Permutation Groups and Polynomial-Time Computation

Eugene M. Luks

CIS 93-01
February 1993

Department of Computer and Information Science
University of Oregon

to appear in
GROUPS AND COMPUTATION
DIMACS Series in Discrete Mathematics and Theoretical Computer Science

Permutation Groups and Polynomial-Time Computation

EUGENE M. LUKS

ABSTRACT. We discuss aspects of computation in permutation groups assuming polynomial time as a measure of efficiency. Of particular interest are problems, such as finding the intersection of two groups, that resemble or generalize the problem of testing graph isomorphism. We also summarize the instances where the problems are known to be solvable in polynomial time and indicate methods that accomplish this. As with graph isomorphism, the computational complexities of the general problems are open, though we can demonstrate polynomial-time reductions and equivalences among them. A typical approach to such issues is shown to involve an NP-complete problem. Several open questions are listed.

1. Introduction

Our focus is on polynomial-time computability. Naturally, in employing so broad a brush, we do not pretend to delineate the present practical frontiers of computational group theory. Polynomial time is, on the other hand, a widely-recognized standard of tractability as well as a robust model in which to measure and compare efficiency. But, we leave even that point to be defended, or disputed, elsewhere. From our perspective, polynomial time provides, independently, a productive and elegant domain in which to study the structure of group-theoretic computation, while the group-theoretic setting provides insight into the structure of unresolved issues in computational complexity. Furthermore, this interface with theoretical computer science motivates attractive problems for the group theorist, a haunting example stemming from the failure of all efforts to develop a *provably* efficient method for testing graph isomorphism. Thus, where the state-of-knowledge about polynomial-time efficiency does not conform to current perceptions of "practical" efficiency, there lie the most tantalizing of the open questions.

We discuss permutation groups $G \leq \mathrm{Sym}(\Omega)$ that are input via generators. It is reasonable to insist that the generating set is "small"; e.g., of cardinality

1991 *Mathematics Subject Classification.* Primary 20B40,68Q25; Secondary 68Q15.

Research supported in part by NSF grant CCR-9013410.

This is the final version of this paper.

CIS-TR-93-01
University of Oregon
February 1993

$< |\Omega|^2$. With this understanding, polynomial time "in the input" translates to an $O(|\Omega|^c)$ bound, for some constant c, on the number of steps required in the worst-case. In this paper, we are not concerned with optimizing the exponent c. Such "low-level" complexity matters are, of course, of great interest and are closer to, even when not identical with, implementational concerns. Nevertheless, these are, in the present context, extraneous issues and we gain more insight into polynomial-time matters by ignoring them. Thus, we *avoid* specification and justification of precise constants in the exponents, favoring clarity of the polynomial-time status over optimization of time or space requirements.

We sample not only what *is* in polynomial time but also a range of problems that, to date, have not met this standard. Our concentration is on issues that are motivated by graph-isomorphism testing. Such issues include important and standard group-theoretic problems, including the computation of intersections of permutation groups, centralizers of elements and stabilizers of subsets. As with graph isomorphism, these problems are not considered hard in practice. Nevertheless, no algorithm has been shown to require less than exponential time in the worst case. On the other hand, there seems some evidence that "decision" versions of the problems are not NP-complete. If that is the case, then another level of difficulty (assuming P$\neq$NP) is represented by the related problem of finding lexicographically least elements in double cosets, for we show (the decision version of) this one is NP-complete.

On the positive side, we offer various proofs of polynomial time. While there is a large polynomial-time library (see [18]), our emphasis again is on instances of the problems that are related to graph isomorphism and its group-theoretic analogues. For most of the problems, there are efficient procedures for solvable groups.

Our discussion also brings out several open questions.

In Section 3, we review basic polynomial-time tools for dealing with permutation groups. Consistent with computational experience, these efficient techniques are rooted in methods of C.C. Sims. However, in Section 4 we move on to some problems for which efficiency has not been theoretically substantiated. Although their complexity is unknown, they can be shown to be polynomial-time equivalent. The NP-hardness of the aforementioned lex-least problem is proved in Section 5. To better understand the difficulty, we offer two proofs. One of these involves abelian groups with small orbits. In that case, we can explain away the difficulty in terms of the choice of linear orderings on the permutation domain. To be precise, we go on, in Section 6, to show that, with an ordering based upon the orbit/imprimitivity-block structure of the group, the lex-least problem is in polynomial time for groups with restricted noncyclic composition factors, thus automatically including all solvable groups. This result, in turn, recovers instances where the graph-isomorphism-inspired problems are in polynomial time. Other polynomial-time instances are discussed in Section 7, where it is seen that the search for subgroups is apparently made easier if the targeted subgroup is normal. For instance, in Sections 7 and 8, we show that the cores of intersections, centralizers and set stabilizers can be found in polynomial time. In Section 8, we give samples of results of Kantor and Luks that indicate how the procedures, as well as the open problems, are extended to quotients of permutation groups, the theme being that, as far as polynomial time is concerned, the problems are no harder when dealing with quotients. There is, however, remarkable additional overhead in generalizing to quotient groups, for some problems that previously had elementary solutions now seem to require much deeper theory. In particular, the solutions make use of Sylow subgroups which are accessible in polynomial time only through results of Kantor that use consequences of the classification of finite simple groups. By contrast, we present, in Section 9, an approach to finding p-cores (maximal normal p-subgroups). While implemented solutions to this problem typically use Sylow subgroups, we describe an elementary, self-contained method that bypasses these. Some other related problems are listed in Section 10, none of which are known to be in polynomial time. In fact, to date, they seem to represent various levels of difficulty, thereby opening up questions even about the existence of polynomial-time reductions between the problems.

2. Notation and Preliminaries

Let G be a group. We write $H \leq G$, respectively $H \trianglelefteq G$, to indicate H is a subgroup of G, respectively a normal subgroup; $H < G$ and $H \triangleleft G$, respectively, indicate strict inclusion. We say H is *subnormal* in G, denoted $H \triangleleft\triangleleft G$, if there exist groups $H_1, ..., H_m$ such that $H \trianglelefteq H_1 \trianglelefteq \cdots \trianglelefteq H_m \trianglelefteq G$. If $H \leq G$, a *right (left) transversal* for H in G is a complete set of right (left) coset representatives for H in G. A *right (left) subcoset* of G is a coset Hx (xH) of a subgroup $H \leq G$. For $A \subseteq G$, $\langle A \rangle$ denotes the subgroup generated by A. For $g, h \in G$, let $h^g = g^{-1}hg$, the *conjugate* of h by g, and let $[g,h] = g^{-1}h^{-1}gh$, the *commutator* of g and h. For $A \subseteq G$, $A^g = \{a^g \mid a \in A\}$; if $H \leq G$, the *centralizer* of A in H is $\mathrm{C}_H(A) = \{h \in H \mid a^h = a, \forall a \in A\}$; for $a \in G$, $\mathrm{C}_H(a) = \mathrm{C}_H(\{a\})$. For subgroups $H, K \leq G$, the *normalizer* of H in K is $\mathrm{N}_K(H) = \{k \in K \mid H^k = H\}$; we let $H^K = \langle \bigcup_{k \in K} H^k \rangle$, this is the *normal closure* of H in $\langle H, K \rangle$, namely the smallest normal subgroup of $\langle H, K \rangle$ that contains H. For $H \leq G$, the *core* of H in G is $\mathrm{Core}_G(H) = \bigcap_{g \in G} H^g$, it is the largest subgroup of H that is normalized by G.

We denote by $\mathrm{Sym}(\Omega)$ the symmetric group on the finite set Ω. Suppose G acts on Ω, that is, there is a homomorphism $\phi : G \to \mathrm{Sym}(\Omega)$; if $G \leq \mathrm{Sym}(\Omega)$, ϕ is understood to be the natural injection. For $\omega \in \Omega$, $g \in G$, ω^g denotes the image of ω under $\phi(g)$; for $\Delta \subseteq \Omega$, $\Delta^g = \{\omega^g \mid \omega \in \Delta\}$; for $\omega \in \Omega$, the *orbit* of ω is $\{\omega^g \mid g \in G\}$ and is denoted ω^G. For $\omega \in \Omega$, G_ω denotes the subgroup fixing ω, namely $\{g \in G \mid \omega^g = \omega\}$; for $\Delta \subseteq \Omega$, G_Δ denotes the *pointwise stabilizer* of Δ, namely $\bigcap_{\omega \in \Delta} G_\omega$; if an ordering $\omega_1, \ldots, \omega_n$ of Ω is understood, and $\Delta_i =$

$\{\omega_1, \ldots, \omega_{i-1}\}$ then $G^{(i)} = G_{\Delta_i}$ (in particular, $G^{(1)} = G$ and $G^{(n)} = 1$). For $\Delta \subseteq \Omega$, the *(set) stabilizer* of Δ in G, denoted $\mathrm{Stab}_G(\Delta)$, is $\{g \in G \mid \Delta^g = \Delta\}$; G *stabilizes* Δ if $\mathrm{Stab}_G(\Delta) = G$. For $a \in \mathrm{Sym}(\Omega), \Delta \subseteq \Omega$, let a^Δ denote the induced function $\Delta \rightarrow \Delta^a$, and for $A \subseteq \mathrm{Sym}(\Omega)$, $A^\Delta = \{a^\Delta \mid a \in A\}$. In particular, if $G \leq \mathrm{Sym}(\Omega)$ and G stabilizes Δ then $G^\Delta \leq \mathrm{Sym}(\Delta)$. A subset $\Delta \subseteq \Omega$ is called a *block* for G if, for all $g \in G$, either $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$. We say that G acts *transitively* on Ω if Ω consists of a single orbit; G acts *primitively* if it acts transitively and there is no block Δ for G with $1 < |\Delta| < |\Omega|$; G acts *regularly* if it acts transitively and $G_\omega = 1$ for any (all) $\omega \in \Omega$.

We refer to Chapter 1 of [30] for elementary results in permutation groups that are not specifically recalled herein.

Unless otherwise indicated, we suppose $n = |\Omega|$. It is useful to recall that an increasing chain of subgroups of $\mathrm{Sym}(\Omega)$ has polynomially-bounded length; for example, Lagrange's Theorem implies that the length cannot exceed $\log_2 n! = O(n \log n)$ (in fact, a linear bound can be proved [1]). Though not explicitly stated, this is often essential to the verification of polynomial running times. For algorithmic purposes, unless indicated otherwise, it is assumed that subgroups of $\mathrm{Sym}(\Omega)$ are specified (input or output) by generating sets.

A problem is said to be in *polynomial time* if it is solvable in $O(m^c)$ steps where m is the size of a reasonable encoding of the input. In saying that a problem $\mathcal{A}$ is *polynomial-time reducible* to a problem $\mathcal{B}$, we mean that if $\mathcal{B}$ is in polynomial time then $\mathcal{A}$ is in polynomial time. However, in the case of reductions between two *decision* problems, i.e., those with a "yes"/"no" answer, we always intend *Karp reductions*, that is, there is a polynomial-time-computable mapping of instances of $\mathcal{A}$ to instances of $\mathcal{B}$, so that "yes", "no" instances map, respectively, to "yes", "no" instances. Two problems are *polynomial-time equivalent* (in either sense) if there are reductions in both directions. See, for example, [12], for elaboration of these issues, including the need for Karp reductions, as well as formal definitions of the classes P, NP, and NP-Complete. Note that these particular terms apply only to decision problems.

3. Basic Polynomial-Time Tools

In this section, we recall some elementary problems that are solvable in polynomial time and that we need to reference later. The techniques will be quite familiar to most readers. Still, it is worth reviewing a few of these not only to emphasize polynomial-time thinking, but also to distinguish these problems from those in Sections 4 and 5, for which presently implemented methods do not meet our measure of efficiency.

We assume that $G = \langle X \rangle \leq \mathrm{Sym}(\Omega)$.

Some efficient procedures follow a divide-and-conquer approach that exploits the orbit and imprimitivity structure of the group. We observe that standard computations of orbits and imprimitivity blocks run in polynomial time.

(3.1) *Given $\omega \in \Omega$, find ω^G, the orbit of ω under G, and for each $\psi \in \omega^G$, find $g \in G$ such that $\omega^g = \psi$.*

A naive transitive closure algorithm involves, at worst, applying each generator to each element of Ω, for a worst-case time of $O(|X|n)$. □

(3.2) *If G acts transitively on Ω, test whether G acts primitively and, if not, find a non-trivial block system.*

For example, the (unique) smallest block containing any given $\alpha, \beta \in \Omega$ is the component of α in the undirected graph $(\Omega, \{\alpha, \beta\}^G)$. □

We often concatenate polynomial-time procedures, up to a polynomial number in fact, to obtain, thereby, a polynomial-time procedure. To do so freely, however, we must be sure that the size of the output of each procedure is bounded by some fixed polynomial in the size of the permutation domain. (Technically, the output of a quadratic procedure could use space that is quadratic in its input size; the concatenation of an unbounded number of such procedures is prohibitive.) In particular, since many procedures involve finding generators for some targeted subgroup, we need to exercise some control on the sizes of generating sets. For example, there is an polynomial-time procedure for

(3.3) *Find a set of $< n^2$ generators for G.*

The underlying logic for this is at the heart of Sims's methods [29]. We review the idea, from which polynomial time is then a straightforward observation.

```
for $i = 1$ to $n - 1$ do
  while $\exists$ distinct $x, y \in X \cap G^{(i)}$ such that $\omega_i^x = \omega_i^y$ do
    replace such a pair $x, y$ by $x, yx^{-1}$.
```

Discarding duplicates, the modified X does not contain distinct elements of any $G^{(i)}$ that lie in the same right coset of $G^{(i+1)}$. Hence, the final X has size at most $\sum_{i=1}^{n-1} |G^{(i)} \colon G^{(i+1)}| \leq \sum_{i=1}^{n-1} (n - i + 1) < n^2$. □

Remark. Henceforth, we assume that all polynomial-time algorithms that output generators for a subgroup return fewer than n^2 generators. We also assume that $|X| < n^2$.

