parallel, methods of [Lu2]), or when orders of the orbit constituents are polynomial. Problems (1) through (5) are in NC in such case. By way of contrast, we refer to the last remark in [LM] where is pointed out that the techniques therein would not always handle membership testing when the orbits are of size 5 or when the constituent groups have order 60.

Along with the new algorithmic tricks, the solution to problem (6) involves some investigation of the algebra, making in-depth use of group structure (Section 6). By comparison, we recall that the polynomial-time algorithm for pointwise set stabilizer [FHL1] is quite direct. In fact, it is the *starting point* in the polynomial-time machinery. It should be noted that pointwise-set-stabilizer is a key to the graph isomorphism applications (Section 7).

4 "Non-abelian linear algebra"

We consider a class of problems that require an extension of our ability to do linear algebra over small fields.

For $i = 1, \ldots, m$, let T_i be a simple group acting nontrivially (and so, faithfully) on a set A_i. Then $L = T_1 \times \cdots \times T_m$ acts in a natural way on the disjoint union $A = A_1 \dot\cup \cdots \dot\cup A_m$ (the ith coordinate acts in the ith set). Given $\Phi \subset L$, we seek NC solutions to the following three problems.

(I) Find the order of G, the group generated by Φ.

(II) Test membership in G.

(III) *Construct* an effective SGS of G.

For simplicity (in the non-technical sense), we shall assume here that the $|T_i|$ are small (polynomial in our problem size), so that a listing of the elements of T_i is available. However, we only need a presentation of T_i in which we can perform essential operations on T_i (verify simplicity, find an SGS, etc.) in polylog time, e.g., by results of [Lu2] it would suffice if T_i were represented on an polylog-size set.

Suppose that each T_i is cyclic of order p. In that case, L is naturally identified with an $m-$dimensional vector space over Z_p and the solution to these problems is standard linear algebra. It is not much more difficult to deal with the case when the T_i are abelian but not all isomorphic for then L is a direct product of vector spaces over various Z_p. As shown by McKenzie [Mc], these vector space factors are obtainable in NC by taking suitable powers of the generators, and the solutions to our problems involve parallel computation in these factors. But what happens to the "standard" methods when we pass to non-abelian T_i? If we were considering only polynomial-time computation, then there is a direct analogue of Gaussian elimination (in fact, the methods of [FHL1] are interpretable in this sense). However, one's ability to find fast parallel solutions to linear algebra questions depends on alternate, and elegant, methods for computing *determinants* (e.g., [Cs], [Be], [BGH], [Mu]). There does not appear to be an analogue of determinants for n-sets of "vectors" in T^n when T is simple non-abelian.

Our new procedures rely, in part, on the following remarkable characteristic of products of non-abelian simple groups (see [Sc, Appendix]).

Lemma 4.1 *Let $T_i, i = 1, \dots, m$, be non-abelian simple group and suppose G is a subgroup of $\prod_i T_i$ that projects onto each factor. Then G is a direct product of "diagonal subgroups." To be precise, the T_i may be arranged in blocks of isomorphic groups so that, after a suitable renumbering of the factors,*

$$G = \mathrm{Diag}(T_1 \times \cdots \times T_{k_1}) \times \cdots \times \mathrm{Diag}(T_{k_{r-1}+1} \times \cdots \times T_{k_r})$$

In other words, having identified the groups in each block, G consists precisely of the elements of the form

$$(\alpha_1, \dots, \alpha_1), \dots, (\alpha_r, \dots, \alpha_r)$$

Note that, in Problems (I),(II),(III), we did not hypothesize projection onto each factor. The general answer to those questions awaits methods of Section 5. For now then, let us assume that G does project onto each T_i.

To answer Problem (I), it suffices to know which coordinates are *linked* in the diagonal blocks, for $|G| = \prod_{i=1}^{r} |T_{k_i}|$. To determine whether any T_i and T_j are so linked in G, we test whether a small modification of the action on A_i affects the action on A_j. Thus, we determine generators for the subgroup of G that fixes a point in A_i ("Schreier generators" are available see [Ha, p.96]) and test whether that subgroup fails to project onto T_j; if so the two simple groups are linked in a diagonal subgroup. In the linked case, we can observe the identifying isomorphism directly by noting a "partner" in T_j for each element in T_i.

It is an easy matter to use the above structure to solve Problem (II). However, we also observe that a solution to (III) will yield a membership test.

Problem (III) would be quite direct now were it not for the demand to *construct* the answer, for an SGS for each diagonal block is observable from an SGS of an involved T_i and a disjoint union of these will suffice. However, our requirement is to construct the SGS from Φ, specifically, via a program that computes, at each stage, products, inverses, and powers of previously computed elements. The reason is that the problems under consideration form but one component of the applications in mind. While our group G is acting in this elementary fashion on A, it also exists elsewhere on a larger domain. How would we extend these blindly-listed SGS elements on A to the larger domain? But, if they have been constructed, we already have the extensions, i.e., assuming we have always computed products, inverses, and powers on the larger domain. (In Section 8 we indicate an instance of a complexity gap between the problems).

