Lectures on Polynomial-time Computation in Groups

Eugene M. Luks

CIS-TR-90-21
December, 1990

Abstract

These are notes from a series of lectures on polynomial-time computation in permutation groups. The notes are fairly self-contained algebraically and algorithmically. Starting with basic issues, such as testing-membership, much of the polynomial-time library is developed. Instances of group-intersection are indicated, with applications to graph isomorphism. Algorithms are given for finding special subgroups including centers, derived series, Fitting subgroups, composition series.

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

Preface

These notes reflect a series of lectures that I enjoyed offering at Northeastern University during Spring Quarter 1990. They were faithfully and enthusiastically recorded and typeset by Peter Mark and Namita Sarawagi (with occasional solutions and other embellishments by these scribes).

The issue of the lectures was polynomial-time computation in permutation groups. As long as there remain significant outstanding questions about the limits of polynomial-time (in group theory problems as elsewhere) this is a worthy focus on its own. Furthermore, restricting to this issue enables us to bypass both the details of implementation decisions and the rigors of complexity arguments. Consequently, it is feasible, within a short series of lectures to tackle a broad range of computational problems, concentrating on the phenomena that put them into polynomial-time.

The reader will see an early indication of this perspective in the verification of polynomial-time for membership-testing in permutation groups. Our explanation does not offer, or require, a clean statement of Sims's method for the problem (though, to be honest, the audience was aware of that procedure). Instead, we observe that it suffices to know how to compute $|G|$ and we then discuss the key ingredients for that, namely: there is a "short" chain of subgroups from G to 1; using such a chain, it is possible to control the size of generating sets; given generators for a group in the chain, coset representatives and then (Schreier) generators (whose number can be controlled) for the next group can be constructed.

So, from this point of view, there is no need to discuss specific exponents in the timings. Indeed, the computational complexities of these algorithms are not optimal, and even naive speedups are easy to obtain. Similarly, these algorithms are not destined to be implemented as presented. The matters of best worst-case timings and practical efficiency for the same problems would independently comprise worthwhile and dense tutorials.

Useful background for these lectures would be any standard text in group theory together with the first chapter of Wielandt's book "Finite Permutation Groups" (1964, Academic Press). Modulo such references, the group theory herein is self-contained, with the single, notable exception of a call to the classification of finite simple groups in order to complete the final step in a test for simplicity (end of lecture 11).

In case any readers want to follow these in a seminar setting, I must warn that the lectures are not of uniform length. Because of dynamic schedules of attendees, sessions varied in length from 45 minutes to 3 hours. There is also some nonuniformity in mathematical explication. According to the whims of the audience, there was selected elaboration in some topics. Also, the scribes selectively filled in some details.

I thank the College of Computer Science, Northeastern University, for its hospitality. In particular, Larry Finkelstein and Gene Cooperman proposed the visit and zestfully kibbitzed throughout the lectures. Special thanks to Peter and Namita for their prodigious effort in preparing these notes, sometimes based only upon cryptic clues left on the whiteboard.

Eugene M. Luks
Computer and Information Science
University of Oregon

Problem: AUT

Given: A graph X.

Find: $Aut(X)$ the group of automorphisms of the graph X.

Claim 2: ISO $\leq_P$ AUT

Proof: By claim 1 we can assume that X_1 and X_2 are connected graphs. Form the disjoint union $X = X_1 \dot{\cup} X_2$. It is easy to see that $X_1 \cong X_2 \iff \exists f \in Aut(X)$ such that $f(X_1) = X_2$. □

This claim by itself does not yield an efficient algorithm, since the group $Aut(X)$ itself could be exponential in the size of the graph. Therefore, merely listing the elements of $Aut(X)$ may take exponential time. However, it is an easy consequence of Lagrange's theorem that every group G has a generating set of size $\log |G|$. If $G \leq S_n$, then $\log |G| \leq \log n! \leq n \log n$. Hence we can modify the problem AUT to be:

Problem: AUT-GEN

Given: A graph X.

Find: a set of generators for $Aut(X)$ the group of automorphisms of the graph X.

Claim 3: ISO $\leq_P$ finding a set of generators of $Aut(X)$.

Proof: Any generating set must contain an element that flips X_1 and X_2 if some element of $Aut(X_1 \dot{\cup} X_2)$ does. □

Problem: STAB

Given: $A \subseteq Sym(\Omega)$ and $\Delta \subseteq \Omega$

Find: Generators of $\langle A \rangle_{\{\Delta\}} = \{g \in \langle A \rangle \mid \Delta^g = \Delta\}$.

Claim: ISO $\leq_P$ STAB.

Proof: Let $X = (V, E)$ be a graph. Then $Aut(X) \leq Sym(V) = G$ (where $\leq$ means subgroup). G also acts on the set $\binom{V}{2}$ = set of all unordered pairs of vertices. Clearly $E \subseteq \binom{V}{2}$ and $Aut(X) = G_{\{E\}}$ under this action. □

Note: The existing algorithms for STAB, although exponential, usually run efficiently in practice. The complexity of STAB is an open question. The reverse reduction, STAB $\leq_P$ ISO, is still open.

Problem: Set Transporter Problem (Generalization of STAB, Decision version)

Given: $G = \langle A \rangle \leq Sym(\Omega), \Delta_1, \Delta_2 \subseteq \Omega$

Question: Does there exist $g \in G$ such that $\Delta_1{}^g = \Delta_2$?

Exercise: Show that Set Transporter $\leq_P$ STAB.

Hints: The reduction uses analogous techniques to the proof of ISO $\leq_P$ AUT-GEN. The difficulties that seem to arise are in achieving the analogue of reducing to the "connected case", which insured

Notation

$Sym(\Omega)$ is the group of permutations of Ω where $|\Omega| = n$.

$Sym(n)$ is the group $Sym(\Omega)$ where $\Omega = \{1, \ldots n\}$.

G is a group.

Definitions

(1) G **acts on** Ω if $\exists$ a homomorphism $G \longrightarrow Sym(\Omega)$.

(2) A homomorphism $G \longrightarrow Sym(\Omega)$ is a **faithful action** if it is injective. For example, if $G \leq Sym(\Omega)$, then G acts faithfully on Ω.

Examples. Let $G \leq Sym(\Omega)$.
(i) G acts (faithfully) on $\Omega \times \Omega$ where $(\alpha, \beta)^g = (\alpha^g, \beta^g)$ for all $(\alpha, \beta) \in \Omega \times \Omega, g \in G$
(ii) G acts (faithfully) on $\binom{\Omega}{2}$, if $|\Omega| > 2$.
(iii) G acts (faithfully) on 2^Ω where $\Delta^g = \{\delta^g | \delta \in \Delta\}$ for all $\Delta \subseteq \Omega$.

(3) Let G act on Ω, $\omega \in \Omega$, then the **orbit** of ω (under G) $= \{\omega^g | g \in G\}$ and is denoted by ω^G.

(4) Let G act on Ω, then G is **transitive** if it has only one orbit i.e. $\omega^G = \Omega$ for all $\omega \in \Omega$.

(5) Let G act on Ω, then $\Delta \subseteq \Omega$ is a **block** (for G) if $\forall g \in G$, $\Delta^g = \Delta$ or $\Delta^g \cap \Delta = \emptyset$. Δ is a *nontrivial* block if $1 < |\Delta| < |\Omega|$.

Examples: Let $G \leq Sym(\Omega)$.
(i) An orbit is a block.
(ii) If N is a normal subgroup of G then the orbits of N are blocks for G.
Proof: Let $\Delta = \delta^N$ be an orbit of N and $g \in G$. Then $\Delta^g = \delta^{Ng} = \delta^{gN}$ (since N is a normal subgroup of G)$= (\delta^g)^N$ which is the orbit of δ^g.
The orbits form a partition, so either $\Delta \cap \Delta^g = \emptyset$ or $\Delta = \Delta^g$.