It is often the case that a subgroup H of our given G is specified only by some testable condition, i.e., there is a polynomial-time procedure which, for any $g \in G$, determines whether $g \in H$. (The subgroups $G^{(i)}$ serve as examples.) In such case, we say that H is *(polynomial-time) recognizable.*

(3.4) *Find generators for a polynomial-time-recognizable subgroup H for which $|G \colon H| = O(n^c)$ and determine $|G \colon H|$.*

A right transversal T for H in G and a set Y of Schreier generators [14] of H are both constructed in:

```
$T := \{1\}$; $Y := \emptyset$; $Q := \{1\}$
while $Q \neq \emptyset$ do
    remove $q$ from $Q$;
    for all $x \in X$ do
        if $\exists t \in T : qxt^{-1} \in H$ then add such $qxt^{-1}$ to $Y$
        else add $qx$ to $T$ and to $Q$.
```

Thus, T has been constructed by a transitive closure method, having right multiplied every element of T by all generators of G to see if this produces any new cosets. In the end, $TX \subseteq \langle Y\rangle T$, so that $G = \langle Y\rangle T$, whence $\langle Y\rangle = H$.

The running time is $O(|G\colon H|^2 n^{c'+c''})$, where $O(n^{c'})$ is the time for a membership-test in H and c'' is an absolute constant, so that an upper bound is $O(n^{2c+c'+c''})$. □

Remarks. It is often the case that H-recognizability also involves some natural interpretation of the cosets, obviating the search through all $t \in T$ to find whether some $qxt^{-1} \in H$; this could eliminate as much as $c+c'$ from the exponent in the timing. The prototypical example is $H = G^{(2)}$, wherein T is keyed by the orbit of ω_1.

It is assumed that the bound $|G\colon H| = O(n^c)$, for some constant c, is known, though the method can also be interpreted as testing this bound in polynomial time; if the procedure takes longer than the predicted number of steps, the bound does not hold.

Applying (3.4) iteratively yields a polynomial-time solution to

(3.5) *Find generators for a subgroup H given that $H = H_m \leq H_{m-1} \leq \cdots \leq H_0 = G$ where the H_i are each polynomial-time-recognizable and $|H_i\colon H_{i+1}| = O(n^c)$ for $0 \leq i < m$.* □

In particular, as each $G^{(i)}$ is polynomial-time recognizable, we can solve in polynomial time

(3.6) *Given any $\Delta \subset \Omega$, find generators for G_Δ (the pointwise stabilizer of Δ).* □

Since $|G| = \prod_{i=1}^{n-1} |G^{(i)}\colon G^{(i+1)}|$, we can, in polynomial time,

(3.7) *Find $|G|$.* □

Noting that $x \in G$ iff $|G| = |\langle G, x\rangle|$, we have a polynomial-time algorithm for membership-testing:

(3.8) *Given $x \in \mathrm{Sym}(\Omega)$, test whether $x \in G$.* □

Remarks. We should emphasize that we are not recommending this indirect approach to membership-testing in practice. We are only reminding the reader of a particular logical interconnection of these problems, through which polynomial time is made clear.

The observation that Sims's methods run in polynomial time was made by Furst, Hopcroft and Luks [11].

More generally we refer later to a polynomial-time procedure for

(3.9) *Given $\psi_1, \dots, \psi_m \in \Omega$, with $m \leq n = |\Omega|$, test whether $\exists x \in G$ such that $\omega_i^x = \psi_i$, for $1 \leq i \leq m$, and, if so, find (the subcoset of) all such x.*

By (3.1), we can find $y \in G$ such that $\omega_1^y = \psi_1$, if any such y exists. Recursively find the subcoset Hz of G_{ω_1} mapping $\omega_i \mapsto \psi_i^{y^{-1}}$, for $2 \leq i \leq m$; return Hzy. (Note that the single recursive call involves a permutation group on at most $n-1$ letters.) □

As an immediate consequence of (3.8), we have a polynomial-time algorithm for

(3.10) *Given $H = \langle Y \rangle \leq \mathrm{Sym}(\Omega)$, test whether $H \leq G$.*

I.e., test $Y \subseteq G$. □

Several applications require normal closures.

(3.11) *Given $H = \langle Y \rangle \leq \mathrm{Sym}(\Omega)$, find H^G.*

To get generators, $\bar{Y}$ of H^G: initialize $\bar{Y} = Y$; while there exist $x \in X, y \in \bar{Y}$ such that $y^x \notin \langle \bar{Y} \rangle$, add such y^x to $\bar{Y}$. □

Remark. We remind the reader that we use (from now on implicitly) the polynomial constraint on the length of any increasing chain of subgroups of $\mathrm{Sym}(\Omega)$.

As the derived group G' is the normal closure in G of $\langle [x,y] \mid x, y \in X \rangle$, we have a polynomial-time procedure for

(3.12) *Find the derived series $G \geq G' \geq (G')' \geq \cdots$. Hence, test whether G is solvable.* □

To compute the lower central series of G, let $L_1(G) = G = \langle X \rangle$; then if $L_i = \langle X_i \rangle$, $L_{i+1}(G) = \langle [x,y] \mid x, \in X, y \in X_i \rangle^G$. Thus we have a polynomial-time procedure for

(3.13) *Find the lower central series of G. Hence, test whether G is nilpotent.* □

An alternative polynomial-time nilpotence test is to check that G is a direct product of p-groups: For each $x \in X$ and each prime p dividing $|G|$, let $\langle x_p \rangle$ be the Sylow p-subgroup of $\langle x \rangle$; for example, $x_p = x^m$ where $|G| = mp^k$ with $(m,p) = 1$ (such large powers of x are computable by "repeated squaring" tricks, though an even faster approach could be to compute the power in each cycle of x, first reducing m modulo the cycle length). Letting $G_p = \langle x_p \mid x \in X \rangle$, verify that $|G_p|$ is a power of p for each p and that the generators of G_p commute with the generators of G_q for $p \neq q$. □

4. Not Known to be in Polynomial Time

There is general agreement that, by all measures, the problems in the preceding section have efficient solutions. We turn now to some which seem to have satisfactory implementations but for which all known algorithms have exponential worst-case complexity. Were there no other reason for looking at them, these would be of theoretical interest because of their relation to the graph-isomorphism problem:

PROBLEM. GRAPH ISOMORPHISM (GRAPH-ISO)
INPUT: *Graphs* $\mathcal{G}_1 = (V_1, E_1)$, $\mathcal{G}_2 = (V_2, E_2)$.
QUESTION: *Are* $\mathcal{G}_1, \mathcal{G}_2$ *isomorphic?*

It is generally felt that GRAPH-ISO is not a hard problem in practice (see, e.g., [23] for an implemented procedure that many have found satisfactory). Nevertheless, although the problem has been extensively studied, nothing close to polynomial time has been proved. Indeed, there is no known approach that has proved to be *subexponential* (say, for example, in $O(n^{\log^c n})$ time) in the worst case. (See remarks at the end of Section 6). On the other hand, there is evidence that GRAPH-ISO is not NP-complete, otherwise there would be a collapse of the "polynomial-time hierarchy" [13]. Indeed, from the earliest expositions of NP-completeness (e.g., see discussion in [12]), there has been speculation that GRAPH-ISO may be one of the few classical decision problems that is neither in polynomial time nor NP-complete.

We recall polynomial-time reductions of GRAPH-ISO to permutation-group problems. To introduce the groups, we consider

PROBLEM. GRAPH AUTOMORPHISM-GROUP (GRAPH-AUTO)
INPUT: *Graph* $\mathcal{G} = (V, E)$.
FIND: *Generators for* $\mathrm{Aut}(\mathcal{G})$, *the automorphism group of* $\mathcal{G}$.

The following is well known.

PROPOSITION 4.1. *GRAPH-ISO and GRAPH-AUTO are polynomial-time equivalent problems.*

PROOF. To reduce GRAPH-ISO to GRAPH-AUTO, we first note that it suffices to consider the GRAPH-ISO case where the graphs $\mathcal{G}_1, \mathcal{G}_2$ are connected, for, in general, one may test all pairs of connected components. Given connected $\mathcal{G}_1 = (V_1, E_1), \mathcal{G}_2 = (V_2, E_2)$, form the disjoint union $\mathcal{G} = (V_1 \dot\cup V_2, E_1 \dot\cup E_2)$ and suppose $\mathrm{Aut}(\mathcal{G}) = \langle X \rangle$. Then $\mathcal{G}_1 \cong \mathcal{G}_2$ iff $\exists x \in X : V_1^x = V_2$.

We turn to the reverse reduction. For this, we first observe that GRAPH-ISO would enable us to solve

PROBLEM. RESTRICTED GRAPH AUTOMORPHISM
(RES-GRAPH-AUTO)
INPUT: *Graph* $\mathcal{G} = (V, E)$ *and, for some* $m \leq |V|$*;*
sequences $v_1, v_2, \ldots, v_m$ *and* $w_1, w_2, \ldots, w_m$ *of distinct vertices in* V.
QUESTION: *Is there some* $g \in \mathrm{Aut}(\mathcal{G})$ *such that* $v_i^g = w_i$*, for* $1 \leq i \leq m$*?*

Reducing RES-GRAPH-AUTO to GRAPH-ISO: Attaching distinguishable "gadgets" to the v_i forming a graph $\mathcal{G}_1$ and similar gadgets to the respective w_i forming a graph $\mathcal{G}_2$, RES-GRAPH-ISO reduces to testing isomorphism of the modified $\mathcal{G}_1$, $\mathcal{G}_2$. A suitable gadget at v_i, respectively w_i, could be a new cycle of length $|V| + i$ through the vertex.

Reducing GRAPH-AUTO to RES-GRAPH-AUTO: Repeated application of a procedure for the decision problem RES-GRAPH-AUTO facilitates the actual construction of a suitable g; for, having received a "yes", we go on to find a possible v_{m+1}^g (using a RES-GRAPH-AUTO procedure to test all candidates) then v_{m+2}^g, etc. Note this will have used $O(|V|^2)$ calls to RES-GRAPH-AUTO.

In this fashion, $O(|V|^3)$ applications of RES-GRAPH-AUTO determine the orbit of v_1 under $\mathrm{Aut}(\mathcal{G})$ and a right transversal for $\mathrm{Aut}(\mathcal{G})^{(2)}$ (the stabilizer of v_1) in $\mathrm{Aut}(\mathcal{G})$. Similarly, we get transversals for each $\mathrm{Aut}(\mathcal{G})^{(i+1)}$ in $\mathrm{Aut}(\mathcal{G})^{(i)}$. The union of such transversals generate $\mathrm{Aut}(\mathcal{G})$. Thus, GRAPH-AUTO has been recovered from $O(|V|^4)$ applications of GRAPH-ISO (to graphs of polynomial size $O(|V|^2)$). □

Observing then that GRAPH-AUTO is the problem that we would have to solve, we consider the natural action of $G = \mathrm{Sym}(V)$ on the set of unordered pairs in V, and see that $\mathrm{Aut}(\mathcal{G})$ is precisely the subgroup that stabilizes E. With this in mind, we define the problem

PROBLEM. SET-STABILIZER (STAB)
INPUT: $G \leq \mathrm{Sym}(\Omega); \Delta \subseteq \Omega$.
FIND: $\mathrm{Stab}_G(\Delta) = \{g \in G \mid \Delta^g = \Delta\}$.

Thus, the above argument showed

PROPOSITION 4.2. *GRAPH-ISO is polynomial-time reducible to STAB.* □

There are two other important formulations of STAB. Consider

PROBLEM. INTERSECTION (INTER)
INPUT: $G, H \leq \mathrm{Sym}(\Omega)$.
FIND: $G \cap H$.

PROBLEM. CENTRALIZER (CENT)
INPUT: $G \leq \mathrm{Sym}(\Omega)$*;* $x \in \mathrm{Sym}(\Omega)$.
FIND: *The centralizer,* $\mathrm{C}_G(x)$*, of* x *in* G.

PROPOSITION 4.3. *The problems STAB, INTER and CENT are polynomial-time equivalent.*

PROOF. Suppose we are given an instance (G, Ω, Δ) of STAB. To reduce this to INTER or CENT, let G act in the diagonal on the disjoint union $\widehat{\Omega} = \Omega_1 \dot{\cup} \Omega_2$ of two copies of Ω (i.e., $(\omega_i)^g = \omega_i^g$, $\forall \omega \in \Omega$, $i = 1, 2$, $\forall g \in G$, where ω_i denotes the Ω_i copy of $\omega \in \Omega$). Let x be the involution in $\mathrm{Sym}(\widehat{\Omega})$ specified by: $(\omega_i)^x = \omega_i$ if $\omega \notin \Delta$ and $(\omega_i)^x = \omega_{3-i}$ if $\omega \in \Delta$. Then $\mathrm{Stab}_G(\Delta) = G \cap G^x = \mathrm{C}_G(x)$.

We indicate reductions in the other direction. For INTERS: let $G \times H$ act on $\Omega \times \Omega$ in the natural way, and set $\Delta = \{(\omega, \omega) \mid \omega \in \Omega\}$; then $\mathrm{Stab}_{G \times H}(\Delta) = \{(g, g) | g \in G \cap H\}$. FOR CENT: $g \in G$ commutes with x iff g, acting diagonally on $\Omega \times \Omega$, stabilizes $\{(\omega, \omega^x) \mid \omega \in \Omega\}$. □

Remarks. In a panel discussion at the DIMACS workshop that gave rise to these Proceedings, the sentiment was generally expressed that STAB, INTER and CENT are "not hard in practice." Thus one should ask:

QUESTION 1. *Are STAB (or INTER or CENT) in polynomial time? Are there even subexponential methods?*

Of course, affirmative answers would carry over to GRAPH-ISO. Until such time as this is resolved, implemented methods that rely on *general* procedures for STAB, INTER or CENT cannot be proved efficient. In particular, they must be excluded from the polynomial-time toolkit.

We emphasize "general" in the last paragraph, for it is entirely plausible that the problems can often be solved efficiently. The challenge that we put forth, therefore, is to back this up with theory.

QUESTION 2. *For what classes of inputs do implemented procedures, or modifications thereof, for STAB, INTER or CENT, have polynomial (or subexponential) worst case performance?*

If the question seems vague, we welcome reformulation, even to the exclusion of polynomial time as a targeted criterion. Assuming there is an acknowledged class of "interesting" groups, what can you *guarantee* about the running time over that class? A proffered system should be able to promise efficiency beyond the observation that a procedure took S seconds for group G on machine M.