Considering Problem (III) then, assume we have determined the r diagonal blocks, so that $G \cong T_{k_1} \times \cdots \times T_{k_r}$. Focussing on this identification, we construct, for each coordinate i, a set B_i of r-tuples of G of the form $(1, \dots, 1, \alpha_i, 1, \dots, 1)$, where α_i ranges over T_{k_i}. Then $\bigcup B_i$ is an SGS and the factorization of $\gamma \in G$ through $\prod B_i$ is immediate. (Note, it would have sufficed to let B_i be an SGS for T_{k_i}). It is easy to see that it suffices to have one such $\alpha_i \neq 1$ in hand, for conjugates of α_i will generate T_{k_i}. We converge on such elements in $\log r$ stages, doubling, in each stage, the number of coordinates that are trivial. Suppose that $i = 1$ and we have

$$\alpha \ = \ (\alpha_1, 1, \dots, 1, \alpha_{k+2}, \dots, \alpha_r),\ \alpha_1 \neq 1$$

$$\beta \;=\; (\beta_1, \ldots, \beta_{k+1}, 1, \ldots, 1, \beta_{2k+2}, \ldots, \beta_r),\ \beta_1 \neq 1$$

so that α and β each have at least k 1's, but in non-overlapping places (we assume that α and β were constructed, simultaneously, in the previous round). We may assume, by substituting a conjugate of α if necessary, that α_1 and β_1 do not commute. Then $\gamma = \alpha^{-1}\beta^{-1}\alpha\beta$ has a 1 in coordinates 2 through $2k+1$ and is $\neq 1$ in the first coordinate.

We require a solution in NC to one other problem involving the above $G \leq L$. This, too, is an analogue of something obtained previously in a classical linear algebra setting. We suppose now that another group, P, is given that acts (as automorphisms) on L. We need an algorithm for

(IV) Find the smallest subgroup H of L such that $G \leq H$ and H is closed under the action of P.

In the abelian case this is reducible to the problem of finding the smallest subspace containing a given set of vectors and closed under a given set of linear transformations, a problem shown to be in random NC in [LM], and so now in NC by virtue of [Mu]. As with the above, the algorithm for the non-abelian case must follow another tack.

Consider first the case where P acts transitively on $\{T_i\}$ (note, non-abelian simple factors are necessarily permuted amongst themselves by any automorphism). We can increase G (staying within target H) so that linked collections of T_i's (diagonal blocks) have the same size. For, suppose this does not hold. Let $\mathcal{C}$ be a smallest linked collection. We may assume $T_1 \in \mathcal{C}$ and we take any $\alpha \neq 1$ in B_1. For each T_j take $\pi_j \in P$ for which $T_1^{\pi_j} = T_j$ (naturally, all in parallel), and add all α^{π_j} to G. Recomputing the linked collections, every T_j lies in a block of size at most $|\mathcal{C}|$. If, at this point, a new block is strictly smaller than $\mathcal{C}$, then a similar process will lead to a split of $\mathcal{C}$. But the smallest section in that split has at most half the size $\mathcal{C}$. Repeating all of this at most $\log r$ times, we obtain equal-size collections. If now the image of any generator of (new) G is mapped into G by each generator of P, we have the desired H. If not, we have found an element that must be added to G. This necessarily forces the split of some linked collection and, once again, one part has at most half the size. Repeating all of this at most $\log r$ times, the new G is closed under the action of P.

In the non-transitive case, we perform the above (in parallel) on each P orbit in $\{T_i\}$. The new G may not yet be closed under P since the image of an element may induce a legal element within each of these orbits but not one which is consistent (with the diagonals) across the two orbits. We can discover all such anomalies by focussing on pairs of orbits (all pairs in parallel, of course). The images under P of the SGS will either be consistent across the pair, whence there is a rightful link, or else they include an element that breaks the link for a pair of T_i's, whence images of that element are used to break all links across the two orbits.

5 Basic group algorithms

We need a refinement of a tool that has been used for efficient permutation-group computation. The notion of a *structure forest* was defined in [LM], though it has predecessors in [FHL2], [GHLSW], [BL], among other places. A structure forest for a permutation group G is a forest on which G acts as automorphisms (fixing the roots), whose leaves form the given permutation domain, and such that no non-trivial levels can be inserted that are consistent with the G-action. Denoting by $G(v)$ the permutation group induced by the stabilizer of node v on the children of v, the latter condition asserts that $G(v)$ is always primitive.

As noted in [LM], NC contains the problem of computing a structure forest.