Note: Usually blocks are defined only when G is transitive. When we need transitivity, it will be clear in context.

Claim: If Δ is a block for G. Then for $g, h \in G$ either $\Delta^g = \Delta^h$ or $\Delta^g \cap \Delta^h = \emptyset$.

Proof: Since $\Delta^g \cap \Delta^h = (\Delta^{gh^{-1}} \cap \Delta)^h$, therefore $\Delta^g \cap \Delta^h \neq \emptyset$, implies $(\Delta^{gh^{-1}} \cap \Delta) \neq \emptyset$. Since Δ is a block, this means that $\Delta^{gh^{-1}} = \Delta$, which implies $\Delta^g = \Delta^h$. □

Algorithms for Finding Orbits and Blocks

Problem: ORBITS

Given: $G = \langle A \rangle \subseteq Sym(\Omega)$

Find: the orbits of the action of G on Ω

Proposition: There is a polynomial time algorithm for ORBITS.

Proof: Use a transitive closure algorithm (but not merely listing G and writing down ω^G). □

Computation of $\Delta = \omega^G$:

```
$\Delta \leftarrow \{\omega\}$
For all $\delta \in \Delta, a \in A$ do
    If $\delta^a \notin \Delta$ then $\Delta \leftarrow \Delta \cup \{a\}$.
```

Remark: Let G act transitively on Ω. If this action is imprimitive, we can find a block system with blocks of minimal size (e.g., choose the β that leads to smallest block). This also implies that the subgroup fixing a block acts primitively on the points in the block. If the action on the set of blocks is imprimitive, we may repeat the process. We continue until the action of G on the blocks is primitive. We may construct a tree by denoting each block by a vertex, and the children of this vertex are taken to be subblocks that it contains from the previous round. For an intransitive group, we construct such a tree in each orbit, yielding a forest whose leaves comprise Ω. The group G now acts on the entire forest as root-fixing automorphisms. Note also that the subgroup of G that stabilizes any node v in the forest acts primitively on the children of v (*Exercise :* Verify!). This forest is called a **structure forest** for G.

Problem: MEMBER (Permutation Group Membership)

Given: $G = \langle A \rangle \leq Sym(\Omega)$ and $x \in Sym(\Omega)$.

Question: Is $x \in G$?

Remark: It may not be immediately clear that MEMBER is even in NP. The naive nondeterministic algorithm of *guessing* a word in the generators could take exponential time. Consider $G = \langle g \rangle$ where $g = (12)(345)(678910)\ldots$ where successive cycles have lengths of successive primes. If the degree of $G = n$, then order(g) is roughly $\exp(\sqrt{n \log n})$. So the shortest word in the generators of G for most elements has exponential length. Nevertheless we will see that MEMBER has a polynomial time algorithm.

Problem: ORDER (Permutation Group Order)

Given: $G = \langle A \rangle \leq Sym(\Omega)$

Find: $|G|$

Proposition: MEMBER $\leq_P$ ORDER .

Proof: $x \in \langle A \rangle \iff |\langle A \rangle| = |\langle A, x \rangle|$ □

Note: Lagrange's theorem: If $H \leq G$ then $|G| = |H|[G : H]$.

Goal: To show that ORDER is in P.

Let $\omega \in \Omega$ and $H = G_\omega = \{g \in G | \omega^g = \omega\}$ then $|G| = |G_\omega|[G : G_\omega]$ by Lagrange's theorem. As $G_\omega g = G_\omega h \iff \omega^g = \omega^h$, $[G : G_\omega] = |\omega^G|$, that is the right cosets of G_ω correspond to the orbit of ω. Therefore $|G| = |G_\omega||\omega^G|$. We can find $|\omega^G|$ since ORBITS is in P, and in the process of finding the orbit, as noted earlier, we also find a complete set of coset representatives for G in G_ω. To find the $|G|$ we now need to compute $|G_\omega|$. As G_ω is a group which permutes one less point than G we can find its order by continuing the process as before, but to do that we need generators for G_ω.

Definition: Given $H \leq G$ a (right) **transversal** R for H in G is a complete set of (right) coset representatives for H in G.

Lecture #2 **9 April 1990**

Completion of membership algorithm; Algorithms for recognizing and determining the structure of nilpotent and solvable groups; Applications to graph isomorphism

Remark: The basic methodology for efficient membership testing in permutation groups is due to Sims. Furst, Hopcroft, and Luks observed that Sims's techniques lead to a polynomial-time test for membership.

Problem: REDUCE GENERATORS

Given: $H = \langle B \rangle \subseteq Sym(\Omega)$, with $|\Omega| = n$.

Find: A set of $< n^2$ generators for H.

Notation: For $H \leq \Omega = \{\omega_1, \omega_2, \ldots \omega_n\}$. Let $H^{(i)}$ = subgroup of H fixing the first $i-1$ points $= \{ h \in H \mid {\omega_j}^h = \omega_j \; \forall 1 \leq j \leq i-1 \}$. In particular, $H = H^{(1)}$.

Proposition: There is a polynomial time algorithm for REDUCE GENERATORS.

Proof: [Sims] Modify B such that no two elements of B are in the same (right) coset of $H^{(2)}$: For this, if $a, b \in B$ are in the same coset (that is when ${\omega_1}^a = {\omega_1}^b$), then replace b by ba^{-1}. Also, throw away any duplicates in B. Then the modified B contains distinct coset representatives for (some) cosets of $H^{(2)}$ in $H^{(1)}$ and (maybe) some elements in $H^{(2)}$. Repeat the same process for $B \cap H^{(2)}$, that is if $a, b \in B \cap H^{(2)}$ are in the same coset of $H^{(3)}$ then replace b by ba^{-1}. Repeat this process for each $B \cap H^{(i)}$. As $H^{(n-1)} = 1$, this process will stop and number of elements in B will be at most $[H^{(1)} : H^{(2)}] + [H^{(2)} : H^{(3)}] + \ldots + [H^{(n-2)} : H^{(n-1)}] < n^2$. $\square$

Remark: This capability to keep the number of generators "small" is fundamental to procedures in this lecture and later. For, it guarantees that we can keep the size of the intermediate outputs under control as we routinely concatenate polynomial-time procedures. We will routinely assume this procedure is invoked as needed.

Proposition: There exists a polynomial time algorithm for ORDER.

Proof: Let ω_1 be any point not fixed by G. As noted earlier, $|G| = |G^{(1)}| = |G^{(2)}|[G^{(1)} : G^{(2)}]$, where $[G^{(1)} : G^{(2)}] = |\omega_1^G|$. We may appeal to a recursive computation of $|G^{(2)}|$ as $G^{(2)}$ moves fewer points than G. (Note here the implicit use of REDUCE GENERATORS. Without it the number of schreier generators, as we pass through successive groups $G^{(i)}$, could grow exponentially). $\square$

Corollary: MEMBER is in P.

Proof: We saw earlier that MEMBER $\leq_P$ ORDER. $\square$

Problem: SUBGROUP?

Given: $G = \langle A \rangle \leq Sym(\Omega)$ and $H = \langle B \rangle$.

Question: Is H a subgroup of G ?

Note: G' is the unique smallest normal subgroup of G such that G/G' is abelian.

Proposition: Let $G = \langle A \rangle$ then $G' = \langle [A,A] \rangle^G$.

Proof: Clearly $\langle [A,A] \rangle \le G'$ and as $G' \triangleleft G$, therefore $\langle [A,A] \rangle^G \le G'$. Let $\pi : G \longrightarrow G/\langle [A,A] \rangle^G$ be the canonical homomorphism. Then $G/\langle [A,A] \rangle^G = \pi(G)$ is abelian since it is generated by $\pi(A)$ and $[\pi(A), \pi(A)] = \pi([A,A]) = 1$. Therefore, $G' \le \langle [A,A] \rangle^G$ (by the note above).

Problem: COMMUTATOR SUBGROUP

Given: $G = \langle A \rangle \subseteq Sym(\Omega)$.