At first glance, the sentiment that STAB, etc., are "not hard in practice," seems entirely consistent with feelings about GRAPH-ISO. However, in the latter case, one can provide some theoretical justification, since there are well-defined and accepted notions of *random graphs*, with respect to which naive isomorphism-testing procedures are provably fast on average (e.g., [5]). Can one do the same for groups?

QUESTION 3. *What is the* average *running time of implemented procedures for STAB, etc.?*

We leave open the choice of probability distributions from which to approach this problem. A uniform distribution over all permutation groups is just one possibility. One could also look at conjugacy classes (in Sym(Ω)) or even isomorphism classes. One should, as well, look at this problem for restricted classes of groups.

In Sections 6 and 7, we do give examples of procedures for STAB, INTER, CENT that are provably in polynomial time for specified input classes, including, for example, solvable groups.

The similarity of these problems to GRAPH-ISO carries over to analogues of the GRAPH-AUTO/GRAPH-ISO relationship. Namely, there is, in each case, an equivalent decision problem in NP. Essentially, these may be obtained by substituting cosets for (one or both of) the groups in the problem and asking whether the targeted set is nonempty. E.g., corresponding to INTER, we ask whether $Gx \cap H \neq \emptyset$. Then, with minor reformulations, we obtain the following problems corresponding, respectively, to STAB, INTER, CENT.

Problem. SET-TRANPORTER (TRANS)
Input: $G \leq \mathrm{Sym}(\Omega)$; $\Delta_1, \Delta_2 \subseteq \Omega$.
Question: *Is there some* $g \in G$ *such that* $\Delta_1^g = \Delta_2$*?*

Problem. DOUBLE-COSET EQUALITY (DC-EQ)
Input: $G, H \leq \mathrm{Sym}(\Omega)$; $x_1, x_2 \in \mathrm{Sym}(\Omega)$.
Question: *Does* $Gx_1H = Gx_2H$*?*

Problem. CONJUGACY OF ELEMENTS (CONJ-ELT)
Input: $G \leq \mathrm{Sym}(\Omega)$; $x_1, x_2 \in \mathrm{Sym}(\Omega)$.
Question: *Is there some* $g \in G$ *such that* $x_1^g = x_2$*?*

Note, we include "ELEMENTS" in the title specifically to distinguish from the corresponding question of conjugacy of groups (see Section 10.2).

We have the following analogue of Proposition 4.1.

Proposition 4.4. *STAB is equivalent to each of the problems TRANS, DC-EQ and CONJ-ELT.*

Proof. We outline this for the equivalence STAB $\equiv$ TRANS. (One can also get CENT $\equiv$ CONJ-ELT and INTER $\equiv$ DC-EQ by establishing TRANS $\equiv$ CONJ-ELT $\equiv$ DC-EQ the way one established STAB $\equiv$ CENT $\equiv$ INTER. Note that DC-EQ is trivially restated as testing non-emptiness of an intersection of a group $H^{x_1^{-1}}$ and a coset $Gx_2x_1^{-1}$.)

Reducing TRANS to STAB: Given an instance (G, Δ_1, Δ_2) of TRANS, consider the wreath product $\hat{G} = G \wr Z_2$ acting on the disjoint union $\Omega\dot{\cup}\Omega$ of two copies of Ω and let $\hat{\Delta} = \Delta_1\dot{\cup}\Delta_2$, in which Δ_i is considered as lying in the ith copy of Ω. Then the answer to TRANS is affirmative iff some generator of $\mathrm{Stab}_{\hat{G}}(\hat{\Delta})$ switches Δ_1 and Δ_2.

For a reverse reduction, we consider the following analogue of RES-AUTO:

PROBLEM. RESTRICTED SET STABILIZER (RES-STAB)
INPUT: $G \leq \mathrm{Sym}(\Omega); \Delta \subseteq \Omega$; *sequences* $\omega_1, \dots, \omega_m$ *and* $\psi_1, \dots, \psi_m$ *in* Ω.
QUESTION: *Is there some* $g \in \mathrm{Stab}_G(\Delta)$ *such that* $\omega_i^g = \psi_i$ *for* $1 \leq i \leq m$.

Reducing RES-STAB to TRANS: Find, using (3.9), the subcoset Hy of G consisting of elements mapping $\omega_i \mapsto \psi_i$ for $1 \leq i \leq m$; apply TRANS to $(H, \Delta, \Delta^{y^{-1}})$.

Reducing STAB to RES-STAB: This proceeds exactly as the reduction of GRAPH-AUTO to RES-GRAPH-AUTO. □

Remark. Extending the analogy to GRAPH-ISO, Babai and Moran [8] have shown that TRANS (therefore DC-EQ and CONJ-ELT) could be NP-complete only if the polynomial-time hierarchy collapses to $\Sigma_2^p = \Pi_2^p$. Thus, even if GRAPH-ISO were to be resolved via other methods (there is a legion of sufferers from the "Graph-Isomorphism Disease", see [28] for traditional attacks) such group-theoretic problems would very possibly remain as outstanding candidates for membership in a complexity class strictly between P and NP-Complete. There is also the possibility of an affirmative answer to the following open question:

QUESTION 4. *Is DC-EQ (equivalently TRANS, CONJ-ELT) polynomial-time reducible to GRAPH-ISO?*

See Section 10 for additional open questions on where group-theoretic decision problems fit in this hierarchy.

5. Not Likely to be in Polynomial Time

A suggested approach to DC-EQ (e.g., [9, 19]) has been to determine, in any given double-coset GxH, its lexicographically least element, as $Gx_1H = Gx_2H$ iff the lex-least element in Gx_1H is the lex-least element in Gx_2H. This is analogous to, and a generalization of, attacking GRAPH-ISO by establishing lexicographically least representations (e.g. lex-least adjacency matrices) as canonical forms.

Of course, proponents realize that the approach has limitations. It is nevertheless worthwhile to provide theoretical substantiation of its difficulty, namely, that it involves an NP-hard problem.

We suppose that Ω is linearly ordered with respect to a relation $<$. Then $\mathrm{Sym}(\Omega)$ acquires a lexicographic ordering $\prec$, specifically, if $x, y \in \mathrm{Sym}(\Omega)$ and $x \neq y$, then $x \prec y$ iff $\omega^x < \omega^y$ for the least $\omega \in \Omega$ such that $\omega^x \neq \omega^y$.

We state a polynomial-time equivalent decision problem in order to bring the question into NP.

PROBLEM. LEXICOGRAPHIC LEADER in DOUBLE COSET (LLDC)
INPUT: *A linearly-ordered set* Ω; $G, H < \mathrm{Sym}(\Omega)$; $x, y \in \mathrm{Sym}(\Omega)$.
QUESTION: *Is there some* $z \in GxH$ *such that* $z \prec y$?

LLDC is in NP, for one can guess permutations g, h and verify that $g \in G$, $h \in H$ and $gxh \prec y$. Clearly, if one could find lex-least elements in polynomial time then LLDC would be solvable in polynomial time. Conversely, a polynomial number of calls to an LLDC procedure would suffice in a binary-search procedure for lex-least elements.

We show

THEOREM 5.1. *LLDC is NP-complete.*

In fact, we give two distinct reductions of known NP-complete problems to LLDC, as they involve different, yet seemingly reasonable, classes of groups and seem to display different reasons for the difficulty. The first shows that LLDC is "hard" even if G and H are symmetric groups (not, of course, in their natural actions). The second reduction shows that LLDC is "hard" even if G and H are abelian and even if the orbits of $\langle G, x, H\rangle$ are small (size 3). (However, see the remark following the proofs.)

FIRST PROOF OF THEOREM 5.1. It is known that the following problem is NP-complete (see, e.g., [12]).

PROBLEM. MAXIMAL CLIQUE (MAX-CLIQ)
INPUT: *Graph $\mathcal{G} = (V, E)$, integer K.*
QUESTION: *Does $\mathcal{G}$ contain a clique of cardinality K?*

(Recall that $W \subseteq V$ is called a *clique* in $\mathcal{G}$ if $\{w_1, w_2\} \in E$, for all $w_1, w_2 \in W$.)

We reduce MAX-CLIQ to LLDC, thereby establishing LLDC is also NP-complete:

Let $(\mathcal{G} = (V, E), K)$ be an instance of MAX-CLIQ. We may assume that V is linearly ordered so that $V = \{v_1, \dots, v_m\}$, the subscripts reflecting the ordering. Let Ω be the set of pairs $\{\{v_i, v_j\} \mid 1 \le i < j \le m\}$. Then Ω is linearly ordered with $\{v_i, v_j\} < \{v_k, v_l\}$, for $1 \le i < j \le m$, $1 \le k < l \le n$, if either $j < l$ or $j = l$ and $i < k$. For any q, $1 \le q \le \binom{m}{2}$, let Ω_q denote the set consisting of the first q elements of Ω in this ordering.

Note that $E \subseteq \Omega$. We may assume that $|E| \ge \binom{K}{2}$. Let x be any permutation in $\mathrm{Sym}(\Omega)$ that maps $\Omega_{|E|}$ to E. Let H be the natural image of $\mathrm{Sym}(V)$ in $\mathrm{Sym}(\Omega)$ and let $G = \mathrm{Sym}(\Omega)_{\Omega \setminus \Omega_{|E|}}$ (so $G \simeq \mathrm{Sym}(\Omega_{|E|})$).

Let $y \in \mathrm{Sym}(\Omega)$ be the transposition switching the elements in positions $\binom{K}{2}$ and $\binom{K}{2} + 1$. Then, for $z \in \mathrm{Sym}(\Omega)$, $z \prec y$ iff z pointwise fixes $\Omega_{\binom{K}{2}}$ We claim there is a clique of size K in $\mathcal{G}$ iff $\exists z \in GxH$ such that $z \prec y$. This follows from the fact that there is a clique in $\mathcal{G}$ of cardinality K iff there exists $h \in H (= \mathrm{Sym}(V))$ such that $(\Omega_{\binom{K}{2}})^h \subseteq E$ (the clique then being $\{v_1^h, \dots, v_K^h\}$). But, as the permutations in Gx map $\Omega_{\binom{K}{2}}$ precisely to the subsets of cardinality $\binom{K}{2}$ in the set E, there is a clique of size K iff some permutation in H agrees with some permutation in Gx on $\Omega_{\binom{K}{2}}$, which is true iff some $z \in GxH$ pointwise fixes $\Omega_{\binom{K}{2}}$, i.e., iff $z \prec y$. □

SECOND PROOF OF THEOREM 5.1. The following is also NP-complete (see, e.g., [12]).

PROBLEM. EXACT 3-COVER (X3C)

INPUT: *A set Σ together with a collection Θ of size-3 subsets of Θ.*

QUESTION: *Is there a subcollection $\Theta' \subseteq \Theta$ with $|\Theta'| = |\Sigma|/3$ such that $\Sigma = \bigcup_{\theta \in \Theta'} \theta$.*

Reduction of X3C to LLDC:

Given an instance (Σ, Θ) of X3C, we construct an instance (Ω, G, H, x, y) of LLDC as follows.

We may assume the triples in Θ are distinct. Let $\Psi = \{\{\theta, \theta'\} \mid \theta, \theta' \in \Theta, \theta \cap \theta' \neq \emptyset\}$, the collection of unordered-pairs of intersecting triples. Let $\Phi = \Sigma \cup \Theta \cup \Psi$. The desired permutation domain is

$$\Omega = \Phi \times \{1, 2, 3\}$$

and we fix any linear ordering of Ω subject only to the condition that $\Phi \times \{1\}$ precedes $\Phi \times \{2\}$ and $\Phi \times \{2\}$ precedes $\Phi \times \{3\}$.

For any $\phi \in \Phi$, we let $a_\phi \in \mathrm{Sym}(\Omega)$ be the 3-cycle $((\phi,1),(\phi,2),(\phi,3))$ (i.e., $(\phi,1) \to (\phi,2) \to (\phi,3) \to (\phi,1)$), and let $b_\phi \in \mathrm{Sym}(\Omega)$ be the transposition $((\phi,1),(\phi,2))$. (Note that $\langle\{a_\phi \mid \phi \in \Phi\}\rangle$ is an elementary abelian 3-group and $\langle\{b_\phi \mid \phi \in \Phi\}\rangle$ is an elementary abelian 2-group.) For $\theta = \{\sigma_1, \sigma_2, \sigma_3\} \in \Theta$, define $c_\theta \in \mathrm{Sym}(\Omega)$ by

$$c_\theta = a_{\sigma_1} a_{\sigma_2} a_{\sigma_3} a_\theta \cdot \prod_{\theta \in \psi \in \Psi} a_\psi \, .$$

Now let

$$\begin{aligned} G &= \langle\{c_\theta \mid \theta \in \Theta\}\rangle; \\ H &= \langle\{b_\phi \mid \phi \in \Theta \cup \Psi\}\rangle; \\ x &= \prod_{\sigma \in \Sigma} a_\sigma^{-1}. \end{aligned}$$

Further, let $y \in \mathrm{Sym}(\Omega)$ be the transposition switching the last point in $\Phi \times \{1\}$ with the first point in $\Phi \times \{2\}$. Then, for $z \in \mathrm{Sym}(\Omega)$, $z \prec y$ iff z fixes $\Phi \times \{1\}$ pointwise.

To establish the reduction we show that LLDC with input (Ω, G, H, x, y) has an affirmative answer iff X3C with input (Σ, Θ) has an affirmative answer.

First suppose that $\Theta' \subseteq \Theta$ is an exact cover of Σ. Let

$$\begin{aligned} g &= \prod_{\theta \in \Theta'} c_\theta \ \in G \\ h &= \prod_{\theta \in \Theta'} (b_\theta \cdot \prod_{\theta \in \psi \in \Psi} b_\psi) \ \in H. \end{aligned}$$

We claim that gxh pointwise fixes $\Phi \times \{1\}$ (whence $gxh \prec y$). To see this: for $\sigma \in \Sigma$, σ is in θ for exactly one $\theta \in \Theta'$ so that $(\sigma,1)^{gxh} = (\sigma,2)^{xh} = (\sigma,1)^h = (\sigma,1)$; for $\theta \in \Theta \setminus \Theta'$, $(\theta,1)$ is fixed by each of g,x,h; for $\theta \in \Theta'$, $(\theta,1)^{gxh} = (\theta,2)^{xh} = (\theta,2)^h = (\theta,1)$; finally, for $\psi \in \Psi$, if $\psi \cap \Theta' = \emptyset$ then $(\psi,1)$ is fixed by each of g,x,h, otherwise $\theta \in \psi$ for precisely one $\theta \in \Theta'$ so that $(\psi,1)^{gxh} = (\psi,2)^{xh} = (\psi,2)^h = (\psi,1)$.