Structure forests are used, typically, to guide divide-and-conquer procedures that are natural to permutation groups - first dividing the set into orbits, then dividing an orbit into a minimal set of imprimitivity blocks, then passing to problems on the subgroup that fixes one or all blocks so that intransitivity is restored, etc. Such algorithms work particularly well for sequential computation with groups in Γ_b ([Lu1]) since we are assured ([BCP]) that the induced primitive actions, i.e., of node-stabilizers on children, involve groups of polynomially-bounded order (the exponent can be shown to be $b \log b + \mathit{constant}$). For the parallel algorithms, however, we shall generally need even more divide-and-conquer and have to dig into the structure of the primitive groups to get it.

Thus, we define an *augmented structure forest* (ASF) *for G acting on A* to be a structure forest F together with an assignment to each node $v \in F$ of a tower of normal subgroups of G

$$1 = G(v)_{m(v)} \lhd \cdots \lhd G(v)_1 \lhd G(v)_0 = G(v) \qquad (1)$$

with semisimple quotients $G(v)_i/G(v)_{i+1}$, and such that the induced action of G on $\{G(v)\}_{v \in F}$ induces, in turn, isomorphisms between subgroups at corresponding places in the towers. The following lemma provides the tool for constructing ASF's in our applications.

Lemma 5.1 NC *contains the problem of constructing an augmented structure forest from a given structure forest F for G if, for each node $v \in F$, either*

(i) *the order of $G(v)$ is polynomial, or*

(ii) *the degree of v is polylog.*

In particular, an ASF *for a group in Γ_b is* NC-*constructible.*

Indication of proof: We restrict our attention to the case when nodes satisfy (i) (the other case uses results in [Lu2]). It suffices to construct, at any given v, a normal series with semisimple quotients, for we need only construct one such tower for a selected v in each G-orbit, copying it (actually conjugating it), using any available element of G to each other point in the orbit. A tower can be constructed bottom-up by starting with $Soc(G)$ (NC-computable in case (i)) then considering, recursively, the quotient group modulo $Soc(G)$. □

Remark. For the remainder of this section, we state results for groups in Γ_b since this class guarantees property (i) in Lemma 5.1 in *any* permutation representation. However, the results apply to particular permutation groups as long a structure forest is available satisfying (i) or (ii) at each node. Note, we do not yet know this to be the case for the result of Section 6 (see comment in Section 8).

Our basic tool is

Theorem 5.2 NC *contains the problem of computing an* SGS *for a given permutation group in Γ_b.*

Indication of proof: With the help of an ASF F, the idea is analogous to the constructions in [LM]. We define a series of normal subgroups of G

$$1 = G_m \lhd \cdots \lhd G_1 \lhd G_0 = G, \qquad (2)$$

with $m = O(\log^2 n)$, and construct, inductively, an SGS for each G/G_{i+1} using an SGS for G/G_i (the SGS for the quotient is retained as a set of inverse images in G). The series in (2) is a refinement of the series

$$1 = K_h \lhd \cdots \lhd K_1 \lhd K_0 = G, \qquad (3)$$

in which K_i is the subgroup of G that fixes all the nodes at level i of F (roots are at level 0). The quotient group K_i/K_{i+1} then captures the action of K_i on the nodes at level $i+1$. We refine series (3) to series (2) by inserting, at each i,

$$K_{i+1} = H_{im_i} \lhd \cdots \lhd H_{i1} \lhd H_{i0} = K_i$$

where H_{ij} is the set of elements whose restriction to each v with $\ell(v) = i$ lies in $G(v)_j$ (see (1) above) and $m_i = \max\{m(v)\}_{\ell(v)=i}$ (we are letting $\ell(v)$ denote the level of v in the F). Note that we do not have the G_i in hand to start. We do know, however, that G_i/G_{i+1} is a subgroup of the semisimple group $L_{jk} = \prod_{\ell(v)=k} G(v)_j/G(v)_{j+1}$, for appropriate j, k. We accumulate elements of this group by "sifting" through the SGS $B_0, \ldots, B_q$ for G/G_i (as in [LM], we define the *sift* of γ to be the unique $\sigma \in G_i$ so that $\gamma = \beta_0 \cdots \beta_q \sigma$ with $\beta_l \in B_l$). We sift the starting generators of G and all products in $B_s B_t$ for all $s \geq t$. Viewing the images of the sifts in L_{jk}, the subgroup generated by these is then closed under the action of G. This is done in the abelian factors in L_{jk} by linear algebra as in ([LM]) and then in the non-abelian part by the algorithm indicated in Section 4 (warning: the present G plays the role of P in Problem (IV)). There results an SGS for G_i/G_{i+1} which is then appended to that of G/G_i. □

It is immediate that

Corollary 5.3 NC *contains the problems of finding the order of, and testing membership in, a permutation group in Γ_b.*

Also,

Corollary 5.4 NC *contains the problem of finding a composition series for a permutation group in Γ_b.*

Indication of proof: The G_i/G_{i+1} in the proof of Theorem 5.2 are seen to be semisimple and their simple factors emerge in the construction of the groups. □

The following is an important tool.