Find: G', the commutator subgroup of G.

Proposition: COMMUTATOR SUBGROUP is in P.

Proof: By the previous proposition, $G' = H^G$ where $H = \langle \{ [a,b] \mid a,b \in A \} \rangle$. The generators for H can be computed in polynomial time from the (polynomial number of) generators of G. The proposition follows as NORMAL CLOSURE is in polynomial time. □

Definition: Let G be a group and G' it commutator subgroup . Then the commutator subgroup of G' is denoted by G''. The **derived series** of G is the following chain of groups.

$$G \supseteq G' \supseteq G'' \supseteq G''' \supseteq \dots$$

(Continue until stable). If the derived series terminates at $\{1\}$ then G is called **solvable**.

Problem: DERIVED SERIES

Given: $G = \langle A \rangle \subseteq Sym(\Omega)$.

Find: The derived series of G.

Proposition: DERIVED SERIES is in P.

Proof: By repeated application of COMMUTATOR SUBGROUP (and REDUCE GENERATORS as needed), we can compute the derived series. The algorithm stops when the chain stabilizes. □

Problem: SOLVABLE

Given: $G = \langle A \rangle \subseteq Sym(\Omega)$.

Question: Is G solvable?

Proposition: SOLVABLE is in P .

Proof: Find the derived series for G. If it terminates in $\{1\}$ then G is solvable. □

Definition: Let G be a group. The **lower central series** of G is the following chain of subgroups.

$$G = L^0(G) \ge L^1(G) \ge L^2(G) \dots$$

where $L^0(G) = G$ and $L^i(G) = [G, L^{i-1}(G)] = \langle \{ [g,h] \mid g \in G, h \in L^{i-1}(G) \} \rangle$. If the lower central series terminates in $\{1\}$ then G is called **nilpotent**.

Proposition: $L^i(G) \lhd G$ for all i. Moreover if $G = \langle A \rangle$ and $L^{i-1}(G) = \langle B \rangle$, then $L^i(G) = \langle \{ [a,b] \mid a \in A, b \in B \} \rangle^G$.

Proof: Similar to the proof of $G' = \langle \{ [a,b] \mid a,b \in A \} \rangle^G$. □

Problem: LOWER CENTRAL SERIES

Given: $G = \langle A \rangle \subseteq Sym(\Omega)$.

Find: The Lower Central series of G.

Proposition: LOWER CENTRAL SERIES is in P .

Proof: Clear from above. □

Problem: NILPOTENT

Given: $G = \langle A \rangle \subseteq Sym(\Omega)$.

Question: Is G nilpotent?

Proposition: NILPOTENT is in P .

Proof: Find the lower central series for G. If it terminates in $\{1\}$ then G is nilpotent. □

Remark: It can be seen (by induction) that $L^i(G) \supseteq G'^{(i)}$. Hence G is nilpotent $\Rightarrow$ G is solvable.

Definitions: Let G be a group.

(i) The center of G is the subgroup $Z(G) = \{ g \in G \mid gg' = g'g, \forall g' \in G \}$.

(ii) The **upper central series** of G is the following chain of subgroups.

$$1 = Z^0(G) \leq Z^1(G) \leq Z^2(G) \leq \dots$$

where $Z^0(G) = 1$, $Z^1(G) = Z(G)$, and $Z^i(G) = \{g \in G \mid [G,g] \subseteq Z^{i-1}(G)\}$. An equivalent description of $Z^i(G)$ is as follows. $Z^0(G) = 1$ and $Z^i(G)/Z^{i-1}(G) = Z(G/Z^{i-1}(G))$.

iii A **central series** in G is a chain of *normal* subgroups

$$G = G_0 \geq G_1 \geq \dots \geq G_r = 1$$

for which $[G, G_{i-1}] \subseteq G_i$ for each i.

(iv) If $H \leq G$ then H is said to be **subnormal in** G if there exists a chain

$$H = L_0 \leq L_1 \leq \dots \leq L_m = G$$

where each $L_{i-1} \lhd L_i$. It is denoted by $H \lhd \lhd G$

(v) For $H < G$, the **normalizer** $H \in G$ is $N_G(H) = \{g \in G \mid g^{-1}Hg = H\}$.

Proof: If we can find a complete set of right coset representatives for H in G, then we can find Schreier generators for H. A naive search for these coset representatives works:

Algorithm:

```
    $R = \{1\}$
    { apply generators (on the right) to elements of $R$}
    For each $r \in R, a \in A$
        if $ra \notin Hr'$ for any $r' \in R$ then $R \leftarrow R \cup \{ ra \}$
```

Note: Testing membership of $ra \in Hr'$ can be performed by testing $rar'^{-1} \in H$, for which we have a polynomial time algorithm. The above algorithm runs in time proportional to $|A||R|^2$·running time of the membership test for H. Note that, if $rar'^{-1} \in H$, then it is a schreier generator for H.

An application to graph isomorphism.

Definition: Let CG_b be the class of vertex-colored graphs of color multiplicity $\leq b$, b a fixed constant, i.e. there are at most b vertices of a given color.

Exercise: Before reading further, give a polynomial time non-group-theoretic algorithm for testing isomorphism of two graphs in CG_2.

Polynomial Time Algorithm for ISO of graphs in CG_b

We reduce (using the observations in the first lecture) ISO for graphs $X_1, X_2 \in CG_b$ to finding automorphism groups for graphs in CG_{2b}, (namely, find $Aut(X)$, where $X = X_1 \dot{\cup} X_2$ and X_1, X_2 are connected). As noted in the first lecture, if we view X as uncolored, $Aut(X)$ is precisely the set stabilizer $Sym(V)_{\{E\}}$, where $E \subseteq \binom{V}{2}$, $X = (V, E)$, and there is no polynomial time algorithm for set stabilizer. However, in the current context, the problem is more constrained. We must not only stabilize E, but also each color class. Let $V = C_1 \cup \ldots \cup C_k$ be a decomposition of V into disjoint color classes. Then $Aut(X) \leq G = Sym(C_1) \times \ldots \times Sym(C_k)$. We can easily find generators for G. Furthermore, if we let $E_{i,j} = \{e \in E \mid$ one of the endpoints of e is a vertex of C_i, the other a vertex of $C_j\}$, and we let $H = G_{E_{i,j}}$, i.e. the subgroup of G (as before, viewed as acting on $\binom{V}{2}$) that fixes the set of edges from color class C_i to color class C_j, then surely $Aut(X) \leq H \leq G$. We can find generators for H using the algorithm for GRS since $[G : H] =$ the number of images of $C_i - C_j$ edges $= |E_{i,j}| \leq 2^{|C_i \times C_j|} \leq 2^{(2b)^2}$ (a crude overestimate) and we can test membership in H, so H is polynomial time recognizable. Having found generators for H, continue to find generators for the subgroup of H that stabilizes edges between another pair of color classes. (This can be done using GRS by the same argument as above). Repeat this process until all pairs of color classes have been exhausted. Then H converges to $Aut(X)$.

Remark: The above argument is essentially due to Babai, who described a random (Las Vegas) algorithm for the problem. Furst, Hopcroft, and Luks observed that Sims's methods obviate the randomness.

Intersection of permutation groups.

Problem: INTERSECTION

Given: $G = \langle A \rangle, H = \langle B \rangle \leq Sym(\Omega)$.

Find: $G \cap H$.

Proof: Exercise.

Solution to exercise: The test for subnormality is constructive in that it inserts the intermediate groups in $H = L_m \triangleleft \cdots \triangleleft L_1 = \langle G, H \rangle$. Since $H \cap L_i$ normalizes L_{i+1}, repeated application of the above algorithm for INTERSECTION-N yields generators for all $H \cap L_i$.□

Claim: There is a polynomial time algorithm for STAB-NIL.