Conversely, suppose that $gxh \prec y$, for $g \in G, h \in H$, so that gxh pointwise fixes Φ_1. We can express

$$\begin{aligned} g &= \prod_{\theta\in\Theta} c_\theta^{e_\theta}, \text{ where } e_\theta = 0,1, \text{ or } 2; \\ h &= \prod_{\theta\in\Theta'} b_\theta \cdot \prod_{\psi\in\Psi'} b_\psi, \text{ where } \Theta' \subseteq \Theta, \Psi' \subseteq \Psi. \end{aligned}$$

We show that Θ' is an exact cover of Σ: For $\theta \in \Theta$, $(\theta,1)^{gxh} = (\theta,1)$ implies $e_\theta = 0$ if $\theta \notin \Theta'$ and $e_\theta = 1$ if $\theta \in \Theta'$. Hence, $g = \prod_{\theta\in\Theta'} c_\theta$. For $\sigma \in \Sigma$, $(\sigma,1) = (\sigma,1)^{gxh} = (\sigma,1)^{ga_\sigma^{-1}}$, and so $\sigma \in \bigcup_{\theta\in\Theta'} \theta$. Finally, we must show that Θ' does not contain θ,θ' with $\theta \cap \theta' \neq \emptyset$. Suppose, to the contrary, that such $\theta,\theta' \in \Theta'$ and let $\psi = \{\theta,\theta'\} \in \Psi$; then $(\psi,1)^{gxh} = (\psi,1)^{c_\theta c_{\theta'} h} = (\psi,3)^h = (\psi,3)$, contradicting the fact that gxh fixes $(\psi,1)$. □

Remarks. The construction in the second proof should be compared with the result of Theorem 6.2, where it is shown that, with a judicious choice of ordering of Ω (determined by G alone), the problem is actually in polynomial time for interesting classes of groups G. This includes all solvable groups as well as all groups with bounded orbits. Either of these conditions are satisfied by the groups of the above reduction, in fact for $\langle G,x,H\rangle$.

On the other hand, we conjecture that there is no analogous fix for the situation encountered in the first proof. Therein Ω is the set of size-2 subsets of a linearly ordered set V, H the natural image of Sym(V) in Sym(Ω) and G is the subgroup of Sym(Ω) fixing all but the first q points for some $q \le \binom{|V|}{2}$. In this special setting, we ask

QUESTION 5. *Given such Ω, G, H, is there a reordering of Ω with respect to which the lexicographically least elements in GxH, for any $x \in$ Sym(Ω), can be found in polynomial time?*

With such an ordering in hand, one could define a polynomial-time computable *canonical form* for graphs $\mathcal{G} = (V,E)$ with $|E| = q$: take any $x \in$ Sym(Ω) such that $\Omega_q^x = E$ (Ω_q remains the first q elements in the original ordering); find z, the lex-least element (with respect to the new ordering) in GxH; set CF($\mathcal{G}$) $= (V, \Omega_q^z)$. The graph CF($\mathcal{G}$) is independent of the choice of x; even more significantly, it is a complete isomorphism invariant (hence a canonical form), that is, $\mathcal{G} = (V,E)$ is isomorphic to $\mathcal{G}' = (V,E')$ iff CF($\mathcal{G}$) $=$ CF($\mathcal{G}'$).

As we do not expect an affirmative answer to Question 5, we suggest looking for evidence to the contrary. Could one show, for example, that with such G and H, LLDC remains NP-complete for *any* prescribed ordering of Ω?

An important restricted case of LLDC occurs when $G = H$ (e.g., [19]). Thus, it is worth observing that this case is equally "hard".

COROLLARY 5.2. *LLDC remains NP-complete when $G = H$.*

PROOF. We describe a polynomial-time reduction of the general LLDC problem to this special case.

Suppose (Ω, G, H, x, y) is an instance of LLDC.

Set $\hat{\Omega} = \Omega \times \{1, 2\}$ and linearly order $\hat{\Omega}$ so that $(\omega, i) \prec (\psi, j)$ if $i < j$ or if $i = j$ and $\omega \prec \psi$. Let $\hat{G} = G \times H$ act on $\hat{\Omega}$ via $(\omega, 1)^{(g,h)} = (\omega^g, 1)$, $(\omega, 2)^{(g,h)} = (\omega^h, 2)$ for $\omega \in \Omega, g \in G, h \in H$ (thus G acts naturally on $\Omega \times \{1\}$ and H acts naturally on $\Omega \times \{2\}$). Let $\hat{x}, \hat{y} \in \mathrm{Sym}(\hat{\Omega})$ satisfy $(\omega, 1)^{\hat{x}} = (\omega^x, 2)$, $(\omega, 1)^{\hat{y}} = (\omega^y, 2)$ and $(\omega, 2)^{\hat{x}} = (\omega, 2)^{\hat{y}} = (\omega, 2)$ for $\omega \in \Omega$.

To establish the reduction, we show that the instance (Ω, G, H, x, y) of LLDC has an affirmative answer iff the instance $(\hat{\Omega}, \hat{G}, \hat{G}, \hat{x}, \hat{y})$ has an affirmative answer.

Suppose $\exists g \in G, h \in H$ satisfying $gxh \prec y$. Then $(g, 1)\hat{x}(1, h) \prec \hat{y}$. This follows from that the fact that, for $\omega \in \Omega$, $(\omega, 1)^{(g,1)\hat{x}(1,h)} = (\omega^{gxh}, 2)$.

Conversely, suppose $\exists (g_1, h_1), (g_2, h_2) \in \hat{G}$ satisfying $(g_1, h_1)\hat{x}(g_2, h_2) \prec \hat{y}$. Then $g_1 x h_2 \prec y$. To see this: the first point in $\hat{\Omega}$ at which $(g_1, h_1)\hat{x}(g_2, h_2)$ and $\hat{y}$ differ must lie in $\Omega \times \{1\}$ for no permutation agrees with $\hat{y}$ on $\Omega \times \{1\}$ and strictly precedes it; the result now follows from the observation $(\omega, 1)^{(g_1,h_1)\hat{x}(g_2,h_2)} = (\omega^{g_1 x h_2}, 2)$. □

Remark. A context in which LLDC arises is the actual enumeration of all double cosets GxH of subgroups G, H of some $L \in \mathrm{Sym}(\Omega)$ (see, e.g., [9]). (For, assuming LLDC is not significantly harder than DC-EQ for the particular groups, it would be most efficient to compute and store the lex-least elements as canonical representatives of their double cosets.) Since the answer is not necessarily of polynomial size, the natural question to ask in this case is how much work has to be done beyond that which is dictated by the output.

QUESTION 6. *Given $G, H, L \in \mathrm{Sym}(\Omega)$ with $G, H \leq L$, can the double cosets, GxH, for $x \in L$, be enumerated in time $O((\nu + n)^c)$, where ν is the number of such double cosets?*

The proof of Theorem 5.1 contains the ingredients of another NP-completeness result. Consider the class of problems

PROBLEM. MEMBERSHIP IN PRODUCT OF m GROUPS (m-MEMB)
INPUT: $A_1, A_2, \ldots, A_m \leq \mathrm{Sym}(\Omega)$; $x \in \mathrm{Sym}(\Omega)$.
QUESTION: *Is $x \in A_1 A_2 \ldots A_m$?*

Of course, 1-MEMB is in polynomial time by (3.8). The problem 2-MEMB is simply a restatement of DC-EQ, which is of unknown complexity.

What happens for $m \geq 3$? Clearly, m-MEMB is in NP, for one can guess and verify a factorization $x = a_1 \cdots a_m$ with $a_i \in A_i$. Now, in the reduction in each proof of Theorem 5.1, the element y was taken to be a transposition of two adjacent elements ω_{i-1}, ω_i in the linearly ordered Ω. Then, for $z \in \mathrm{Sym}(\Omega)$, $z \prec y$ iff $z \in \mathrm{Sym}(\Omega)^{(i)}$. Hence, there exists $z \in GxH$ preceding y iff $GxH \cap \mathrm{Sym}(\Omega)^{(i+1)} \neq \emptyset$, which occurs iff $x \in G\mathrm{Sym}(\Omega)^{(i+1)}H$. It follows that

PROPOSITION 5.3. *m-MEMB is NP-complete for $m \geq 3$.* □

Remark. Variations on these reductions can be used to show that even the special case of testing membership in GHG, for $G, H \leq \mathrm{Sym}(\Omega)$, is NP-complete. Furthermore, this remains NP-complete even when G and H are both abelian.

6. A Polynomial-Time Instance of LLDC

Following techniques introduced in [20] and [6], we show that the difficulty of the LLDC instance used in the second proof of Theorem 5.1 is attributable entirely to the particular linear ordering of Ω. That is, for a more general choice of G, one can define (in polynomial time) a linear ordering of Ω so that the lexicographically-least element in any GxH is obtainable in polynomial time. This will, in turn, yield polynomial-time solutions to special cases of INTER, STAB and CENT.

The restriction to be placed, on G alone, is a limit on the sizes of the noncyclic composition factors. Specifically, for any fixed integer d, let Γ_d denote the class of groups all of whose non-cyclic composition factors are isomorphic to subgroups of S_d. So, in particular, Γ_d includes all solvable groups. The following is proved in [2].

LEMMA 6.1 (BABAI, CAMERON, PÁLFY). *There is a constant c such that if G is a primitive subgroup of* $\mathrm{Sym}(\Psi)$ *and $G \in \Gamma_d$ then $|G| = O(|\Psi|^{cd})$.*

Remark. Since many problems on permutation groups have natural reductions to the primitive case, results that bound the size of primitive groups under various conditions are often essential to the analysis (see, e.g., [7] for other examples). Indeed, the investigations leading to Lemma 6.1 were inspired by computational complexity applications. In particular, the lemma enables a simplification as well as a wider applicability of the set-stabilizer algorithm in [20].

In this section, the lemma comes into play in the base case of a "divide-and-conquer" algorithm that is guided by the orbit/imprimitivity structure of G. For convenience, we keep track of this in a *structure forest* $\mathcal{F}$ for G. Such a forest includes one *structure tree* for each orbit. The (rooted) tree $\mathcal{T}_\Delta$ on the orbit Δ has leaf set Δ and is such that the action of G on Δ can be lifted to automorphisms of $\mathcal{T}_\Delta$, with the further property that the subgroup of G that fixes any node acts primitively on the children of that node. The polynomial-time construction of a suitable $\mathcal{T}_\Delta$ is an easy consequence of (3.2): if G is primitive then simply attach all points to a root, else find any non-trivial block Δ_1 and recursively construct a structure tree for the action of G on $\{\Delta_1\}^G$ and a tree for the action of $\mathrm{Stab}_G(\Delta_1)$ on Δ_1, using G to copy the latter to the other blocks (in this case, $\mathrm{Stab}_G(\Delta_1)$ is the stabilizer of a single "point" in the action on $\{\Delta_1\}^G$).

We establish some additional notation that is convenient for a recursive exploitation of orbits and blocks. Let Ω be a fixed linearly-ordered set. For any $\Delta \subseteq \Omega$, $\mathrm{Sym}(\Omega)^\Delta$ acquires a lexicographic linear ordering (via $f_1 \prec f_2$ iff $f_1(\delta) < f_2(\delta)$ for the least $\delta \in \Delta$ such that $f_1(\delta) \neq f_2(\delta)$). Define $\mu : \mathrm{Sym}(\Omega) \times \mathrm{Sym}(\Omega) \rightarrow \mathrm{Sym}(\Omega)$ by $\mu(g,h) = g^{-1}h$ and let $\mathrm{pr}_i : \mathrm{Sym}(\Omega) \times \mathrm{Sym}(\Omega) \rightarrow \mathrm{Sym}(\Omega)$, for $i = 1,2$ be the projections onto the first and second factors, respectively. For $A \subseteq \mathrm{Sym}(\Omega) \times \mathrm{Sym}(\Omega)$, $\Delta \subseteq \Omega$, let $\ell\ell_\Delta(A)$ denote the lexicographically-least element in $\mu(A)^\Delta$; observe that the lex-least element of GxH is $\ell\ell_\Omega((1,x)G \times H)$. We also consider the elements that induce $\ell\ell_\Delta(A)$, namely $\mathrm{LL}_\Delta(A) = \{z \in A \mid \mu(z)^\Delta = \ell\ell_\Delta(A)\}$. We need the following facts.

FACT 1. *If $\Delta = \Delta_1 \dot{\cup} \Delta_2$ with the elements of Δ_1 strictly preceding all those of Δ_2 then*

$$\mathrm{LL}_\Delta(A) = \mathrm{LL}_{\Delta_2}(\mathrm{LL}_{\Delta_1}(A)).$$

FACT 2. *If $A = A_1 \cup A_2 \cup \cdots \cup A_m$ then*

$$\begin{aligned} \ell\ell_\Delta(A) &= \text{lex-least}\{\ell\ell_\Delta(A_i) \mid 1 \leq i \leq m\} \\ \mathrm{LL}_\Delta(A) &= \bigcup\{\mathrm{LL}_\Delta(A_i) \mid \ell\ell_\Delta(A_i) = \ell\ell_\Delta(A),\ 1 \leq i \leq m\} \end{aligned}$$

FACT 3. *If A is a left coset of M and Δ is invariant under $\mathrm{pr}_1(M)$, then $\mathrm{LL}_\Delta(A)$ is a left coset of $\mathrm{Stab}_M(\{(\delta, \delta^{\ell\ell_\Delta(A)}) \mid \delta \in \Delta\})$.*

The proofs of Facts 1 and 2 are straightforward. Fact 3 follows from the observation that, for $u, v \in \mathrm{Sym}(\Omega) \times \mathrm{Sym}(\Omega)$, if $\mathrm{pr}_1(u^{-1}v)$ stabilizes Δ then $\mu(u)^\Delta = \mu(v)^\Delta$ iff $u^{-1}v$ stabilizes $\{(\delta, \delta^{\mu(u)}) \mid \delta \in \Delta\}$. □

These facts are used in the main theorem of this section:

THEOREM 6.2. *Let d be fixed. Given $G < \mathrm{Sym}(\Omega)$, with $G \in \Gamma_d$, in polynomial time one can establish a linear ordering of Ω with respect to which one can then find, in polynomial time, the lexicographical least element in GxH for any given $x \in \mathrm{Sym}(\Omega)$ and any given $H \leq \mathrm{Sym}(\Omega)$.*

PROOF. Let $\mathcal{F}$ be a fixed structure forest for G. Taking any planar layout of $\mathcal{F}$, with the leaves (i.e., the set Ω) situated at the same level, order the leaves left-to-right.