Theorem 5.5 NC *contains the problem of finding the normal closure of a subgroup of a permutation group in Γ_b.*

Indication of proof: We are given $H \leq G$, where G is in Γ_b. The construction of an SGS for N, the normal closure of H, is similar to the construction in the proof of Theorem 5.2. This time, sift (into G_i) the given generators for H, the products $B_s B_t$, for all $s \geq t$, from a current SGS (for $N/(N \cap G_i)$), and the conjugates of the SGS via the generators of G. (See [LM, Theorem 1.3]). □

Corollary 5.6 NC *contains the problem of finding the kernel of an action (i.e., on a set other than the given permutation domain) of a permutation group in Γ_b.*

Indication of proof: We are given $G \leq Sym(A)$ and are considering a second action $\Psi : G \rightarrow Sym(D)$. Find an SGS for $\Psi(G)$, always keeping track of the inverse images in G. Sift the generators of G and the products $B_s B_t$ for $s \geq t$. Take the normal closure of the group generated by the sifts. □

For groups in Γ_b, Corollary 5.6 is superceded by the more general pointwise-set-stabilizer in Section 6, but it is used along the way to that result. It is also used in

Theorem 5.7 NC *contains the problem of finding the centralizer of a normal subgroup of a group in* Γ_b. *In particular, the problem of finding the center of a group in* Γ_b *is in* NC.

Indication of proof: The problem is reducible to finding kernels [Lu2].

6 Pointwise set stabilizers

For a permutation group G in the class Γ_b, we need to determine generators of the subgroup that fixes all the points in a specified subset of the permutation domain. This problem has previously been shown to be in NC only for nilpotent groups [LM] (i.e., direct products of p-groups).

It is useful to illustrate one of the key underlying ideas with the following subcase; it is, in a sense, the "smallest" subcase not covered by the algorithms of [LM]. Start with the 6-element symmetric group $Sym(A)$ acting on the 3-element set, A. Then $Sym(A)^n$ acts in a natural way on the disjoint union $A_1 \dot{\cup} \cdots \dot{\cup} A_n$ of n copies of A. We suppose now that we are given generators Φ for a subgroup G of $Sym(A)^n$ and we have specified one point in each A_i to be fixed. The algorithm makes strong use of a special normal subgroup of G. We want the elements of G that induce, in every A_i, an element of the (3-element) alternating subgroup $Alt(A)$. These comprise a normal subgroup N that is obtainable in NC (for it is the kernel of an induced action on $\prod_i Sym(A_i)/Alt(A_i)$). This subgroup N is a direct product of cyclic groups of order 3, i.e., a vector space over Z_3. Suppose $|N| = 3^y$. It is an easy matter, using linear algebra techniques, to find a set of y coordinates so that N induces the complete $Alt(A)^y$ on the set Y of corresponding orbits. We then use an appropriate canonical basis of $N = (Z_3)^y$ to modify (in parallel) each of the elements of Φ so that it fixes the target points in Y (in parallel). The modified Φ generates a subgroup H. Now, since $G = HN$, and H fixes some of the target points, the answer to the problem lies in HN_1, where N_1 is the subgroup of N that fixes this subset of target points. However, no non-trivial element of N can fix these points. We conclude first that $N_1 = 1$, and second that $H \cap N = 1$. It follows not only that the answer to the problem lies entirely in H but that H has no elements of order 3. So H is, essentially, a vector space over Z_2, and the pointwise-set-stabilizer is a subspace obtainable by linear algebra.

The general case involves reduction to the situation where G is primitive on each orbit and then involves the location of an appropriate analogue of the N of the previous paragraph.

Reducing to the primitive case is easy: We may assume that there is at most one point to be fixed in each orbit (else make copies of the orbit designating different points in each one). On each such orbit we build a minimal block system. The subgroup fixing the designated point must fix the block containing that point. So the intermediate goal involves fixing the block, a 'point' in a primitive action.

We show:

Theorem 6.1 *Let G in Γ_b be a subgroup of $Sym(A)$, $|A| = n$, with G acting primitively on each orbit. Suppose a set of points in A have been designated as "target" points (to be fixed). Then there is a normal subgroup N of G and a collection Y of orbits such that*

(a) *The subgroup of G that fixes the target points in the orbits in Y induces a proper subgroup in a significant fraction, i.e., $1/(\log^c n)$, of the orbits that contain target points. ($c = c(b)$).*

(b) *$Y = Y_1 \dot{\cup} \cdots \dot{\cup} Y_q$ with $q = O(\log n)$ and, letting N_i be the subgroup of N that fixes the target points in $Y_1 \dot{\cup} \cdots \dot{\cup} Y_{i-1}$, then N_i restricted to Y_i is a direct product of its constituents there, each of which is transitive and T-semisimple for some T.*