For this we will:

1. Reduce STAB-NIL to STAB-P (set stabilizer for p-groups).
2. Solve STAB-2 and briefly indicate how this solution generalizes to STAB-P.
3. For this, we will have to investigate the structure of Sylow p-subgroups of $Sym(\Omega)$.

Proof: (of 1.) Without loss of generality, we may assume G is a p-group. (Recall that if $H \leq G$, nilpotent, then $H = \langle P \cap H \mid P$ the Sylow p-subgroup of G, for each p dividing $|G|\rangle$, so that $G_{\{\Delta\}} = P_{1_{\{\Delta\}}} \times \ldots \times P_{k_{\{\Delta\}}}$.) □

Form a structure forest for G

Focus, for the moment on any node, v, in this forest. Lift G's action to the entire forest. By construction, G_v acts primitively on the children of v.

Claim: G a primitive p-group $\Rightarrow$ G is cyclic of order p and acts on a set of size p.

Proof: G primitive on $\Omega \Rightarrow G_\omega$ is a maximal subgroup of G. Maximal subgroups of p-groups have index p. The index of a point stabilizer, $[G : G_\omega]$ is precisely the size of the orbit containing ω, which, in this case, is all of Ω, since G is transitive on Ω. Therefore, G is a primitive p-group acting transitively on a set of size p, so G must be cyclic of order p.

Corollary: The structure forest for a p-group consists of complete p-ary trees.

Sylow p-subgroups of $Sym(\Omega)$

For simplicity, consider first the case $p = 2$. To construct a Sylow 2-subgroup, build a forest of complete binary trees whose leaves are points of Ω, subject to the criteria that the trees in this forest be as "large" as possible (in the sense that no two trees have equal height, since those could be joined to form a single larger tree). [Call such forests *maximal*.] Then the group of all automorphisms of this forest induces on Ω *precisely* a Sylow 2-subgroup. Note that if $n = b_d \ldots b_1 b_0$ is the binary representation of n, then for each $b_i = 1$ there will be a complete binary tree of height i in this forest.

Note: If we have one Sylow 2-subgroup of $Sym(\Omega)$, we "know" them all, since all Sylow 2-subgroups are conjugate. (It is easy to see that conjugacy in $Sym(n)$ amounts to renaming the points: the permutations σ and $\sigma^g = g^{-1}\sigma g$ have the same cycle structure, in fact, the cycles of σ^g are obtained from σ by replacing each $i \in \{1 \ldots n\}$ by $i^{g^{-1}}$).

One can check that the construction above indeed gives a Sylow 2-subgroup by comparing its order with the order with the largest power of 2 dividing $n!$. The order of the group may be computed as the product of the sizes of the automorphism group of each tree in the forest.

Exercise: (1) Find the order of the automorphism group for a complete binary tree of height m. (2) Show that the above construction yields a Sylow 2-subgroup of $Sym(n)$.

Let G be a 2-group $\leq Sym(\Omega)$. We can embed G in a Sylow 2-subgroup of $Sym(\Omega)$ (i.e. find a Sylow 2-subgroup of $Sym(\Omega)$ containing G) by finding the structure forest for G, extending it to a maximal complete binary forest, and considering the automorphism group of this forest. See [Aho, Hopcroft, Ullman] for a description of polynomial time algorithms for testing isomorphism of trees. From the methodology presented there, it is possible to develop an algorithm for finding automorphism groups of trees (*Exercise!*).

problem of finding *color automorphisms* in a 2-group, that is, finding the subgroup fixing each of several "colored" subsets; clearly the problem is polynomial-time equivalent to STAB).

(Editorial comment by lecturer: Note-takers felt the above reduction could be omitted since the lecture did closely approximate the discussion in the cited paper. That was not the case for STAB-NIL; see remarks at start of next lecture).

Note: (i) STAB reduces to finding centralizers of involutions. (ii) In the above reduction, $h \notin G$ (considering $G \leq Sym(\bar{\Omega})$). If $h \in G$ then $C_G(h)$ is called **Internal Centralizer.** We can reduce STAB to INTERNAL CENTRALIZER, by finding $C_{\langle G,h \rangle}(h)$. As the set $\mathcal{B} = \{\{\omega, \omega'\} \mid \omega \in \Omega\}$ is a block system for $\langle G, h\rangle$, each generator of $C_{\langle G,h\rangle}(h)$ induces a permutation in $Sym(\mathcal{B}) \equiv Sym(\Omega)$. These induced permutaions give generators for $C_G(h)$.

Definition: Let $g \in Sym(\Omega)$, then the **graph** of g is $\Delta_g = \{(\omega, \omega^g) \mid \omega \in \Omega\} \subseteq \Omega \times \Omega$.

Let $Sym(\Omega)$ act on $\Omega \times \Omega$ in the natural way: $(\alpha, \beta)^g = (\alpha^g, \beta^g)$.

Facts: Let $g, h, h_1, h_2 \in Sym(\Omega)$,

(i) $\Delta_{h_1} = \Delta_{h_2} \iff h_1 = h_2$.

(ii) $(\Delta_h)^g = \Delta_{h^g}$.

(iii) $gh = hg \iff (\Delta_h)^g = \Delta_h$.

Proof:

(i) Clear, by the definition of *graph*.

(ii) $(\Delta_h)^g = \{(\omega^g, \omega^{hg}) \mid \omega \in \Omega\} = \{(\pi, \pi^{g^{-1}hg}) \mid \pi \in \Omega\} = \Delta_{g^{-1}hg} = \Delta_{h^g}$.

(iii) $gh = hg \iff g^{-1}hg = h \iff \Delta_{g^{-1}hg} = \Delta_h$ [by (i)] $\iff (\Delta_h)^g = \Delta_h$ [by (ii)].

Remark: CENTRALIZER $\leq_P$ STAB. By (iii) above, $C_G(h) = G_{\{\Delta_h\}}$. Hence by a previous proposition, CENTRALIZER $\equiv_P$ STAB.

Remark: Since CENTRALIZER is as hard as STAB, and so at least as hard as ISO, we will not attempt to solve this in our attack on CENTER. The critical observation that will put CENTER in polynomial-time is that the solution to the problem is a normal subgroup. In fact, we will solve the more general problem of finding the centralizer of a normalized group.

Exercise: If $G, H \leq Sym(\Omega)$ and G normalizes H, then $C_G(H) \triangleleft G$.

Solution: For any g, H, note that $g^{-1}C_G(H)g = C_G(g^{-1}Hg)$.

Problem: CENTRALIZER-N

Given: $G = \langle A \rangle$, $H = \langle B \rangle \leq Sym(\Omega)$, where G normalizes H.

Find: $C_G(H) = \{g \in G \mid gh = hg, \forall h \in H\}$.

Proposition: CENTRALIZER-N is in P.

Proof: For each $b \in B$ form $\Delta_b \subseteq \Omega \times \Omega$. Then $C_G(H) = \{g \in G \mid \Delta_b{}^g = \Delta_b, \forall b \in B\}$. By the exercise above $C_G(H) \triangleleft G$.

Define an equivalence relation $\sim$ on $\Omega \times \Omega$ as follows: for $\alpha, \beta \in \Omega \times \Omega$, $\alpha \sim \beta \iff \alpha, \beta$ lie in exactly the same Δ_b's for $b \in B$. Let the induced partition Π consist of equivalence classes $\Pi_1, \Pi_2, \ldots \Pi_r$; then $C_G(H) = \{g \in G \mid \Pi_i{}^g = \Pi_i, \forall 1 \leq i \leq r\}$. Now, for any $x \in G$, the cells in the partition $\Pi^x = \{\Pi_1^x . \Pi_2^x, \ldots \Pi_r^x\}$ are stabilized by $x^{-1}C_G(H)x = C_G(H)$. Hence $C_G(H)$ is the subgroup of G fixing the classes in the common refinement, $\{\Pi_i \cap \Pi_j^x \mid \Pi_i \cap \Pi_j^x \neq \emptyset, 1 \leq i, j \leq r\}$, of Π, Π^x. Thus, it follows similarly that $C_G(H)$ is the stabilizer of the cells in the coarsest refinement $\tilde{\Pi}$ of Π that is compatible with the action of G, i.e., such that $\tilde{\Pi}^x = \tilde{\Pi}$ for $x \in G$.