To accommodate recursion, we describe a general procedure for finding $\mathrm{LL}_\Omega(zM)$ where zM is any left coset of $M \leq \mathrm{Sym}(\Omega) \times \mathrm{Sym}(\Omega)$ and $\mathrm{pr}_1(M) < G$. (Our overall goal is the special case $\ell\ell_\Omega((1,x)G \times H)$.) Note that we may consider M as acting on Ω via either pr_1 or pr_2, and $\mathrm{pr}_1(M)$ also acts on $\mathcal{F}$.

Assuming the orbits of G occur in the order $\Omega_1, \dots, \Omega_m$, we have, by Fact 1,

$$\mathrm{LL}_\Omega(zM) = \mathrm{LL}_{\Omega_m}(\cdots(\mathrm{LL}_{\Omega_1}(zM))\cdots).$$

By Fact 3, the intermediate answers are always left cosets.

Thus it suffices to describe the construction of $\mathrm{LL}_\Phi(zM)$ where Φ is the set of roots descendent from a node ν in $\mathcal{F}$ and ν (therefore Φ) is fixed by $\mathrm{pr}_1(M)$.

If ν is a leaf, then $\Phi = \{\phi\}$. In this case, $\ell\ell_\Phi(zM)$ is the least element in the orbit of $\phi^{\mu(z)}$ under $\mathrm{pr}_2(M)$. If wL is the subcoset of M mapping $\phi^{\mu(z)}$ to $\ell\ell_\Phi(zM)$ (via the pr_2 action) then $\mathrm{LL}_\Phi(zM) = zwL$.

If ν is not a leaf, then let L be the subgroup of M that fixes the immediate children, $\nu_1, \dots, \nu_r$ (listed left-to-right), of ν (L is found by an application of (3.6)) and find a left transversal $\{w_1, \dots, w_{|M:\,L|}\}$ for L in M, so

$$M = \bigcup_{i=1}^{|M:\,L|} w_i L.$$

By Fact 2, computation of $\mathrm{LL}_\Phi(zM)$ follows from computation of $\mathrm{LL}_\Phi(zw_iL)$ for $1 \leq i \leq |M:\,L|$. By Fact 3, each contributing subanswer, i.e., each $\mathrm{LL}_\Phi(zw_iL)$, for which $\ell\ell_\Phi(zw_iL) = \ell\ell_\Phi(zM)$, is a coset v_iK of the same subgroup $K = \mathrm{Stab}_L(\{(\delta, \delta^{\ell\ell_\Delta(zM)}) \mid \delta \in \Delta\})$, so that the subanswers $v_{i_1}K, \dots, v_{i_s}K$ glue together to a coset as in:

$$v_{i_1}K \cup \cdots \cup v_{i_s}K = v_{i_1}\langle K, \{v_{i_1}^{-1}v_{i_t}\}_{2\leq t\leq s}\rangle.$$

Finally, to compute $\mathrm{LL}_\Phi(zw_iL)$, we exploit the fact that $\mathrm{pr}_1(L)$ stabilizes each ν_i, in the iterative approach

$$\mathrm{LL}_\Phi(zw_iL) = \mathrm{LL}_{\Phi_r}(\cdots(\mathrm{LL}_{\Phi_1}(zw_iL))\cdots),$$

where Φ_i denotes the set of leaves descendent from ν_i.

For the purpose of timing, we observe that, that $|M:\,L|$ is bounded by the size of the primitive group in the action of G_ν on $\{\nu_1, \dots, \nu_r\}$. Suppose now that $G \in \Gamma_d$. Then $|M:\,L| \leq O(r^{cd})$. Thus, the one problem on Φ has involved at most $O(r^{cd+1})$ recursive calls to problems on sets of size $|\Phi|/r$. It follows that the timing for the entire procedure is $O(n^{cd+c'})$. □

Since double cosets can be compared when lex-least elements are available, Theorem 6.2 has immediate applications to the problems of Section 4.

COROLLARY 6.3. *Let d be fixed. Given $G < \mathrm{Sym}(\Omega)$, with $G \in \Gamma_d$, in polynomial time one can*

(i) *for any $\Delta_1, \Delta_2 \subseteq \Omega$, test whether there exists $g \in G$ such that $\Delta_1^g = \Delta_2$;*
(ii) *for any given $H \leq \mathrm{Sym}(\Omega)$ and any $x_1, x_2 \in \mathrm{Sym}(\Omega)$ test whether $Gx_1H = Gx_2H$;*
(iii) *for any $x_1, x_2 \in \mathrm{Sym}(\Omega)$, test whether there exists $g \in G$ such that $x_1^g = x_2$.* □

The methods of Section 4 yield polynomial-time equivalent "AUTO" versions for the statements in Corollary 6.3.

COROLLARY 6.4. *Let d be fixed. Given $G < \mathrm{Sym}(\Omega)$, with $G \in \Gamma_d$, in polynomial time one can*

(i) *for any $\Delta \subseteq \Omega$, find $\mathrm{Stab}_G(\Delta)$;*
(ii) *for any given $H \leq \mathrm{Sym}(\Omega)$, find $G \cap H$;*
(iii) *for any $x \in \mathrm{Sym}(\Omega)$, find $\mathrm{C}_G(x)$.* □

Remarks. The timing in all of these results, as implied by the proof of Theorem 6.2, can be expressed in the form $O(|\Omega|^{cd})$, for constant c. An improvement described in [4] results in the timing $O(|\Omega|^{cd/\log d})$.

If $\mathcal{G} = (V, E)$ is a connected graph of valence d, and $e \in E$, then $\mathrm{Aut}(\mathcal{G})_e \in \Gamma_{d-1}$. This observation, together with the result in Corollary 6.4(i), was used in [20] to establish a polynomial-time isomorphism test for graphs of bounded valence. Using the improved timing as above, one gets isomorphism-testing for valence-d graphs in time $O(|V|^{cd/\log d})$ (so the exponent is $o(d)$). Together with the "valence-reduction" trick of Zemlyachenko [31], this, in turn, yields the best-known timing for general graph isomorphism, $O(n^{\sqrt{cn/\log n}})$ [4].

The result of Corollary 6.4(i) also underlies polynomial-time isomorphism tests for a broader class of graphs generalizing both bounded valence and bounded genus [25, 26].

We remark, finally, that Corollary 6.4 can be approached directly, and possibly a bit more compactly, than via Theorem 6.2. However, there is some dividend in the lex-least approach. For example, one can apply it to find *canonical forms* in the above graph classes [6].

7. Exploiting Normality

Problems that involve finding *normal* subgroups often have efficient solutions according to the criterion of this paper. We illustrate the point, in this section, with the problems of Section 4. Other examples are given in Sections 8 and 9.

We first consider INTER.

PROPOSITION 7.1. *Given $G, H \leq \mathrm{Sym}(\Omega)$, where G normalizes H, then $G \cap H$ can be found in polynomial time.*

PROOF. This is an application of (3.5), for we have the tower

$$G \cap H = G \cap G^{(n)}H \leq G \cap G^{(n-1)}H \leq \cdots \leq G \cap G^{(2)}H \leq G \cap G^{(1)}H = G.$$

Generators for $G^{(i)}H$ are available (union of generators for $G^{(i)}$ and generators for H) and so membership-testing in both G and $G^{(i)}H$, therefore in $G \cap G^{(i)}H$, is in polynomial time. Moreover,

$$|G \cap G^{(i-1)}H : G \cap G^{(i)}H| \leq |G^{(i-1)}H : G^{(i)}H| \leq |G^{(i-1)} : G^{(i)}| \leq n - i.$$

Hence, (3.5) applies. □

The result generalizes to

COROLLARY 7.2. *Given G and H such that $H \lhd\lhd \langle G, H\rangle$. Then $G \cap H$ can be found in polynomial time.*

PROOF. If G normalizes H then apply Proposition 7.1. Otherwise, since $H \lhd\lhd \langle G, H\rangle$, $H^G < \langle G, H\rangle$ and so $G \cap H^G < G$. It suffices then to observe that $G \cap H = H \cap (G \cap H^G)$, which we compute recursively ($G \cap H^G$ being obtained by the proposition). □

Remark. In particular, Corollary 7.2 offers an alternative approach for intersecting subgroups of a nilpotent group (wherein all subgroups are subnormal). The method appears substantially different from the orbit and imprimitivity-blocks divide-and-conquer that led to Corollary 6.4(ii).

If a targeted normal subgroup $N \trianglelefteq G$ can be interpreted as the kernel of some induced action $\pi : G \to \mathrm{Sym}(\Psi)$, then $N = G_\Psi$ (obtainable in polynomial time by (3.6)). We use this in several places, including the following.

PROPOSITION 7.3. *Given $G, H \leq \mathrm{Sym}(\Omega)$, where G normalizes H, then $\mathrm{C}_G(H)$ can be found in polynomial time.*

PROOF. We describe an action $\pi : G \to \mathrm{Sym}(\Psi)$, with $|\Psi| \leq |\Omega|$. Then if $K = \ker(\pi)$, we describe a new action $\phi : K \to \mathrm{Sym}(\Omega)$ such that $\mathrm{C}_G(H) = \ker(\phi)$.

Let Ψ be the set of equivalence classes in Ω relative to the relation defined by $\alpha \sim \beta \Leftrightarrow H_\alpha = H_\beta$. Let $\pi : G \to \mathrm{Sym}(\Psi)$ be the action of G induced by conjugation. Note that $\mathrm{C}_G(H) \leq K = \ker(\pi)$.

To define ϕ, fix a point α_Δ in each orbit Δ of H. Then for $k \in K$ let

$$(\alpha_\Delta^h)^{\phi(k)} = \alpha_\Delta^{h^k}.$$

Since $H_{\alpha_\Delta}^k = H_{\alpha_\Delta}$, ϕ is well-defined, whence it is immediate that ϕ is a homomorphism. We need to verify only that $\mathrm{C}_G(H) = \ker(\phi)$

Clearly, if k centralizes H then $k \in \ker(\phi)$. Conversely, suppose $k \in \ker(\phi)$. Let $h \in H$; we must show $hk = kh$. For any $\omega \in \Omega$, $\omega = \alpha_\Delta^{h_1}$ for some Δ and some $h_1 \in H$. Since $k \in \ker(\phi)$,

$$\omega^h = \alpha_\Delta^{h_1 h} = \alpha_\Delta^{(h_1 h)^k} = (\alpha_\Delta^{h_1^k})^{h^k} = (\alpha_\Delta^{h_1})^{h^k} = \omega^{h^k}.$$

Hence $h^k = h$. □

Remarks. In [21], it is observed that, when G normalizes H, $\mathrm{C}_G(H)$ can be directly interpreted as a kernel, though the action is on a set of size $O(|\Omega^2|)$. The above approach avoids this blowup in space demands.

Proposition 7.1 offers still another approach, as centralizers in $\mathrm{Sym}(\Omega)$ can be found in polynomial time (see, e.g., [10] or [15]). With that in mind, we can employ $\mathrm{C}_G(H) = G \cap \mathrm{C}_{\mathrm{Sym}(\Omega)}(H)$.

Proposition 7.3 has the immediate corollary

Corollary 7.4. *Given $G \leq \mathrm{Sym}(\Omega)$, then the center of G can be found in polynomial time.* □

In practice, centers are typically computed by cutting down to the centralizers of successive generators. Since the elements to centralize are chosen in a special way, for example, the first one from within the group itself, one might ask whether there may be a polynomial-time approach of this sort, notwithstanding the uncertain complexity of general CENT. However, we observe that in the first round, one is already solving a problem as hard as CENT. Consider

Problem. INTERNAL-CENTRALIZER (INT-CENT)
Input: $G \leq \mathrm{Sym}(\Omega)$; $x \in G$.
Find: $\mathrm{C}_G(x)$.

Unfortunately,

Proposition 7.5. *INT-CENT is polynomial-time equivalent to STAB.*

Proof. Suppose INT-CENT is in polynomial time. Then, with notation as in the reduction of STAB to CENT (in proof of Proposition 4.3), in the faithful action of $\langle G, x\rangle$ on $\widehat{\Omega}$ we could find $\mathrm{C}_{\langle G,x\rangle}(x)$ Observe, however, that $\langle G, x\rangle$ also acts on the system $\overline{\Omega} = \{\{\omega_1, \omega_2\} \mid \omega \in \Omega\}$, which may be identified with Ω (via $(\omega_1, \omega_2) \leftrightarrow \omega$); the resulting action of $\mathrm{C}_{\langle G,x\rangle}(x)$ on Ω is $\mathrm{C}_G(x)$. □

Remarks. We note that the proof shows finding $\mathrm{C}_G(x)$ is "no easier" when $x \in G$ is an involution.

CONJ-ELT (Section 4) has an analogous "internal" case in which x_1, x_2 are assumed to be in G. Again, this is polynomial-time equivalent to the general problem.

Corollary 7.2 inspires the question of whether the following is also in polynomial time.

PROBLEM. SUBNORMAL-CENTRALIZER (SUBNORM-CENT)
INPUT: $G, H \leq \mathrm{Sym}(\Omega)$ *with* $H \lhd\lhd G$.
FIND: $\mathrm{C}_G(H)$.

However, this problem, too, is no easier then CENT.

PROPOSITION 7.6. *SUBNORM-CENT is polynomial-time equivalent to STAB.*

PROOF. In the above discussion of INT-CENT, $\langle x\rangle^G$ is an elementary abelian 2-group, so that $\langle x\rangle \trianglelefteq \langle x\rangle^G \trianglelefteq \langle G, x\rangle$, whence $\langle x\rangle \lhd\lhd \langle G, x\rangle$. □

Proposition 7.3 does give a bit of information about general centralizers.