(c) *NC contains the problem of finding Y, N, $\{Y_i\}$, $\{N_i\}$.*

Roughly, Theorem 6.1 is applied as follows: First consider the case when the targeted-orbit constituents have polynomial-size. Consider, in each orbit, the primitive action on a minimal block system, marking the targets. Treating $Y_1, Y_2, \ldots$ in succession, N_i is used, as in the illustration to cut down the present G to HN_{i+1}, which fixes the target points in Y_i. Repeat, focussing only on the constituents that have not been affected. By (a), all constituents will have been cut in at most $O(\log^{c+1} n)$ passes. Now restore primitivity in each orbit by considering the action on minimal block systems and repeat all of the above. Since the constituents have polynomial-size, we reach a pointwise-set-stabilizer in at most $O(\log n)$ repetitions. Finally, we proceed to the general case by building a structure forest (which has $O(\log n)$ levels) and work down the tree fixing marked nodes that are ancestors of target points. At each level, we are dealing with constituents of polynomial-size.

The N in Theorem 6.1 is constructed out of the socles of the primitive groups. But the location of N, Y satisfying both the *largeness* condition of (a) and the *independence* condition of (b) requires additional ammunition. It is easy to show

Lemma 6.2 *Let $\mathcal{T}$ be a collection of isomorphism types of simple groups. For any finite group G, there is a unique minimal $N \triangleleft G$ such that G/N is $\mathcal{T}$-semisimple.*

We call N the *residual* of G with respect to $\mathcal{T}$, denoting it by $Res_{\mathcal{T}}(G)$. A *residual tower* in G is a normal series

$$1 = R_m \triangleleft \cdots R_1 \triangleleft R_0 = G$$

in which $R_{i+1} = Res_{\mathcal{T}_i}(R_i)$ for some $\mathcal{T}_i$.

Lemma 6.3 *If G is primitive then the smallest nontrivial group in any residual tower is always $Soc(G)$.*

Now suppose $G \leq G_1 \times \cdots \times G_r$. Denoting the ith coordinate projection by pr_i, assume that $pr_i(G) = G_i$. Assort the composition factors of G into a sequence of classes as follows: the abelian composition factors comprise the first class; each of the other classes contain all the non-abelian composition factors of a given order (most classes contain just one group) and these classes are sorted by increasing order of the groups therein. Now let $\mathcal{T}$ be the first class such that $Res_{\mathcal{T}}(G_i) \neq G_i$ for some i, and let $\mathcal{R}(G)$ denote the subgroup of elements that project into $Res_{\mathcal{T}}(G_i)$ at all i. For G in Γ_b, $\mathcal{R}(G)$ is NC-constructible when G_i's have polynomial size. It follows easily from the definition of residuals that $pr_i(\mathcal{R}(G)) = Res_{\mathcal{T}}(G_i)$ (in fact, this relation does not require the minimality of $\mathcal{T}$), so that unless the projections are trivial, they still contain the socles in primitive coordinates (Lemma 6.3). We shall want, in fact, a series of groups with this property:

$$1 \triangleleft \cdots \triangleleft \mathcal{R}(\mathcal{R}(\mathcal{R}(G))) \triangleleft \mathcal{R}(\mathcal{R}(G)) \triangleleft \mathcal{R}(G) \triangleleft G \qquad (4)$$

The next two lemmata isolate critical features of series (4).

Lemma 6.4 *Let N be a group in the series (4). Then, for any subgroup $H \leq G$, $pr_i(H) = G_i$ implies $pr_i(H \cap N) = pr_i(N)$.*

Lemma 6.5 *Let $G \leq Sym(A)$ with G in Γ_b and suppose that the orbit constituents $\{G_i\}$ are primitive. Then the length of series (4) is $O(\log^c |A|)$ where $c-1$ is the number of orders taken on by non-abelian simple subgroups of S_b.*

Assuming the conditions of Theorem 6.1, each $Soc(G_i)$ will appear as the projection of a subgroup in series (4). We may (by Lemma 6.5) select N in that series that projects onto the socles in at least $1/(\log^c n)$ of the targeted orbits; let X be that collection of orbits (X is the "significant fraction" of (a)). To understand Y, we must look more closely at the socles of primitive permutation groups.

Lemma 6.6 (O'Nan,Scott [Ca]) *Let $G \leq Sym(A)$ be a primitive permutation group, let $a \in A$, and let $K = Soc(G)$, $K_{(a)}$ = the subgroup of K that fixes a. Then K is $T-$semisimple for some simple group T and one of the following holds*

(i) *K is abelian, regular on A, and is the unique minimal normal subgroup of G; and $K_{(a)} = 1$.*

(ii) *K is non-abelian, transitive on A, and is the unique minimal normal subgroup of G.*

(iii) *$K = K_1 \times K_2$, where K_1, K_2 are isomorphic, non-abelian, are each regular on A, and are the unique minimal normal subgroups of G; and $K_{(a)} =$ $\mathrm{Diag}(K_1 \times K_2)$.*

The cases (i),(ii),(iii) involve different sources of N, Y, so assume only one holds for above X (by cutting down to $\geq 1/3$ of these orbits).