Timing: Let $T(G)$ denote the time required to solve SNS for G. Assuming we can find a proper normal subgroup N of G in polynomial time, we have $T(G) = T(N) + n^c$, (for some fixed constant c) if N does not have a solvable normal subgroup, and otherwise $T(G) = T(N) + T(C_G(N)) + n^c$. The key observation is that we only have to consider $C_G(N)$ when SNS returns "no" for N, and in that case, $C_G(N) \cap N = Z(N) = 1$ (since $Z(N)$ is a solvable normal subgroup of N). Thus, if the second recursive call to SNS is invoked, we know that $|G| \geq |N||C_G(N)|$. It follows that $T(G) = \mathcal{O}(\log(|G|)n^c)$, and hence SNS is in P.

Next time: Special case of Proper Normal Subgroup, that is when G has a solvable normal subgroup.

Discussion and Proof: It is not difficult to extend basic (Sims's) membership testing algorithm to this case of partial permutations (though we may have obscured the issue with a particularly high-level approach to MEMBER in lectures 1,2). However, it seems worth observing another approach that reduces the problem directly to point stabilizer (i.e., the case when f is the identity on Δ). This approach is reminiscent of the reductions such as ISO to finding automorphism group and of SET-TRANSPORTER to STAB (*Exercise:* Explore that!). Also, it is particularly useful in a parallel (class NC) approach to the partial permutation problem for the ordinary (sequential) membership test is not available, though pointwise set stabilizers are.

We assume that we have an algorithm for pointwise set stabilizers. Note first that if $g \in G$ is any extension of f then the set of all such extensions is given by $G_\Delta g$.

The group $G \times G$ acts naturally on $\Omega \times \Omega$ (via $(\alpha, \beta)^{(g,h)} = (\alpha^g, \beta^h)$). Define $x \in Sym(\Omega \times \Omega)$ by $(\alpha, \beta)^x = (\beta, \alpha)$ and let $H = \langle G \times G, x\rangle$ (thus, H is the wreath product $G \wr Z_2$). Let $\bar{\Delta} = \{(\delta, f(\delta)) \mid \delta \in \Delta\}$. Find $L = H_{\bar{\Delta}}$.

(1) If $L \leq G \times G$ then there is no $g \in G$ extending f,

else take $y \in L - G \times G$; then $yx = (g, h) \in G \times G$ and

(2) g is an extension of f. □

Exercise: Prove (1) and (2) above.

Returning to main track -

Definition: A subgroup $H \leq G$ is a **characteristic** subgroup if H is invariant under all automorphisms of G, i.e. for all $\sigma \in Aut(G), \sigma(H) = H$.

Exercise:

(i) A characteristic subgroup $H \leq G$ is a normal subgroup of G.

(ii) For any group G, G' the commutator subgroup of G, is a characteristic subgroup.

(iii) If K is characteristic in H and $H \triangleleft G$ then $K \triangleleft G$.

(iv) If K is characteristic in H and H is characteristic in G then K is characteristic in G.

Recall from the previous page that we have reduced PNS-S to the case where G is primitive. If G has a solvable normal subgroup $1 \neq H \triangleleft G$, then G has an abelian normal subgroup (the last non-trivial term in the derived series of H is an abelian subgroup, and by the above exercise, it is normal in G).

Definition: $G \leq Sym(\Omega)$ is **regular** if G is transitive and $\forall \omega \in \Omega, H_\omega = 1$.

Exercise:

(i) G is regular $\iff \forall \alpha, \beta \in \Omega\ \exists!\ g \in G$ such that $\alpha^g = \beta$. $(\Rightarrow |G| = |\Omega|)$.

(ii) G transitive and abelian $\Rightarrow G$ regular, and $C_{Sym(\Omega)}(G) = G$.

Hence, if G is primitive and has an abelian normal subgroup H, then H is transitive (orbits of normal subgroups of G are blocks for G) and hence regular (by exercise above). H *must* be proper, otherwise G would be a regular primitive group, hence of prime order, a case excluded in the problem statement. If we actually had generators for H, we'd be done.

Lectures #6, 7 **30 April, 1 May 1990**

Algorithms for computing radical and fitting subgroups

Definitions: Let G be any group.

(i) The **Radical** of G is the maximal solvable normal subgroup of G, denoted by $Rad(G)$.

(ii) The **Fitting subgroup** of G is the maximal nilpotent normal subgroup of G, denoted by $Fit(G)$.

(iii) The p-**Core** is the maximal normal p-subgroup of G, denoted by $Fit_p(G)$ or $O_p(G)$.

Remarks: (i) The subgroups defined above are all unique, as the subgroup generated by two normal solvable/nilpotent/p-subgroups of G is again a normal solvable/nilpotent/p-subgroup of G.
(ii) The term "radical", for maximal solvable normal subgroup is not standard.

Note:

(i) $O_p(G) \leq Fit(G) \leq Rad(G)$.

(ii) $Fit(G) = \prod_{p\,prime} O_p(G)$.

(iii) $O_p(G) = \bigcap_{g \in G} P^g$, where P is a Sylow p-subgroup of G.

Remark: Finding $O_p(G)$ via (iii) would require use of classification of finite simple groups, which is presently essential for polynomial-time computation of Sylow subgroups [Kantor].

Problem: RADICAL

Given: $G = \langle A \rangle \leq Sym(\Omega)$

Find: $Rad(G)$.

Claim: There is a polynomial time algorithm for RADICAL.

Proof: Since we know how to find *a* solvable normal subgroup H of G (invoke SNS with input G), one might suppose we could recursively invoke SNS with G/H. However, we have no faithful permutation representation of G/H.

Instead, we proceed as follows. Let $1 \neq H \triangleleft G$, with H abelian (if K is the solvable normal subgroup returned by SNS with input G, then let H be the last nontrivial term in the derived series for K). Let $\Delta_1, \Delta_2, \ldots, \Delta_r$ be the orbits of H, and $H^{\Delta_1}, H^{\Delta_2}, \ldots, H^{\Delta_r}$ be the constituents of H (the **constituent** of H on Δ_i, denoted H^{Δ_i} is the group induced by H on Δ_i). H^{Δ_i} is transitive and abelian, so it is regular on Δ_i, and $|H^{\Delta_i}| = |\Delta_i|$. Let $\Sigma = \dot{\bigcup}_{1 \leq i \leq r} H^{\Delta_i}$ (*disjoint union*). Then $|\Sigma| = |\Omega|$, and G acts on Σ as follows: let $g \in G$, and $h_i \in H^{\Delta_i}$, and suppose $\Delta_i{}^g = \Delta_j$ (the orbits of H are blocks for G), then $h_i{}^g$ is $g^{-1}h_ig$ restricted to Δ_j (note that the identity of H^{Δ_i} is mapped to the identity of H^{Δ_j}). Let $G \xrightarrow{\pi} Sym(\Sigma)$ denote this action, and $K = Ker(\pi) \triangleleft G$. Then $H \leq K$ (as H fixes Δ_i and commutes with H^{Δ_i}). K stabilizes Δ_i (since, in the action on Σ it fixes the identity of H^{Δ_i}) and K^{Δ_i} centralizes H^{Δ_i} so $K^{\Delta_i} = H^{\Delta_i}$ (H^{Δ_i} is its own centralizer in $Sym(\Delta_i)$). Hence K is an abelian normal subgroup of G, and $G/K \hookrightarrow Sym(\Sigma)$. Now, equipped with this faithful action of G/K, we can recursively find the $Rad(G/K)$. Since $Rad(G/K) = Rad(G)/K$, we finish by forming the pullback of $Rad(G)/K$ in G (see lec. 5, p. 1).