COROLLARY 7.7. *Given* $G, H \leq \mathrm{Sym}(\Omega)$, *then* $\mathrm{Core}_G(\mathrm{C}_G(H))$ *can be found in polynomial time.*

PROOF. $\mathrm{Core}_G(\mathrm{C}_G(H)) = \mathrm{C}_G(H^G)$. □

Following the reduction of STAB to CENT, this immediately yields

COROLLARY 7.8. *Given* $G \leq \mathrm{Sym}(\Omega)$ *and* $\Delta \subseteq \Omega$, *then* $\mathrm{Core}_G(\mathrm{Stab}_G(\Delta))$ *can be found in polynomial time.* □

Corollaries 7.7 and 7.8 prompt the question of whether $\mathrm{Core}_G(G \cap H)$ can be computed in polynomial time. It can. However, we do not know an "elementary" proof (see Proposition 8.6).

Remarks. The polynomial-time methods for Propositions 7.1, 7.3 ultimately utilize the fact that the targeted subgroup $H \leq G$ lies in a chain

$$H = H_m \leq H_{m-1} \leq \cdots \leq H_0 = G \tag{1}$$

with $|H_i : H_{i+1}|$ "small". In fact, this is true for any $H \lhd\lhd G$, where "small" can be interpreted as $\leq n$. To show this, it suffices to consider to assume $H \trianglelefteq G$, in which case

$$H = HG^{(n)} \leq \cdots HG^{(2)} \leq HG^{(1)} = G.$$

This suggests that (3.5) should provide the tool for finding targeted normal subgroups much more generally. The difficulty that arises, however, is that we do not have, a priori, ways of "recognizing" the intermediate groups. (See Question 10, for example.)

One expects, also, to find normal subgroups as kernels of actions. However, for arbitrary $N \lhd G \leq \mathrm{Sym}(\Omega)$, G/N may not be representable on a polynomial-size set [27]. One knows, however, for $H \lhd\lhd G$, there is a chain

$$H = L_m \trianglelefteq L_{m-1} \trianglelefteq \cdots \trianglelefteq L_0 = G$$

with $L_i/L_{i+1} \hookrightarrow \mathrm{Sym}(\Omega)$ for each i. (To show this, we may assume $H \trianglelefteq G$; using the chain in (1), inductively let L_{i+1} be the kernel of the right-multiplication action of L_i on the right cosets of H_{i+1} in H_i.) Call the minimal such m the *depth* of L in G. It is not hard to show that if $L \trianglelefteq G$ then $m = O(\log^2 |\Omega|)$. (This reduces easily to the primitive case, wherein one uses the Cameron classification of primitive groups, see, e.g., [7].)

Though we are not sure of polynomial-time implications, the following question seems of interest.

QUESTION 7. *What is the least upper bound on the depths of normal and subnormal subgroups in permutation groups?*

The proof of Proposition 7.3 shows, for example, that the depth of the centralizer of a normal subgroup is at most 2.

8. Quotient Groups

In [18], Kantor and Luks suggest the thesis that problems that are in polynomial time for permutation groups remain in polynomial time for quotients of permutation groups. The justification is not, however, via routine consideration of the quotients as permutation groups, as is often the case in available systems, inasmuch as quotients may not have any small (polynomial-size) faithful permutation representations [27]. The generalizations of problems INTER and CENT provide good illustrations of the techniques that are brought to bear in [18].

PROBLEM. QUOTIENT-INTERSECTION (Q-INTER)
INPUT: $G, H, K \leq \mathrm{Sym}(\Omega)$ *with* $K \trianglelefteq G$, $K \trianglelefteq H$.
FIND: $G/K \cap H/K$.

PROBLEM. QUOTIENT-CENTRALIZER (Q-CENT)
INPUT: $G, K \leq \mathrm{Sym}(\Omega)$; $x \in \mathrm{Sym}(\Omega)$, *with* $K \trianglelefteq G$ *and* x *normalizing* K.
FIND: $\mathrm{C}_{G/K}(xK/K)$.

(G/K, H/K may be considered as contained in the group $\langle G, H\rangle/K$ and G/K, xK/K in the group $\langle G, x\rangle/K$.)

Since permutation representations of the quotients may be infeasible, the question arises of whether these problems present a still higher level of challenge. However,

PROPOSITION 8.1. *Q-INTER and Q-CENT are polynomial-time equivalent to STAB.*

PROOF. It is obvious that Q-INTER is no harder than INTER, since $G/K \cap H/K = (G \cap H)/K$.

Reduction of Q-CENT to STAB: Let (G, K, x) be an instance of Q-CENT. Let $\hat{G} = \{(g, gk) \mid g \in G, k \in K\}$ acting on $\Omega \times \Omega$. For $x \in G$ let $\Delta(x) = \{(\omega, \omega^x) \mid \omega \in \Omega\} \subseteq \Omega \times \Omega$. Then for $(g, gk) \in \hat{G}$, $\Delta(x)^{(g,gk)} = \Delta(x^g k)$, so that (g, gk) stabilizes $\Delta(x)$ iff $[x, g] = k^{-1}$. But $gK \in \mathrm{C}_{G/K}(xK/K)$ iff there exists $k \in K$ such that $[g, x] = k$. Hence, if we compute $\mathrm{Stab}_{\hat{G}}(\Delta(x))$ and let H be its first coordinate projection, we have $\mathrm{C}_{G/K}(xK/K) = H/K$. $\square$

Remark. Similarly, the quotient versions of DC-EQ and ELT-CONJ are polynomial-time equivalent to the permutation-group cases.

Thus, it seems, from a polynomial-time perspective, that these problems do not get any harder for quotients. In a positive direction, we next show that the instances where INTER and CENT are in polynomial time generalize to quotient groups.

The following is just a repeat of Proposition 7.1.

PROPOSITION 8.2. *Given $G, H, K \leq \mathrm{Sym}(\Omega)$, with $K \trianglelefteq G$ and $K \trianglelefteq H$, and where G/K normalizes H/K, then $G/K \cap H/K$ can be found in polynomial time.*

PROOF. $G/K \cap H/K = (G \cap H)/K$ and the hypotheses imply G normalizes H. □

Corollary 7.2 generalizes immediately, as well.

Generalizations of Corollary 6.4(ii,iii) and Proposition 7.3 require a surprising amount of additional machinery.

The following is proved in [**16**, **17**].

LEMMA 8.3 (KANTOR). *Given $G \leq \mathrm{Sym}(\Omega)$ then*

(i) *For any prime p dividing $|G|$, a Sylow p-subgroup of G can be found in polynomial time.*

(ii) *Given Sylow p-subgroups P_1, P_2 of G, some $g \in G$ such that $P_1^g = P_2$ can be found in polynomial time.*

(iii) *Given $K, P \leq \mathrm{Sym}(\Omega)$ with P a Sylow p-subgroup of K and $K \trianglelefteq G$, then $\mathrm{N}_G(P)$ can be found in polynomial time.*

Quite unlike the methods being described in this paper, which have relied on elementary group theory, the algorithms and proofs underlying Lemma 8.3 use substantial consequences of the classification of finite simple groups, including detailed knowledge of simple-group types. Nevertheless, that having been done, it is demonstrated in [**18**] that one can effectively use the result as a "black-box" in further, once again elementary, arguments. We illustrate first with a generalization of Corollary 6.4(iii). This involves a constructive version of the well-known

FRATTINI ARGUMENT. *Let $P \leq K \trianglelefteq G$ with P a Sylow p-subgroup of K. Then $G = K\mathrm{N}_G(P)$.*

PROPOSITION 8.4. *Given $G, K \leq \mathrm{Sym}(\Omega)$ and $x \in \mathrm{Sym}(\Omega)$, where $K \trianglelefteq G$, x normalizes K, and with $G/K \in \Gamma_d$, then $\mathrm{C}_{G/K}(xK/K)$ can be found in polynomial time.*

PROOF. Note that the Γ_d hypothesis applies only to G/K. If, however, $G \in \Gamma_d$ then the reduction in the proof of Proposition 8.1 would lead to an instance of STAB with a group, $\hat{G}$, in Γ_d, whence we could apply 6.4(i).

So suppose $G \notin \Gamma_d$. Then $K \notin \Gamma_d$. In particular, K is not nilpotent, so that, for some prime p dividing $|K|$, any Sylow p-subgroup, P, of K is not normal. Find generators for such a P as well as $G_1 = \mathrm{N}_G(P)$ and $K_1 = \mathrm{N}_K(P)$, and find $k \in K$ such that $P^k = P^x$ (Lemma 8.3). Let $y = xk^{-1}$, so y normalizes P and therefore normalizes K_1.

Recursively compute $L/K_1 = \mathrm{C}_{G_1/K_1}(yK_1/K_1)$. Then $\mathrm{C}_{G/K}(xK/K) = LK/K$.

The recursive procedure runs in polynomial time since $G_1 < G$. The correctness is a consequence of a Frattini argument: Since $G = G_1K$, it suffices to show, for $g_1 \in G_1$ that g_1K centralizes xK/K (in $\langle G, x\rangle/K$) iff g_1K_1 centralizes yK_1 (in $\langle L, y\rangle/K_1$). But g_1K centralizes xK/K iff $[g_1, x] \in K$ iff $[g_1, y] \in K$ iff $[g_1, y] \in K_1$ (since $K_1 = \mathrm{N}_K(P)$ and both g_1 and y normalize P) iff g_1K_1 centralizes yK_1. □

A similar Frattini argument (see [18]) is used for the following extension of Corollary 6.4(ii).

PROPOSITION 8.5. *Given $G, H, K \leq \mathrm{Sym}(\Omega)$, where $K \trianglelefteq G$, $K \trianglelefteq H$, and with $G/K \in \Gamma_d$, then $G/K \cap H/K$ can be found in polynomial time.* □

We reiterate that, while Corollary 6.4(ii,iii) has been extended to quotient groups, the fact that the extensions are dependent upon Lemma 8.3, means that we have now had to invoke the classification of simple groups. On the other hand, in the special case when $\langle G/K, H/K\rangle$ is solvable, there are "elementary" proofs of Propositions 8.4 and 8.5 based upon results in [22].

QUESTION 8. *Is there an "elementary" construction of $\mathrm{C}_{G/K}(H/K)$ and/or $G/K \cap H/K$ if only G/K is assumed to be solvable?*

Our extension of Proposition 7.3 to quotient groups requires the ability to compute cores of given subgroups of permutation groups. In practice, this is commonly done by intersecting conjugates until the resulting group is normal. Since intersections are not presently available, this approach is not yet feasible in polynomial time. Nevertheless, cores are attainable. Following the theme of Section 7, we observe that the normality of the targeted group facilitates this.

PROPOSITION 8.6. *Given $G, H \leq \mathrm{Sym}(\Omega)$, then $\mathrm{Core}_G(G \cap H)$ can be found in polynomial time.*

PROOF. For each prime p, find a Sylow p-subgroup, P_p, of G. Since $\mathrm{Core}_G(G \cap H) \trianglelefteq G$,

$$\mathrm{Core}_G(G \cap H) = \langle \{P_p \cap \mathrm{Core}_G(G \cap H) \mid p \text{ divides } |G|\}\rangle.$$

It suffices to determine $P_p \cap \mathrm{Core}_G(G \cap H)$ for each p. This is made feasible by the fact that we can test membership in $\mathrm{Core}_G(G \cap H)$, that is, if $g \in G$ then $g \in \mathrm{Core}_G(G \cap H)$ iff $\langle g \rangle^G \leq H$. Thus, initially setting $T = P_p$, test whether $T^G \leq H$ and, if so, output T; else we can find $g \in G$ such that $T^g \not\leq H$ (the computation of generators for the normal closure, (3.11), can maintain generators as conjugates of the generators of T) and repeat with $T := T \cap H^{g^{-1}}$ (intersect by Corollary 6.4(ii)).

The procedure succeeds because the relation $T \cap \mathrm{Core}_G(G \cap H) = P_p \cap \mathrm{Core}_G(G \cap H)$ is maintained. □

It is immediate that

COROLLARY 8.7. *Given $G, H \leq \mathrm{Sym}(\Omega)$ with $H \leq G$, then $\mathrm{Core}_G(H)$ can be found in polynomial time.* □

Remark. The proof of Proposition 8.6 provides a striking counterpoint to that of Corollaries 7.7 and 7.8. While the latter two were elementary, the former uses Lemma 8.3 which, in turn, uses the classification of finite simple groups. On the other hand, we observe in Section 9 that another problem (finding p-cores) which, in practice, has seemed to require construction of Sylow subgroups, has a direct and elementary approach. Once again we are led to questions about the existence of non-classification-dependent arguments.

QUESTION 9. *Is there an "elementary" approach to finding $\mathrm{Core}_G(G \cap H)$ or even for finding $\mathrm{Core}_G(H)$ when $H \leq G$?*

In particular, considering the remarks at the end of Section 7,

QUESTION 10. *Is there an "elementary" construction of a chain*

$$\mathrm{Core}_G(H) = N_0 \trianglelefteq N_1 \trianglelefteq \cdots \trianglelefteq N_m = G$$

in which N_{i+1} is the kernel of a "small" degree representation of N_i?

Of course, these issues may lie with Lemma 8.3 itself.

QUESTION 11. *Is there an "elementary" approach to finding Sylow subgroups?*

Can one even get started?

QUESTION 12. *Is there an "elementary" method for locating an element of order p where p is a prime dividing $|G|$?*

Returning to the main issue, we complete this section with the following extension of Proposition 7.3.

COROLLARY 8.8. *Given $G, H, K \leq \mathrm{Sym}(\Omega)$, with $K \trianglelefteq G$ and $K \trianglelefteq H$, and where G normalizes H, then $\mathrm{C}_{G/K}(H/K)$ can be found in polynomial time.*

PROOF. Consider $G \times G$ acting naturally on the disjoint union, $\Omega\dot{\cup}\Omega$, of two copies of Ω. Let $L = \{(g, gk) \mid g \in G, k \in K\}$ and $M = \{(g, gh) \mid g \in G, h \in H\}$. Find $\mathrm{Core}_M(L)$ (Corollary 8.7) and let C be the group obtained by restricting $\mathrm{Core}_M(L)$ to the first copy of Ω. Output C/K.