Case (i): Of course, this case will most resemble the illustrated example. Assuming any ordering, $O_1, O_2, \ldots$, of the orbits in X, determine (in parallel for every i) the subgroup $N_i \leq N$ that vanishes on $\bigcup_{j<i} O_j$ (this is a kernel). Then $pr_i(N_i)$ is a normal subgroup of G_i, so by (i), it is trivial or $Soc(G_i)$. Take Y to be the set of orbits where the latter holds. Thus, if any element of N fixes the target points in Y it is, by (i), trivial on Y and therefore (by choice of Y) trivial on X. Furthermore, N restricted to the orbits in Y is precisely the direct product of its constituents there. By linear algebra methods (augmented by McKenzie's methods [Mc] to separate the characteristics) one can so factor N (each factor being trivial on all but one orbit in Y). As in the illustration, we replace G by $HN_1 = H$. Again, $H \cap N$ is trivial in X and so, by Lemma 6.4, the group has been cut down in every orbit of X.

Case (ii): On the orbits in X, N projects onto each non-abelian socle and therefore acts as a product of diagonal subgroups (Section 4). Determine the diagonally-linked factors. Since the socles are the unique minimal normal subgroups, a link across two orbits implies a link between the entire socles in those places. Form Y by selecting one orbit corresponding to each linked collection of socles. By the linking, if for some $H \leq G$, $H \cap N$ does not induce N on any of the orbits in Y, then it does not induce N on any of the orbits in X and so, by Lemma 6.4, H will be proper in G in all these places. By the selection of Y and Section 4, we can find SGS elements that each act non-trivially on only one orbit. These may be used to modify (in parallel) the generators of G so that they fix the targeted points in Y. Again, G is replaced by HN_1. Since the latter group fixes a point in each orbit of Y, it has lost part of the socle there. By the above remarks, the group has been cut down in every orbit of X.

Case (iii): This is the only case in which $q > 1$ (see (b) in Theorem 6.1). There are now two minimal normal subgroups in each socle; we refer to these as *socle-parts.* Again, determine the diagonally-linked factors in the action of N on the orbits in X. The links crossing two orbits may not extend to the entire socles but could link just one socle-part to a socle-part. In fact, we may now consider the diagonally-linked blocks on the socle-parts occurring in X. We call the number of such diagonal collections the *rank* of N in X. Form the graph whose vertices are the orbits in X and whose edges represent the linkings of socle-parts. We select a maximal independent set in this graph ([KW],[Lub]) and let Y_1 be the corresponding set of orbits. In each orbit in Y_1 select one of the two socle-parts; by their independence we can use the collection $\mathcal{D}_1$ of these to fix the targeted points in Y_1. Note that, by (iii), we shall have introduced new links in N. Consider next the set $\mathcal{D}_2$ of diagonally-linked socle-parts that were not represented (in either socle-part, selected or not) in Y_1. Form the collection Y_2 by selecting, for each member of $\mathcal{D}_2$, an orbit in which it is paired to a factor that was in Y_1 (it exists since Y_1 was maximal). Use the members of $\mathcal{D}_2$ now to fix the targeted points in Y_2. Again, this establishes new links to groups that are already in or linked to $\mathcal{D}_1$. But then the rank of the remaining part of N is at most half of that of the original (for it is half the size of the maximal independent set). We repeat the entire process on the "untouched" orbits in X. In at most $\log 2|X|$ rounds, the rank is reduced to 1, so no full socles remain in X. We conclude by Lemma 6.4. □

Remark. The precise parallel complexity of the pointwise-set-stabilizer algorithm is dependent upon the length of series (4), which puts some function of b in the exponent (Lemma 6.5). If, however, the G_i are absolutely bounded the series length is constant. This case is all that is needed to take care of isomorphism-testing CG_b, for fixed b.

7 Graph isomorphism

To apply pointwise-set-stabilizer to isomorphism-testing in CG_b, we reduce the latter, as usual, to computing automorphism groups ([Ba1],[Lu1]). For these, we consider the group $G = \prod Sym(C_i)$; G acts, simultaneously, on the sets of subsets of each $C_i \times C_j$, in which we want to fix the 'point' corresponding to the set of edges from color class C_i to color class C_j. For this, then, one needs only pointwise-set-stabilizer for permutation groups with *bounded orbits.*

We briefly outline the NC-reduction of trivalent graph isomorphism to subset stabilizer for 2-groups (the higher-valence reduction works similarly). We remark first that, given a set-stabilizer algorithm, we can find subgroups stabilizing a relation (i.e., by stabilizing a subset of the square of the domain), and subgroups stabilizing any number of color classes (i.e., by stabilizing the relation "belong to same class" and following with our "pointwise" stabilization of the different color classes).