Exercise: Verify that the above algorithm runs in polynomial time.

Let $K = O_p(G)$, then $[H, K] \leq H$ and is normal in G (since both H, $K \triangleleft G$). By the minimality of H, $[H, K] = 1$ or $[H, K] = H$. Since H is nilpotent, $[H, K] = 1$. $\Box$

If we can find a minimal normal p-subgroup H of G (in polynomial time) then the action π of G on the orbits of H, has a p-subgroup K as kernel, and so $O_p(\pi(G)) = O_p(G)/K$, and we can recurse to find $O_p(G)$.

Given any non-trivial normal p-subgroup, we can find an abelian normal p-subgroup and then an elementary abelian normal subgroup (*Exercise. Verify that!*). Thus we assume H is an elementary abelian normal p-subgroup. Then G acts on H, by viewing H as a vector space and the actions are linear transformations. Hence, finding minimal normal subgroups of a group G is reduced to finding an irreducible subspace for a set of linear transformations of a vector space over a finite field. The latter was an open problem for some time and was proposed by Kantor, to complete this approach to $O_p(G)$. This problem was solved by Rónyai.

An amusing aspect of the above version is that, ignoring Rónyai's ultimate contribution, Kantor had reduced finding a maximal normal p-subgroup to finding a minimal normal p-subgroup.

Version 3′

This is merely a hybrid, not a different approach. We observe that method in the algorithm for RADICAL enables us the avoid the problem of finding minimal normal p-subgroups (for which Ronyai has to introduce considerable machinery, including a constructive version of the Wedderburn theory for rings). An alternative approach to the second case ($p \mid |H|$) above is to use the action π on the set of constituents of an abelian normal p-subgroup H. In this case, the kernel, K, of π is an abelian, normal p-subgroup of G and so $K \leq O_p(G)$ and $O_p(\pi(G)) = O_p(G)/K$.

Remark: There is another algorithm due to P. Neumann.

Remark: $Soc(G) = M_1 \times \ldots \times M_s$, where the M_i's are minimal normal in G. The socle is therefore a direct product of simple groups.

Note, finally

Lemma: Any two minimal normal subgroups of a group centralize each other.

Proof: If M_1, M_2 are distinct minimal normal subgroups in G then $[M_1, M_2] \triangleleft G$, and $[M_1, M_2] \leq M_1 \cap M_2 < M_1$. By the minimality of M_1, $[M_1, M_2] = 1$. (This also follows directly from the first Proposition in this lecture). □

Note that above M_1 and M_2 "commute" not just in the weak sense that $M_1M_2 = M_2M_1$, but additionally that each of these groups centralizes the other.

Socles of primitive permutation groups

Let $G \leq Sym(\Omega)$ be a primitive group. Let $N = Soc(G)$. Recall that any normal subgroup of a primitive group is transitive. If $N = M_1 \times \ldots \times M_s$, (each M_i minimal normal in G) then each M_i acts transitively on Ω. If $s > 1$, then each M_i commutes with each of the other M_j's $(i \neq j)$, So for example, M_1 and M_2 are commuting, transitive groups.

Remark: We will soon see that $s \leq 2$.

Definition: A group $K \leq Sym(\Omega)$ is called **semiregular** if its point stabilizers, K_ω, for $\omega \in \Omega$, are trivial. (So a group is regular iff it is transitive and semiregular).

So, if K is semiregular and $\alpha, \beta \in \Omega$ then there is at most one element in G mapping α to β (for all such elements lie in the same right coset of K_α).

Lemma: If K is centralized by a transitive group H, then K is semiregular.

Proof: $K_\alpha = {K_\alpha}^h = K_{\alpha^h} = K_\beta$, if $\alpha^h = \beta$. Since H is transitive, for each $\beta \in \Omega, \exists h \in H$ such that $\alpha^h = \beta$. It follows that $K_\alpha = K_\beta \; \forall \beta \in \Omega$, i.e.$K_\alpha = 1$ □

We use this to show

Proposition: If M_1, M_2 are commuting transitive groups then M_1 and M_2 are both regular and $M_1 = C_{Sym(\Omega)}(M_2)$, $M_2 = C_{Sym(\Omega)}(M_1)$.

Proof: By the previous lemma, M_1 and M_2 are both regular. Since $C_{Sym(\Omega)}(M_1)$ commutes with the transitive group M_1, it contains, for any $\alpha, \beta \in \Omega$, at most one element mapping α to β. But there is already such an element in $M_2 \leq C_{Sym(\Omega)}(M_1)$. Hence $M_1 = C_{Sym(\Omega)}(M_2)$. Similarly $M_2 = C_{Sym(\Omega)}(M_1)$. □

From this discussion we immediately get

Corollary: If M_1, M_2 are distinct minimal normal subgroups of a primitive group $G \leq Sym(\Omega)$ then M_1 and M_2 are both regular and $M_1 = C_{Sym(\Omega)}(M_2)$, $M_2 = C_{Sym(\Omega)}(M_1)$.

Since M_2 could have been *any* minimal normal subgroup distinct from M_1, it follows immediately that

Corollary: A primitive group has at most two minimal normal subgroups.

In fact, if there are exactly two minimal normal subgroups, we can say more.

Let G be a group. G acts on itself via right multiplication: $\rho : G \to Sym(G), g^\rho : h \mapsto hg$. This action is called the **right regular action** of G on itself. We can also define the left **regular** that corresponds to $n^g = g^{-1}ng$, are the same. The former point is ω^{ng}, the latter is $\omega^{n^g} = \omega^{g^{-1}ng}$. Note that g fixes ω, so $\omega^{ng} = \omega^{g^{-1}ng}$, which is exactly what we needed to show.

Since G_ω's action on Ω is precisely G_ω's action on N by conjugation, G_ω's acts faithfully on Ω as a group of linear transformations. (Faithful because any element of G_ω in the kernel of that action would have to centralize N, but N is an abelian, transitive and is therefore its own centralizer and is regular, so the centralizer of N in G_ω is $N_\omega = 1$.) So $G_\omega \hookrightarrow GL(d,p)$.

G is the semidirect product $G = G_\omega N$, and we now know that G_ω is a subgroup of the set of linear transformations of Ω and N is the full group of translations of Ω. Therefore $G \hookrightarrow AGL(d,p)$, the affine group of a vector space of dimension d over a field of characteristic p.

In fact we can say even more: G_ω acts *irreducibly* in Ω. For if there were a proper invariant subspace of Ω, (bearing in mind that Ω and N are identified), this subspace would constitute a normal subgroup of G, properly contained in N, contradicting the minimality of N.

Remark: *All* primitive groups with an abelian socle have this structure. To build examples of primitive groups with abelian socles, pick d, p, form $\Omega = {Z_p}^d$, and include in a set of generators enough translations to generate the full translation group, and enough linear transformations to guarantee an irreducible action on Ω.

Exercise: For G primitive in $Sym(n)$ with abelian socle, verify that $|G| \leq n^{1+\log n}$.

Point stabilizers in socles of primitive groups with no regular normal subgroup

Let $G \leq Sym(\Omega)$ be primitive, $N = Soc(G) = T_1 \times \ldots \times T_r$, where the T_i's are isomorphic, nonabelian simple groups, and N is the unique minimal normal subgroup. Then G acts by conjugation on N by permuting the set $\{T_1, \ldots, T_r\}$. This action must be transitive, since otherwise a nontrivial orbit would generate a normal subgroup of G properly contained in N, contradicting the minimality of N.

For a point $\omega \in \Omega$, $G = G_\omega N$ but since G has no regular normal subgroup, $N_\omega = G_\omega \cap N \neq 1$. Hence although G factors as $G = G_\omega N$, G is not the semidirect product of these two subgroups. Since N acts trivially on $\{T_1, \ldots, T_r\}$, and G acts transitively, G_ω must act transitively on $\{T_1, \ldots, T_r\}$ as well.