We show that $C/K = \mathrm{C}_{G/K}(H/K)$, i.e., that for $g \in G$, $(g, gk) \in \mathrm{Core}_M(L)$ for some $k \in K$ iff gK centralizes H/K (in $\langle G, H\rangle/K$). Since $K \times K \trianglelefteq M$, we have $K \times K \leq C$ so that $K \leq C$. Then, for $(g, gk) \in L$,

$$\begin{aligned}(g,gk) = (g,g)(1,k) \in \mathrm{Core}_M(L) \quad &\text{iff} \quad (g,g)^M \subseteq L\\ &\text{iff} \quad (g,g)^{(1,h)} \in L,\ \forall h \in H\\ &\text{iff} \quad g^{-1}g^h \in K,\ \forall h \in H\\ &\text{iff} \quad gK \text{ centralizes } H/K. \quad \square\end{aligned}$$

9. p-Cores

For any prime p and group G, the *p-core* of G is the (unique) maximal normal p-subgroup of G and is denoted $\mathrm{O}_p(G)$.

THEOREM 9.1. *Given $G \leq \mathrm{Sym}(\Omega)$, then $\mathrm{O}_p(G)$ can be found in polynomial time.*

A suggested method for computing the p-core of a permutation group has been to find a Sylow p-subgroup $P \leq G$ and then use

$$\mathrm{O}_p(G) = \mathrm{Core}_G(P).$$

This does give a polynomial-time solution. However, the conceptual overhead in this approach to $\mathrm{O}_p(G)$ is that the known method for finding P (Lemma 8.3) uses the classification of finite simple groups. Nevertheless, unlike the situation for general cores, we offer a self-contained elementary proof of Theorem 9.1, giving another measure of support for the theme that *normal* targets are easier to locate. (See [27] for another direct approach to p-cores.)

A few lemmata are required.

LEMMA 9.2. *Given a transitive $G \leq \mathrm{Sym}(\Omega)$ with $|G| > n$, one can find a proper normal subgroup or else establish that G does not have a* regular abelian *normal subgroup.*

PROOF. If $G^{(3)} = 1$ then $|G| < n^2$. (Recall that $G^{(3)}$ is the subgroup fixing ω_1 and ω_2.) In that case, the elements of G can be listed and the normal closure of the group generated by each can be computed in polynomial time. If none of these yield a proper normal subgroup then G does not have a regular abelian normal (or *any* proper normal) subgroup.

Assume $G^{(3)} \neq 1$ and let $\Psi = \{(G^{(3)})^g \mid g \in G\}$. Then $1 < |\Psi| \leq \binom{n}{2}$ and G acts transitively (via conjugation) on Ψ. Let $\mathcal{B}$ be a minimal G-block system in Ψ (i.e., start with $\mathcal{B} = \Psi$ and while G does not act primitively on $\mathcal{B}$, replace

$\mathcal{B}$ by a nontrivial partition of $\mathcal{B}$ into blocks of imprimitivity). Output $G_{\mathcal{B}}$ (the kernel of the action of G on $\mathcal{B}$) if it is proper, else declare that G does not have a regular abelian normal subgroup.

We must show, under the assumption that G has a regular abelian normal subgroup A that G does not act faithfully on $\mathcal{B}$. Since A is regular, there is a unique $a \in A$ such that $\omega_1^a = \omega_2$. Such a normalizes, in fact centralizes, $G^{(3)}$, for if $x \in G^{(3)}$, both a and $x^{-1}ax$ are elements of A mapping ω_1 to ω_2 so that $a = x^{-1}ax$. Hence a fixes the block in $\mathcal{B}$ containing $G^{(3)}$. We conclude that A does not act regularly on $\mathcal{B}$. But then A cannot act faithfully on $\mathcal{B}$, for a normal subgroup of a primitive group is transitive and so, if it is abelian, it is regular. □

Remarks. The above algorithm simplifies one with an analogous purpose in [21]. The modification is due to Á. Seress. (See also [7].)

Note that the output of a proper normal subgroup leaves open the question of whether there is a regular abelian normal subgroup, thus leading us to ask

QUESTION 13. *Given $G \leq \mathrm{Sym}(\Omega)$, can one determine, in polynomial time, whether G has a regular abelian normal subgroup and, if so, find one?*

More generally,

QUESTION 14. *Given $G \leq \mathrm{Sym}(\Omega)$, can one determine, in polynomial time, whether G has a regular normal subgroup and, if so, find one?*

Both of these issues are in polynomial time for primitive groups: if a primitive group G has an abelian normal subgroup N, then $N = \mathrm{O}_p(G)$ for (the unique) prime p dividing n; in general, if H is the smallest nontrivial term in a composition series for G ([21]), then G has a regular normal subgroup iff H^G is regular.

LEMMA 9.3. *Given $G \leq \mathrm{Sym}(\Omega)$, in polynomial time one can find a proper normal subgroup of G or else establish that $\mathrm{O}_p(G) = 1$.*

Remark. Output of a proper normal subgroup does not yet mean $\mathrm{O}_p(G) \neq 1$.

PROOF OF LEMMA. Let Δ be any nontrivial orbit of G and construct a minimal G-block system $\mathcal{B}$ in Δ (so that G acts primitively on $\mathcal{B}$). Let $\phi : G \to \mathrm{Sym}(\mathcal{B})$ be the induced action. If $K = \ker(\phi) \neq 1$, output generators for K. Otherwise, the primitive group $\phi(G)$ is isomorphic to G. We may assume $|G| > n$ else G is listable and the lemma resolvable by brute force. Apply Lemma 9.2 to $\phi(G)$. If $\phi(G)$ has no abelian regular normal subgroup then $\mathrm{O}_p(G) = 1$ (else, if A is the the last nontrivial term in the derived series for $\mathrm{O}_p(G)$, $\phi(A)$ would be a regular normal subgroup of the primitive group). Otherwise, the call to Lemma 9.2 produces Y, generating a proper normal subgroup, in which case return $N = \phi^{-1}(Y)$. (The lifting $\phi^{-1}(y)$ for $y \in Y$ is computed, for example, (3.9); alternatively, while computing Y, keep track of liftings of elements yielding $Y' \subseteq G$ with $\phi(Y') = Y$; then $N = \langle Y', K \rangle$.) □

LEMMA 9.4. *Given $G \leq \mathrm{Sym}(\Omega)$, in polynomial time one can find a nontrivial normal p-subgroup of G or else establish that $\mathrm{O}_p(G) = 1$.*

PROOF. We describe a procedure p-NORM(G) with output as indicated.

Apply Lemma 9.3. If we discover $\mathrm{O}_p(G) = 1$ then return that information. Otherwise we have $1 \neq N \lhd G$ and we proceed as follows.

Recursively call p-NORM(N). If the call returns $P \trianglelefteq N$, then output P^G. Else ($\mathrm{O}_p(N) = 1$) recursively call p-NORM($\mathrm{C}_G(N)$) (using Proposition 7.3 to find $\mathrm{C}_G(N)$). If the call returns for $P \trianglelefteq \mathrm{C}_G(N)$ then output P^G. Else report "$\mathrm{O}_p(G) = 1$."

The procedure succeeds since $\mathrm{O}_p(N) = 1$ implies $\mathrm{O}_p(G) \cap N = 1$, whence $\mathrm{O}_p(G) \leq \mathrm{C}_G(N)$.

Timing concern: What if both recursive calls are made? That only happens when $\mathrm{O}_p(N) = 1$ so that p does not divide $|N \cap \mathrm{C}_G(N)|$, whence

$$|N|_p|\mathrm{C}_G(N)|_p = |N\mathrm{C}_G(N)|_p \leq |G|_p$$

where sub-p denotes p-part. Thus, except for multiplicative contributions from known polynomial timings, the time is linear in $\log |G|_p$. □

LEMMA 9.5. *Given $P, G \leq \mathrm{Sym}(\Omega)$ with $P \lhd G$ where P is a p-group and G is not a p-group, one can construct in polynomial time another action $\phi : G \to \mathrm{Sym}(\Omega)$, where $\ker(\phi)$ is a nontrivial normal p-subgroup of G.*

PROOF. Replacing P, if necessary, by the last nontrivial term in its derived series, we may assume that P abelian. Let $\{\Delta_i\}_{i \in I}$ be the set of orbits of P and let $\pi : G \to \mathrm{Sym}(I)$ be the naturally induced action, i.e., $\Delta_i^g = \Delta_{i^{\pi(g)}}$ for $i \in I, g \in G$. Choose $\delta_i \in \Delta_i$ for each $i \in I$. Then ϕ is defined via

$$(\delta_i^x)^{\phi(g)} = \delta_{i^{\pi(g)}}^{x^g},$$

for $i \in I, x \in P$. (The superscripts x, x^g denote the given action.) That ϕ is well defined follows from the fact that P^{Δ_i} is regular (since it is abelian), for, if $\delta_i^x = \delta_i^y$ for $x, y \in P$, then x and y act identically on Δ_i so that x^g and y^g act identically on $\Delta_{i^{\pi(g)}}$. From this it is straightforward to see ϕ is a homomorphism. Since $\ker(\phi)$ stabilizes each Δ_i and commutes with the action of P thereon, $\ker(\phi)^{\Delta_i} = P^{\Delta_i}$. Hence $\ker(\phi)$ is an abelian p-group containing P. It is proper in G as G is not a p-group. □

PROOF OF THEOREM 9.1. We may assume G is not a p-group. By Lemma 9.4, we establish immediately that $\mathrm{O}_p(G) = 1$ or else obtain a proper normal p-subgroup K. In the latter case, we apply Lemma 9.5 to obtain an action $\phi : G \to \mathrm{Sym}(\Omega)$ with $1 < K = \ker(\phi) \lhd G$. Recursively, compute $\langle Y \rangle = \mathrm{O}_p(\phi(G))$. Then $\mathrm{O}_p(G) = \phi^{-1}(Y)$ (computed, say, via (3.9), wherein it is convenient to consider $\phi(G)$ as acting on a disjoint copy of Ω).

For correctness, we observe that, since $\ker(\phi)$ is a p-group, $\phi(\mathrm{O}_p(G)) = \mathrm{O}_p(\phi(G))$.

For the timing, note that the recursive call involves a smaller group $\phi(G)$ on a permutation domain of the same size. □

Remark. In [18] it is pointed out that, in polynomial time, one can construct the maximal normal subgroup with composition factors in any specified collection of simple groups, but the general result ultimately makes use of the classification of finite simple groups.

10. Other Problems and their Relationships

We comment on several other problems resembling GRAPH-ISO and STAB, etc. There are open questions, not only about when they are in polynomial time, but in the relationships among them.

10.1. Finding Subgroups. Possibly presenting a challenge beyond STAB is

PROBLEM. NORMALIZER (NORM)
INPUT: $G, H \leq \mathrm{Sym}(\Omega)$.
FIND: $\mathrm{N}_G(H)$.

Techniques announced in [22] show that NORM is in polynomial time when $\langle G, H\rangle$ is solvable. Questions that immediately arise include

QUESTION 15. *Is NORM in polynomial time when only G is assumed to be solvable?*

The next step up the group ladder would appear to be

QUESTION 16. *Is NORM in polynomial time when $\langle G, H\rangle$ is in Γ_d (See Section 6).*

How is NORM related to the problems of Section 4? STAB reduces to NORM, either by Proposition 7.1, or, following the reduction of STAB to CENT in Proposition 4.3, the fact that x is an involution implies $N_g(\langle x\rangle) = C_G(x)$. But is NORM, in general, "harder" than STAB, etc.?

QUESTION 17. *Is there a polynomial-time reduction of NORM to STAB?*

For this question, notice that it would suffice to find a polynomial-time solution to the special case

PROBLEM. NORMALIZER IN SYMMETRIC GROUP (NORM-SYM)
INPUT: $G \leq \mathrm{Sym}(\Omega)$.
FIND: $\mathrm{N}_{\mathrm{Sym}(\Omega)}(G)$.

Recall that *centralizers* in the symmetric group are computable in polynomial time (see, e.g., [10]). However, the complexity of NORM-SYM is open.

QUESTION 18. *Is NORM-SYM in polynomial time? Is there even a subexponential solution?*

If NORM-SYM were in polynomial time, then NORM would reduce to INTER (since $\mathrm{N}_G(H) = \mathrm{N}_{\mathrm{Sym}(\Omega)}(H) \cap G$). In fact, even if polynomial-time algorithms are not available, reductions between the problems are of interest.

QUESTION 19. *Is NORM-SYM polynomial-time reducible to STAB? Is STAB polynomial-time reducible to NORM-SYM?*

Affirmative answers would, respectively, put NORM equivalent to STAB or NORM-SYM.

One of the reasons that Questions 18 and 19 are particularly intriguing is that GRAPH-AUTO *is* polynomial-time reducible to NORM-SYM (as well as to STAB). *Reduction:* Given a graph $\mathcal{G} = (V, E)$, we construct Ω, $G \le \mathrm{Sym}(\Omega)$ and describe an epimorphism $\phi : \mathrm{N}_{\mathrm{Sym}(\Omega)}(G) \to \mathrm{Aut}(\mathcal{G})$. Let $I = \{1, 2, \ldots, 2|V|\}$. Set $\Omega = V \times I \,\dot{\cup}\, E \times \{1, 2\}$ (so $|\Omega| = 2(|V|^2 + |E|)$). For each $v \in V$, let g_v be the involution in $\mathrm{Sym}(\Omega)$ that transposes $(v, 2i-1)$ with $(v, 2i)$, for $1 \le i \le |V|$, and transposes $(e, 1)$ with $(e, 2)$, for every $e \in E$ having endpoint v, while leaving other points fixed; thus g_v moves precisely $2(|V| + \mathrm{degree}(v))$ points. Set $G = \langle \{g_v\}_{v \in V} \rangle$ (an elementary abelian 2-group). Within G the only non-identity elements that move $< 4|V|$ points are the g_v. Hence, permutations in $\mathrm{N}_{\mathrm{Sym}(\Omega)}(G)$ permute the g_v, so that there is an induced homomorphism $\phi : \mathrm{N}_{\mathrm{Sym}(\Omega)}(G) \to \mathrm{Sym}(V)$. Since $\{v, w\} \in E$ iff g_v and g_w move the same point (i.e., the point $(\{v, w\}, 1)$), it is clear that $\phi(G) \le \mathrm{Aut}(\mathcal{G})$. It is straightforward to show that ϕ is surjective. □

Notice that the above reduction involved an elementary abelian 2-group. Thus, Question 18 is interesting *and open* even in this case.