It is sufficient ([FHL2],[Lu1]) to compute $Aut_e(X)$, the group of automorphisms stabilizing edge e in a connected trivalent graph X. We insert a new vertex v in the middle of e and refer then to $Aut_v(X)$. Also, we ignore, in a first pass, all "cross edges", that is, edges between vertices at the same distance from v, for these can be accommodated via a set-stabilizer application in the group otherwise obtained. For each x in X and each $r > 0$, we consider the subgraph $X_r(x)$ induced on "descendents" of x working "away from" the direction of v. We determine *all* $Iso(X_r(x), X_r(y))$, the sets of all rooted-isomorphisms where x and y are at the same distance from v; it is convenient to view such sets as 2-groups by expanding them to the group of all automorphisms of the *disjoint* union of the graphs. The trick is to double r in each round. The group $Iso(X_{2r}(x), X_{2r}(y))$ is formed in two stages. In stage one, we look at pairs of points w, z at level r out from x, y and look up whether $X_r(w)$ and $X_r(z)$ are isomorphic. Points so related induce a coloring. We cut down $I = Iso(X_r(x), X_r(y))$ so that colors are stabilized. We then use a construction reminiscent of a technique of Miller ([Mi1], see also [BL]). If we imagine that descendents of the distance r points are mutually disjoint, i.e., by temporarily making duplicate copies, I is extendible to level $2r$ by piecing together level r groups (a wreath product). In stage two, we must reconcile this illegally extended I with the real graph, i.e., we must cut it down so that classes of equivalent points (duplicates of same real point) are mapped to classes of equivalent points. Another application of set-stabilizer guarantees that. The action of the residual I on real points is then extracted. At last, $Aut_v(X)$ is derived from $Iso(X_n(v), X_n(v))$.

We comment, finally, on the essential ingredients in the modification of the above reduction to produce an NC-algorithm for the cited special case of bounded-valence isomorphism (bounded color valence, $O(\log n)$ points in a color class).

We exploit the fact that the groups I are in Γ_b, assuming a bound of $b+1$ on the color-valences (by an easy extension of results of [Lu1]). Within each color class then, the group has size $O(n^c)$ [BCP]. But when orbit-constituents have polynomial size, set-stabilizer reduces to pointwise-set-stabilizer (look at action on collections of cosets of "local" stabilizers, fix the "points" corresponding to the stabilizers themselves). Since orbit-constituents remain of polynomial-size when we square the domain, we can also stabilize relations on the set. One of the consequences of this observation is that we can omit many of the edges in the graphs when we make the Iso extensions from level r to level $2r$, provided we omit in a canonical fashion and the graph is still connected, for we can then use a set-stabilizer algorithm to cut back to the automorphisms stabilizing the set of omitted edges. We want to omit edges so that points at levels $> r$ are descended from at most one color class at level r. This can be done, for example, so that the ancestors with the least (in some ordering) color capture the descendents. Thus, we may assume that the descendents of different color classes at level r are disjoint. In extending Iso, let us first focus on one color class C at level r. Instead of attempting a wreath product (which may result in large orbits at level $2r$), we generate all the polynomial number of elements in the group acting on C. We reject any which map w to z when $X_r(w)$ and $X_r(z)$ are not isomorphic. (Note: we have also had to cut down $Iso(X_r(w), X_r(z))$ so that they fix the omitted edge set). Each element γ that remains induces a coset of isomorphisms (on the 'disjointed' graph). It is a coset of a group whose orbit constituents have polynomial-size. Extending our results to such cosets, we are able to stabilize the classes of equivalent points (again, duplicates of the same real point), so the action on the real graph is extracted. The answers are pieced together over all γ. Doing this in parallel across all C, we then know the elements of I that work in each color class. We can cut I down so that it lies in the legal group in each C (again by a point = coset stabilization). For each generator of the resulting group, we need list only one extended isomorphism; the rest of the group comes from considering the isomorphisms that fix all points at level r, and this is a direct product of subgroups of groups that were obtained with $\gamma = 1$.

8 Comments

The parallel complexity of membership-testing in general permutation groups remains open. In light of the machinery laid out herein, it is interesting to speculate about the methods that might be needed for this problem. Using the augmented-structure-forest approach, the principle obstruction to membership-testing is manifest. It arises in the instance of a complete symmetric group acting on a set of non-trivial blocks. If we view the "sliced" semisimple problem, (I), (II) of Section 4 are actually solvable. To be precise, one can recognize (in NC) that this is, indeed, the complete symmetric group (so that membership-testing is clear). But, the algebraic machinery that goes into that recognition is impressive. We can test that the group is at least 6-transitive, i.e., that every 6 point sequence maps to every other one. It is known that the only 6-transitive groups are the alternating or symmetric groups, and one can distinguish between the two. So the membership test does not come along with a ready solution to (III). One might speculate that the proof of the 6-transitivity result may carry more constructive information and it probably does. However, that proof cites the monumental classification of finite simple groups. Thus, the obstruction may simply lie in our ability to comprehend the algorithm.