Consider the group N_ω. This is a G_ω-invariant subgroup, and $1 < N_\omega < N$.

Claim: N_ω is a *maximal* G_ω-invariant subgroup of N.

Proof: Suppose there is a G_ω-invariant subgroup H such that $N_\omega \leq H \leq N$. Since G_ω normalizes H, $G_\omega H$ is a group, and $G_\omega \leq G_\omega H \leq G$. Since G is primitive, G_ω is a maximal subgroup, so either $G_\omega H = G_\omega$, or $G_\omega H = G$. Consider the first possibility: $G_\omega H = G_\omega$ implies $H \leq G_\omega$. Since we also know that $H \leq N$, we find that $H \leq G_\omega \cap N = N_\omega$. So in this case, we find $H = N_\omega$. Now consider the second possibility: $G_\omega H = G$ implies $H \triangleleft G$ (it is normalized by both G_ω and H), but, as N is the unique minimal normal subgroup, $N \leq H$. Since, by hypothesis, $H \leq N$, we must have $H = N$. □

Case III

We define a case III group to be a primitive group G with no regular normal subgroup and $\pi_i(N_\omega) = T_i$ for all i. Our analysis of this case will depend critically on the following (folklore) lemma, to the proof of which we devote the remainder of the lecture.

Lemma : Let $G \hookrightarrow H = T_1 \times \ldots \times T_k$ be a subdirect product (i.e. $\pi_i(G) = T_i$), where each T_i is a simple nonabelian group. Then after some rearrangement of the factors, we may write

$$H = (T_1 \times \ldots \times T_{i_1}) \times (T_{i_1+1} \times \ldots \times T_{i_2}) \times \ldots \times (T_{i_{(k-1)}+1} \times \ldots \times T_{i_k})$$

such that $T_{i_j+1} \cong T_{i_j+2} \cong \ldots \cong T_{i_{j+1}}$ for all $0 \leq j \leq k-1$ (where $i_0 = 0$), and

$$G = diag(T_1 \times \ldots \times T_{i_1}) \times diag(T_{i_1+1} \times \ldots \times T_{i_2}) \times \ldots \times diag(T_{i_{(k-1)}+1} \times \ldots \times T_{i_k}),$$

(i.e. after appropriate identifications, $G = \{(\alpha \ldots \alpha)(\beta \ldots \beta) \ldots (\kappa \ldots \kappa)\}$).

Proof: Define a relation on $\{1, \ldots, r\}$ such that $i \sim j \iff \forall g \in G, \pi_i(g) = 1 \Rightarrow \pi_j(g) = 1$, i.e. $ker(\pi_i) \leq ker(\pi_j)$, or equivalently, $\pi_j(ker(\pi_i)) = 1$.

Claim: $\sim$ is an equivalence relation. *Proof:* Reflexivity and transitivity are immediate, so we verify symmetry. Suppose $i \sim j$. We need to show $j \sim i$, i.e. $L = \pi_i(ker(\pi_j)) = 1$. $ker(\pi_j) \triangleleft G \Rightarrow L = \pi_i(ker(\pi_j)) \triangleleft \pi_i(G) = T_i$. Suppose $L \neq 1$. Then we must have $L = T_i$, since T_i is simple. Let $g \in G$ such that $\pi_j(g) \neq 1$ (this is possible since G is a subdirect product). Then there exists some $h \in ker(\pi_j)$ such that $\pi_i(h) = \pi_i(g)$ (since $L = T_i$). Now $\pi_i(gh^{-1}) = 1$, but $\pi_j(gh^{-1}) \neq 1$. But this contradicts our assumption that $i \sim j$. □

Next we show that the equivalence classes of this relation correspond to diagonal blocks of G. Let $\{B_1, \ldots, B_k\}$ be the equivalence classes. For $i = 1, \ldots, k$, let $D_i = \{g \in G \mid \pi_j(g) = 1 \ \forall j \notin B_i\}$.

Claim: $\pi_s(D_i) = T_s, \forall s \in B_i$. *Proof:* Let $s \in B_i$. $D_i \triangleleft G \Rightarrow \pi_s(D_i) \triangleleft T_s$. Since T_s is simple, it suffices to show that $\pi_s(D_i) \neq 1$. Pick $g \in G$ such that $\pi_s(g) \neq 1$ and $|\{l \mid \pi_l(g) \neq 1\}|$ is minimal. It suffices to show that $g \in D_i$ (since then $1 \neq \pi_s(g) \in \pi_s(D_i)$). Suppose, to the contrary, that $g \notin D_i$. Then there is some $j \notin B_i$ such that $\pi_j(g) \neq 1$. Let $g_s = \pi_s(g)$. There exists $t_s \in T_s$ such that $[g_s, t_s] \neq 1$ ($Z(T_s) = 1$ since T_s is nonabelian simple). Since $s \in B_i$ and $j \notin B_i$, we know that $s \not\sim j$. This implies that $\pi_s(ker(\pi_j)) = T_s$ and so there exists $h \in ker(\pi_j)$ such that $\pi_s(h) = t_s$. We will complete the proof by showing that the element $[g, h]$ contradicts our choice of g, i.e. $\pi_s([g, h]) \neq 1$ and $|\{l \mid \pi_l([g, h]) \neq 1\}| < |\{l \mid \pi_l(g) \neq 1\}|$. Observe that $\pi_s([g, h]) = [\pi_s(g), \pi_s(h)] = [g_s, t_s] \neq 1$. Since $\pi_m(g) = 1 \Rightarrow \pi_m([g, h]) = 1$, we have $\mathcal{U} = \{l \mid \pi_l([g, h]) \neq 1\} \subseteq \mathcal{V} = \{l \mid \pi_l(g) \neq 1\}$. However $\pi_j([g, h]) = [\pi_j(g), \pi_j(h)] = [\pi_j(g), 1] = 1$, so $\mathcal{U} \subset \mathcal{V}$ $(j \in \mathcal{V} \setminus \mathcal{U})$. □

By this claim we know that $\forall s \in B_i \ \pi_s : D_i \to T_s$ is surjective, and $D_i \cap ker(\pi_s) = 1$. So for all $s \in B_i$ we have $\pi_s : D_i \to T_s$ is an isomorphism, and $D_i = diag(\prod_{s \in B_i} T_s)$.

All that remains is to show that $G = D_1 \times \ldots \times D_k$. Clearly $G \geq D_1 \times \ldots \times D_k$. For $g \in G$, we need to show that $g \in D_1 \times \ldots \times D_k$. For each i, $1 \leq i \leq k$, pick an $s_i \in B_i$ and $h_i \in D_i$ such that $\pi_{s_i}(h_i) = \pi_{s_i}(g)$. Then $\pi_{s_i}(g(h_1 h_2 \ldots h_k)^{-1}) = 1$ for all i, so $\pi_l(g(h_1 h_2 \ldots h_k)^{-1}) = 1$ for all $l \in B_i$ and for all i, which implies that $g(h_1 h_2 \ldots h_k)^{-1} = 1$ i.e. $g = (h_1 h_2 \ldots h_k) \in D_1 \times \ldots \times D_k$. This completes the proof of the lemma. □

Problem: PNS-1

Given: $G = \langle A \rangle \leq Sym(\Omega)$

Find: One of the following:

(i) Generators for a proper normal subgroup of G.

(ii) The report " G is simple".

(iii) A faithful action of G on a domain of size at most $|\Omega|/2$.

Note : Clearly repeated application of PNS-1 when the output is of type (iii) yields an algorithm for PNS. Hence PNS $\leq_P$ PNS-1 and the running time for PNS is $\log(|\Omega|)$ times the running time of PNS-1. It would suffice (in PNS-1) to produce in (iii) a faithful action of G on any domain of size smaller than $|\Omega|$, in which case the running time for PNS would be a factor of $|\Omega|$ slower than the running time of PNS-1. However it is useful for application in the parallel algorithms to observe that the size of the domain is halved.