For any finite field $\mathrm{GF}(q)$, there is a natural action of $\mathrm{Sym}(\Omega)$ on $\mathrm{GF}(q)^{\Omega}$ via permutation of coordinates. Then $g \in \mathrm{Sym}(\Omega)$ stabilizes $\Delta \subseteq \Omega$ iff g stabilizes the vector $(a_\omega)_{\omega \in \Omega}$ with $a_\omega = 1$ for $\omega \in \Delta$ and $a_\omega = 0$ otherwise. Thus STAB is polynomial-time reducible to

PROBLEM. VECTOR STABILIZER (VEC-STAB)
INPUT: $G \le \mathrm{Sym}(\Omega)$; *a representation* $\phi : G \to \mathrm{GL}(V)$, *where* V *is a finite dimensional vector space over* $\mathrm{GF}(q)$; $v \in V$
FIND: $G_v = \{g \in G \mid v^g = v\}$.

Here, we assume that V is specified via a basis and ϕ is specified on the given generators of G. By results of [22], VEC-STAB is solvable in polynomial time if G is solvable. ("Polynomial in the input" is taken to be $O((|\Omega| + \dim(V) + \log q)^c)$.) In fact, it is also indicated there that the following is solvable in polynomial time if G is solvable.

PROBLEM. SUBSPACE STABILIZER (SUBSP-STAB)
INPUT: $G \leq \mathrm{Sym}(\Omega)$; *a representation* $\phi : G \to \mathrm{GL}(V)$, *where* V *is a finite dimensional vector space over* $\mathrm{GF}(q)$; *a subspace* $W \leq V$
FIND: $\mathrm{Stab}_G(W) = \{g \in G \mid W^g = W\}$.

QUESTION 20. *Is VEC-STAB in polynomial time for* $G \in \Gamma_d$*? Is SUBSP-STAB in polynomial time for* $G \in \Gamma_d$*?*

VEC-STAB is polynomial-time reducible to SUBSP-STAB. The "obvious" reduction seems to be to stabilize first the 1-dimensional W =Span(v), after which we only need the kernel of a homomorphism $\phi : \mathrm{Stab}_G(W) \to \mathrm{GF}(q)^*$ (multiplicative group). While this is not difficult to complete, we refer instead to the reduction between the corresponding decision problems (VEC-TRANS $\leq$ SUBSP-TRANS) in Section 10.2.

Remark. One can also show that the problem of finding $\mathrm{N}_G(H)$ when $H \lhd\lhd \langle G, H\rangle$ is polynomial-time reducible to SUBSP-STAB.

Another question that arises is whether normality helps for some of these problems. Techniques of [18] (in particular the method of Theorem 8.6 of this present paper), can be used to find $\mathrm{Core}_G(G_v)$ and $\mathrm{Core}_G(\mathrm{Stab}_G(W))$ (where v, W are a vector and subspace, respectively). We wonder, however, about

QUESTION 21. *Given* $G, H \leq \mathrm{Sym}(\Omega)$, *can one find* $\mathrm{Core}_G(\mathrm{N}_G(H))$ *in polynomial time?*

Note that we have found $\mathrm{Core}_G(G \cap H)$, which is the kernel of the right-multiplication-action of G on right cosets of H by G, while $\mathrm{Core}_G(\mathrm{N}_G(H))$ is the kernel of the conjugacy action of G on the conjugates of H by G.

10.2. Decision Questions. The problems of Section 10.1 suggest decision analogues.

Corresponding to NORM:

PROBLEM. CONJUGACY OF GROUPS (CONJ-GROUP)
INPUT: $G, H_1, H_2 \leq \mathrm{Sym}(\Omega)$.
QUESTION: *Is there some* $g \in G$ *such that* $H_1^g = H_2$*?*

As in the STAB $\equiv$ TRANS equivalence, NORM is polynomial-time equivalent to CONJ-GROUP.

The right analogue of NORM-SYM would seem to be

PROBLEM. CONJUGACY IN THE SYMMETRIC GROUP (CONJ-SYM)
INPUT: $H_1, H_2 \leq \mathrm{Sym}(\Omega)$.
QUESTION: *Is there some* $x \in \mathrm{Sym}(\Omega)$ *such that* $H_1^x = H_2$*?*

Here, we do not see the equivalence. While a reduction of CONJ-SYM to NORM-SYM is not difficult, we do not have a reverse reduction. Thus, we ask

QUESTION 22. *Is NORM-SYM polynomial-time equivalent to CONJ-SYM?*

VEC-STAB and SUBSP-STAB are, respectively, polynomial-time equivalent to

PROBLEM. VECTOR TRANSPORTER (VEC-TRANS)
INPUT: $G \leq \mathrm{Sym}(\Omega)$; *a representation* $\phi : G \to \mathrm{GL}(V)$, *where* V *is a finite dimensional vector space over* $\mathrm{GF}(q)$; $v_1, v_2 \in V$
QUESTION: *Is there some* $g \in G$ *such that* $v_1^g = v_2$*?*

and

PROBLEM. SUBSPACE TRANSPORTER (SUBSP-TRANS)
INPUT: $G \leq \mathrm{Sym}(\Omega)$; *a representation* $\phi : G \to \mathrm{GL}(V)$, *where* V *is a finite dimensional vector space over* $\mathrm{GF}(q)$; *subspaces* $W_1, W_2 \in V$
QUESTION: *Is there some* $g \in G$ *such that* $W_1^g = W_2$*?*

The reduction of VEC-TRANS to SUBSP-TRANS is worth noting. While a polynomial-time reduction can be completed along the lines begun in the VEC-STAB to SUBSP-STAB discussion, that would not then be a Karp reduction (yes/no instance mapping to yes/no instance). Here then is another approach: Let (G, ϕ, V, v_1, v_2) be an instance of VEC-TRANS; we may assume that $v_2 \neq 0$. Then G acts naturally on the tensor product $V \otimes V$ (so that $(v \otimes w)^g = v^g \otimes w^g$) and therefore there is an induced action $G \to \mathrm{GL}(V \oplus (V \otimes V))$. Let $W_i = \mathrm{Span}((v_i, v_i \otimes v_i))$, for $i = 1, 2$. We claim that for $g \in G$, $v_1^g = v_2$ iff $W_1^g = W_2$. The only-if direction is clear. Assume $W_1^g = W_2$. Then, for some $c \in \mathrm{GF}(q)$, $(v_1, v_1 \otimes v_1)^g = c(v_2, v_2 \otimes v_2)$, so that $v_1^g = cv_2$ and $(v_1 \otimes v_1)^g = c(v_2 \otimes v_2)$. Thus, $c^2(v_2 \otimes v_2) = (cv_2 \otimes cv_2) = (v_1^g \otimes v_1^g) = (v_1 \otimes v_1)^g = c(v_2 \otimes v_2)$. It follows that $c = 1$, proving the claim. □

We add two more problems that seem of particular interest.

PROBLEM. GROUP ISOMORPHISM (GROUP-ISO)
INPUT: *Cayley tables for groups* G, H.
QUESTION: *Are* G *and* H *isomorphic?*

Here "polynomial in the input" translates to polynomial in $|G|$ (presumably $|G| = |H|$). It is not hard to reduce GROUP-ISO to GRAPH-ISO. (See [24] for a discussion of this and related issues.) But is this problem easier? In particular,

QUESTION 23. *Is GROUP-ISO in polynomial time?*

It can be solved in subexponential $O(|G|^{c+\log_2 |G|})$ time since there is a set of $\leq \log_2 |G|$ generators, and a homomorphism $G \to H$ is determined by the images of the generators. This, however, appears to be the best result known for general groups.

Possibly on the "harder" side is

Problem. PERMUTATION-GROUP ISOMORPHISM
(PERM-GROUP-ISO)
Input: $G, H \leq \mathrm{Sym}(\Omega)$.
Question: *Are G and H isomorphic?*

PERM-GROUP-ISO is in NP: Supposing $G = \langle X \rangle$, one can guess an isomorphism $f : G \to H$ by guessing $f(x)$ for all $x \in X$ and then verifying that f is indeed an isomorphism by checking that $|G| = |H| = |\langle\{(x, f(x)\}_{x \in X}\rangle|$ (the latter being considered as a subgroup of $G \times H$ acting, say, on $\Omega\dot{\cup}\Omega$). □

It is shown in [3] that CONJ-GROUP is polynomial-time reducible to PERM-GROUP-ISO.

To summarize the known relationships, letting "≤" denote "is polynomial-time Karp-reducible to" we have:

$$\begin{array}{rcl}
\text{GROUP-ISO} & \leq & \text{GRAPH-ISO} \\
\text{GRAPH-ISO} & \leq & \text{TRANS} \\
\text{GRAPH-ISO} & \leq & \text{CONJ-SYM} \\
\text{TRANS} & \leq & \text{CONJ-GROUP} \\
\text{TRANS} & \leq & \text{VEC-TRANS} \\
\text{CONJ-SYM} & \leq & \text{CONJ-GROUP} \\
\text{CONJ-GROUP} & \leq & \text{PERM-GROUP-ISO} \\
\text{VEC-TRANS} & \leq & \text{SUBSP-TRANS}
\end{array}$$

(And recall, TRANS ≡ DC-EQ ≡ CONJ-ELT.)

Question 24. *Are there any other reductions between these problems except as implied by the above?*

We do not anticipate seeing a negative answer very soon (as that would necessarily include a proof of P $\neq$ NP), but we believe a search for other relationships could shed additional light on these problems.

References

1. L. Babai, *On the length of subgroup chains in the symmetric group*, Comm. in Alg. 14 (1986), 1729–1736.
2. L. Babai, P. Cameron and P.J. Pálfy, *On the order of primitive groups with restricted nonabelian composition factors*, J. Algebra 79 (1982), 161–168.
3. L. Babai, S. Kannan and E.M. Luks, *A bounded round interactive proof of permutation group non-isomorphism*, in preparation.
4. L. Babai, W.M. Kantor and E.M. Luks, *Computational complexity and the classification of finite simple groups*, Proc. 24th IEEE FOCS (1983), 162–171.
5. L. Babai and L. Kučera, *Canonical labelling of graphs in linear average time*, Proc. 20th IEEE FOCS (1979), 49–46.
6. L. Babai and E.M. Luks, *Canonical labeling of graphs*, Proc. 15th ACM STOC (1983), 171–183.
7. L. Babai, E.M. Luks and Á. Seress, *Computing composition series in primitive groups*, in these Proceedings.
8. L. Babai and S. Moran, *Arthur-Merlin games: A randomized proof system and a hierarchy of complexity classes*. J. Comp. Sys. Sci. 36 (1988), 254–276.

9. G. Butler, *On computing double coset representatives in permutation groups*, in Computational Group Theory, M.D. Atkinson, ed., Academic Press, 1984, 283–290.
10. G. Cooperman, L. Finkelstein and E. Luks, *Reduction of group constructions to point stabilizers*, Proc. 1989 ACM-SIGSAM Intern. Symp. on Symbolic and Algebraic Comp., (1989), 351-356.
11. M.L. Furst, J. Hopcroft and E.M. Luks, *Polynomial time algorithms for permutation groups*, Proc. 21st IEEE FOCS (1980), 36–41.
12. M.R. Garey and D.S. Johnson, Computers and Intractability, A Guide to the Theory of NP-completeness, W.H. Freeman, 1979.
13. O. Goldreich, S. Micali and A. Wigderson, *Proofs that yield nothing but their validity and a methodology of cryptographic protocol design*, Proc. 27th IEEE FOCS, 1986, 174–187.
14. M. Hall, Jr., The Theory of Groups, Macmillan, New York 1959.
15. C.M. Hoffmann, Group Theoretic Algorithms and Graph Isomorphism, Lect. Notes in Comp. Sci. 136, Springer 1982.
16. W.M. Kantor, *Sylow's theorem in polynomial time*, J. Comp. Syst. Sci. 30 (1985) 359-394.
17. W.M. Kantor, *Finding Sylow normalizers in polynomial time* J. Algorithms 11 (1990), 523–563.
18. W.M. Kantor and and E.M. Luks, *Computing in quotient groups*, Proc. 22nd ACM STOC, 1990, 524–563.
19. J. Leon, *Computing automorphisms of combinatorial objects*, in Computational Group Theory, M.D. Atkinson, ed., Academic Press, 1984, 321–335.
20. E.M. Luks, *Isomorphism of graphs of bounded valence can be tested in polynomial time*, J.Comp. Sys. Sci. 25 (1982), 42–65.
21. E.M. Luks, *Computing the composition factors of a permutation group in polynomial time*, Combinatorica 7 (1987), 87–99.
22. E.M. Luks, *Computing in solvable matrix groups*, Proc. 33rd IEEE FOCS, 1992, 111–120.
23. B. McKay, *Nauty Users Guide (Version 1.5)*, Tech. Rep. TR-CS-90-02, Dept. Comp. Sci., Austral. Nat. Univ. 1990.
24. G.L. Miller, *Graph isomorphism, general remarks*, J. Comp. Sys. Sci. 18 (1979), 128–142.
25. G.L. Miller, *Isomorphism of k-contractible graphs, a generalization of bounded valence and bounded genus*, Inform. and Control 56 (1983), 1–20.
26. G.L. Miller, *Isomorphism of graphs which are pairwise k-separable*, Inform. and Control 56 (1983), 21–33.
27. P.M. Neumann, *Some algorithms for computing with finite permutation groups*, Proc. of Groups-St Andrews 1985 (Eds. E.F. Robertson and C.M. Campbell), London Math. Soc. Lect. Note 121, Cambridge U. Press 1987, 59–92.
28. R.C. Read and D.G. Corneil, *The graph isomorphism disease*, J. Graph Theory, 3 (1979), 339–363.
29. C.C. Sims, *Some group-theoretic algorithms*, Springer Lect. Notes in Math. 697 (1978), 108–124.
30. H. Wielandt, Finite Permutation Groups, Acad. Press, New York 1964.
31. V. Zemlyachenko, N. Kornienko, R. Tyshkevich, *Graph isomorphism problem* (Russian), The Theory of Computation I, Notes Sci. Sem. LOMI 118 (1982).

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

E-mail address: luks@cs.uoregon.edu