We comment, also, on two interesting complexity gaps that have opened.

In these proceedings, Babai [Ba2] has borrowed some of our machinery in an algorithm for pointwise-set-stabilizer in the class of permutation groups with polynomial-size orbit constituents (without the Γ_b hypothesis). His methods, however, are *random*, putting the result into Las Vegas NC. We conjecture that that will be improvable to NC.

In both [Ba1] and [Lu1], it was observed that the polynomial-time isomorphism tests did not seem to guarantee computation of *canonical forms* in the graph classes. After a time, these issues were resolved [BL],[FSS]. We reopen it now, for we do not know how to compute canonical forms for CG_b in NC. Amongst the difficulties in extending the present method could be the canonical choice of maximal independent sets (case (iii) in Section 6.)

Acknowledgements

We are pleased to acknowledge rewarding conversations with L. Babai and W. Kantor.

References

[Ba1] Babai, L., *Monte Carlo algorithms in graph isomorphism testing*, Tech. Rep. 79-10, Dép. Math. et Stat., Univ. de Montréal, 1979.

[Ba2] Babai, L., *A Las Vegas - NC algorithm for isomorphism of graphs with bounded multiplicity of eigenvalues*, Proc. 27th IEEE FOCS, 1986.

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

[BGM] Babai, L., Grigoryev, D.Y., Mount, D.M., *Isomorphism of graphs with bounded eigenvalue multiplicity*, Proc. 14th ACM STOC, 1982, 310-324.

[BL] Babai, L., Luks, E.M., *Canonical labeling of graphs*, Proc. 15th ACM STOC, 1983, 171-183.

[BKL] Babai, L., Kantor, W., Luks, E.M., *Computational complexity and the classification of finite simple groups*, Proc. 24th IEEE FOCS, 1983, 162-171.

[Be] Berkowitz, S.J., *On computing the determinant in small parallel time using a small number of processors*, Univ. of Toronto, 1983.

[BGH] Borodin, A., von zur Gathen, J., Hopcroft, J., *Fast parallel matrix and GCD computations*, Information and Control, 52 1982, 241-256.

[Ca] Cameron, P.J., *Finite permutation groups and finite simple groups*, Bull. London Math. Soc., 13, 1981, 1-22.

[Co] Cook, S.A., *The classification of problems which have fast parallel algorithms*, Proc. 1983 Intern. FCT Conf., Lecture Notes in Comp. Sci. 158, Springer-Verlag, 78-93.

[Cs] Csanky, L., *Fast parallel matrix inversion algorithms*, SIAM J. Comp. 5, 1976, 618-623.

[FHL1] Furst, M., Hopcroft, J., Luks, E., *Polynomial time algorithms for permutation groups*, Proc 21st IEEE FOCS, 1980, 36-41.

[FHL2] Furst, M., Hopcroft, J., Luks, E., *A subexponential algorithm for trivalent graph isomorphism*, Tech. Rep.0-426, Dept. Comp. Sci., Cornell Univ., 1980.

[FSS] Fürer, M., Schnyder, W., Specker, E., *Normal forms for trivalent graphs and graphs of bounded valence*, Proc. 15th ACM STOC, 1983, 161-170.

[GHLSW] 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, 118-125.

[Ha] Hall, M., *The Theory of Groups*, Macmillan, 1959.

[KW] Karp, R.M., Wigderson, A., *A fast parallel algorithm for the maximal independent set problem*, Proc. 16th ACM STOC, 1984, 266-272.

[Lub] Luby, M., *A simple parallel algorithm for the maximal independent set problem*, Proc. 17th IEEE FOCS, 1985, 1-10.

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

[Lu2] Luks, E.M., *Computing the composition factors of a permutation group in polynomial time*, Dept. of Computer and Information Sciences Tech. Rep. 85-07, Univ. of Oregon, 1985. To appear in Combinatorica.

[LM] Luks, E.M., McKenzie, P., *Fast parallel computation with permutation groups*, Proc. 26th IEEE FOCS, 1985, 505-514.

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

[MC] McKenzie, P., Cook, S.A., *The parallel complexity of the abelian permutation group membership problem*, Proc. 24th IEEE FOCS, 1983, 154-161.

[Mi1] Miller, G.L., *Isomorphism of k-contractible graphs, a generalization of bounded valence and bounded genus*, Information and Control 56, 1983, 1-20.

[Mi2] Miller, G.L., *Isomorphism of graphs which are pairwise k-separable*, Information and Control 56, 1983, 21-33.

[Mu] Mulmuley, K., *A fast parallel algorithm to compute the rank of a matrix over an arbitrary field*, Proc. 18th STOC, 1986, 338-339.

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

[Sc] Scott, L.L., *Representations in characteristic p*, in "The Santa Cruz Conference on Finite Groups", 1980, Amer. Math. Soc., 319-322.

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