Claim : PNS-1 (and hence PNS) can be solved in polynomial time.

Proof : Let $G \leq Sym(\Omega)$ be given. We may assume that $G \leq Sym(\Omega)$ has no orbits of size 1.

Algorithm (for PNS-1)

```
Step 1:
    If $G$ is not transitive then
        $\Delta \leftarrow$ the second largest orbit of $G$'s action on $\Omega$. (Therefore, $|\Delta| \leq |\Omega|/2$.)
        Let $G \xrightarrow{\pi} Sym(\Delta)$ be the induced action on $\Delta$.
        $K \leftarrow Ker(\pi)$.
        If $K \neq 1$ then {$K$ is a proper normal subgroup of $G$ since $|\Delta| > 1$.} output $K$ .
        Else output $G \overset{\pi}{\hookrightarrow} Sym(\Delta)$ where $|\Delta| \leq |\Omega|/2$.
    Else {$G$ is transitive.}
Step 2:
        If $G$ is not primitive then
            $\mathcal{B} \leftarrow$ a non-trivial block system. (Then, $|\mathcal{B}| \leq |\Omega|/2$.)
            Let $G \xrightarrow{\pi} Sym(\mathcal{B})$ be the induced action on the blocks.
            $K \leftarrow Ker(\pi)$.
            If $K \neq 1$ then {$K$ is a proper normal subgroup of $G$ since $G$ is transitive.} output $K$ .
            Else output $G \overset{\pi}{\hookrightarrow} Sym(\mathcal{B})$ where $|\mathcal{B}| \leq |\Omega|/2$.
        Else {$G$ is primitive.}
```

We may now assume that G is primitive on Ω.

Remark : We could look for proper normal subgroups by computing G' or $Z(G)$, and if they were not proper we could assume $G = G'$ etc. but this would not get us too far.

```
Step 3:
            If $|G| = |\Omega| = n$ then output "G is simple of prime order"
```

$|G| = |\Omega|$ implies that G is regular (acts on itself). Moreover G is primitive so it has no proper subgroups (the cosets of a proper subgroup would form a non-trivial block system for this regular action) and hence it is of prime order.

Lecture #11 **21 May 1990**

An algorithm for testing simplicity – contd.

Problem: PNS-1

Given: $G = \langle A \rangle \leq Sym(\Omega)$

Find: One of the following:

(i) Generators for a proper normal subgroup of G.

(ii) The report "G is simple".

(iii) A faithful action of G on a domain of size at most $|\Omega|/2$.

Recall from last lecture, that the problem of finding a proper normal subgroup PNS, reduced to PNS-1. The goal is to show that PNS-1 can be solved in polynomial time. In the previous lecture we began describing a polynomial time algorithm for PNS-1. We saw that if G is not primitive, or if G is primitive and has a normal subgroup of index $\leq |\Omega|$, or has a regular normal subgroup then the algorithm (described so far) would have terminated with an appropriate output.

The rest of this lecture completes the description of the algorithm for PNS-1.

We can now assume that G is primitive on Ω, does not have a proper normal subgroup of index $\leq n = |\Omega|$, and does not have a regular normal subgroup. In the previous lectures #9,#10, we classified primitive permutation groups into 3 cases. Since G does not have a regular normal subgroup, G is not in Case I under this classification.

Algorithm (for PNS-1) contd:

Step 6:

```
Fix $\alpha \in \Omega$.
For all $\beta, \gamma, \delta \in \Omega$ do
    $H \leftarrow \langle G_{\alpha\beta}, G_{\gamma\delta} \rangle$.
    If $G = H$ reject $\{\beta, \gamma, \delta\}$.
    Else $\Delta \leftarrow$ a minimal block system for $G$'s transitive action on the cosets of $H$.
        Let $G \xrightarrow{\pi} Sym(\Delta)$ be this primitive action.
        $K \leftarrow ker(\pi)$.
            If $K \neq 1$ output the proper normal subgroup $K$.
            Else if $|\Delta| \leq |\Omega|/2$
                output $G \overset{\pi}{\hookrightarrow} Sym(\Delta)$.
                Else reject $\{\beta, \gamma, \delta\}$.
```

Claim: Suppose the action of G on Ω is in Case II (see lecture #9) with $r > 1$ or Case III (see lecture #10) with $l > 1$. If step 6 is reached, the algorithm will halt there.

Proof: In these two cases we have

(a) $Soc(G) = N_1 \times \ldots \times N_m$, with $m > 1$

(b) G acts (by conjugation) transitively on $\{N_1, \ldots, N_m\}$

(c) For any $\alpha \in \Omega$, $Soc(G)_\alpha = (N_1)_\alpha \times \ldots (N_m)_\alpha$

(d) $(N_i)_\alpha$ is a proper normal subgroup of N_i.

there exists α, β such that $\alpha \neq \beta$ and $\alpha^{t_1} = \beta$. If no kernel was found in the action of G on Γ then in the faithful primitive action of G on Δ, t_1 has a fixed point (say $\hat{\alpha}$), namely the block containing $\{\alpha, \beta\}$. Therefore this action is a case II action and $\Delta = (|T_1|/|(T_1)_{\hat{\alpha}}|)^r$. Since G's action on Ω is a case III action we have $|\Omega| = |T_1|^{r-1}$. *Subclaim:* $|\Delta| \leq |\Omega|/2$. *Proof:* Suppose not. Therefore $|T_1|^r/|(T_1)_{\hat{\alpha}}|^r \geq |T_1|^{r-1}/2$. This implies that $|T_1| \geq |(T_1)_{\hat{\alpha}}|^r/2 \geq 2^{r-1}$. Now $n = |T_1|^{r-1} \geq 2^{(r-1)(r-1)} \geq r!$. Since G acts transitively on $\{T_i\}_{1 \leq i \leq r}$ the kernel of this action has index $\leq r! \leq n$. This contradicts the fact that G has no proper normal subgroup of index $\leq n$. □.

Step 8:

Output "G is simple (nonabelian)"

By the above, if Step 8 is reached, the action of G must fall into Case II with $r = 1$. Thus $Soc(G) = T_1$ is a nonabelian simple group. Let $G \xrightarrow{\pi} Aut(T_1)$ be the natural map ($T_1 \triangleleft G$). Then $ker(\pi) = 1$, for otherwise $ker(\pi) = C_G(T_1) \neq 1 \Rightarrow C_G(T_1) \cap T_1 = 1$ (since T_1 is nonabelian simple) $\Rightarrow T_1$ is not the unique minimal normal subgroup, which contradicts $Soc(G) = T_1$. Hence we have $T_1 \cong Inn(T_1) \hookrightarrow G \hookrightarrow Aut(T_1)$. Also, $G = G'$ (otherwise as remarked in the previous lecture, G would have a normal subgroup of index $\leq n$). By the Schreier conjecture (which is proved due to the classification of simple groups), $Aut(T_1)/Inn(T_1)$ is solvable. Hence $G = T_1$ (since G/T_1 is solvable and $(G/T_1)' = G/T_1$). *Note that is is the only place in the algorithm where the classification of finite simple groups is needed.*

This proves the claim (in the previous lecture) that PNS-1 can be solved in polynomial time. □

Proof: Repeated application of MAXIMAL NORMAL gives an algorithm for COMPOSITION SERIES. □

The polynomial time library for permutation groups

There is somewhat more to be said about the expanding toolkit for polynomial-time computation in permutation groups. For a summary, with references, of the status of the field as of Spring 1990, we refer the reader to [W.M. Kantor and E.M. Luks *Computing in quotient groups,*, Proc. 22nd ACM Symposium on Theory of Computing, May 1990, pp. 524–534]. This is available as a Technical Report [CIS-TR-90-07] from the Department of Computer and Information Science, University of Oregon, Eugene, OR 97403. (The TR has a footnote updating two of the open problems of the STOC version).