Technical Report NU-CCS-90-16
College of Computer Science
Northeastern University

Lectures on Polynomial-Time Computation in Groups

Eugene M. Luks
University of Oregon

Notes recorded and prepared by
Peter D. Mark and Namita Sarawagi

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

Lecture #1 **2 April 1990**

Computational Complexity and Permutation Groups

Issue: "Efficient" Computation in permutation groups. For sequential computation, we take the point of view that "Efficient" ≡ Polynomial time

Motivation: Problems such as

Problem: Graph Isomorphism: ISO

Given: $X_1 = (V_1, E_1)$ and $X_2 = (V_2, E_2)$ graphs.

Question: Is $X_1 \cong X_2$?

Remarks: ISO is clearly in NP, (one can guess the isomorphism, then easily verify in polynomial time), but is it NP-complete? Or is it in P? Both questions are open. These questions have added significance since ISO is the only problem whose complexity is still unresolved of the three leading candidates for problems that might be of complexity intermediate between P and NP-complete, as proposed by Karp in his famous paper on NP completeness. The other two such problems were Linear Programming (since shown to be in P) and Primality Testing (since shown to be in P - assuming the Riemann Hypothesis).

There appears strong evidence that ISO is not NP-complete (e.g., else, by results of Goldreich, Micali, and Wigderson, the polynomial-time hierarchy would collapse to $\Sigma_2^p = \Pi_2^p = AM$). On the other hand, it has stubbornly resisted efforts to bring it into P. We will point out that ISO easily reduces to certain natural questions about the complexity of permutation group problems, motivating efforts to put define the limits of polynomial-time computation in groups. Group theory also provides a natural setting for the ISO problem. The algebraic methods extend, substantially, the class of graphs for which isomorphism can be tested in polynomial time. Also, ISO is seen to be typical of a class of algebraic problems with similarly unresolved complexity status.

Efficiency for ISO

In practice, there are naive algorithms that work efficiently on most graphs. In fact, it has been shown that isomorphism can be tested *on average* in linear time [Babai and Kučera]; this remains the case even for regular graphs [Kučera].

Polynomial time reductions

The following discussion illustrates a close connection between graph isomorphism and permutation group algorithms.

Problem: ISO-C

Given: $X_1 = (V_1, E_1)$ and $X_2 = (V_2, E_2)$ connected graphs.

Question: Is $X_1 \cong X_2$?

Claim 1: ISO $\leq_P$ ISO-C

Proof: Just compare pairs of connected components.

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 $f(X_1) = X_2$. $\square$

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. $\square$

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. $\square$

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 that the only automorphisms of the disjoint union would fix or switch the two graphs. Form the disjoint union of two copies of $\bar{\Omega} = \Omega\dot{\cup}\Omega'$. $G \times G$ acts on $\bar{\Omega}$ but we need also to be able to switch Ω and Ω' without introducing too many extraneous permutations. Let $t \in Sym(\bar{\Omega})$ switch corresponding points in Ω and Ω' and set $\bar{G} = \langle G \times G, t\rangle$ (i.e., $\bar{G}$ is a wreath product $G \wr Z_2$). The set to be stabilized? Clearly that ought to be $\bar{\Delta} = \Delta_1\dot{\cup}\Delta_2'$. So how does $\bar{G}_{\{\bar{\Delta}\}}$ solve the problem?

Remark: Babai and Moran have shown that, like ISO, if the decision version of Set Transporter were NP-complete, then the polynomial-time hierarchy would collapse to ${\Sigma_2}^P = {\Pi_2}^P = AM$.

Exercise: Show the following problems are equivalent to ISO:

Problem: #ISO

Given: Two graphs, X_1, X_2

Find: The number of distinct isomorphisms from X_1 to X_2

Problem: ISO-X1

Given: Two graphs, X_1, X_2

Find: An isomorphism from X_1 to X_2

Problem: ISO-Xall

Given: Two graphs, X_1, X_2

Find: All isomorphisms from X_1 to X_2

Hints: To reduce ISO-X1 to ISO: first find a pair of corresponding points by attaching unique "gadgets" to different pairs of points and calling ISO; when corresponding points are located, leave these gadgets in place and search for second pair of corresponding points, etc. For ISO-Xall (and AUT-GEN), use similar techniques to find *right transversals* in the *point stabilizer chain* in Aut(X) (see later this lecture and next) .

Exercise: State and prove the analogous set of equivalences for STAB.

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 $\Delta \leftarrow \Delta \cup \{a\}$.
```

This algorithm is clearly in polynomial time since there are at most $|A||\Delta|$ iterations, and each iteration is in polynomial time.

Note: We can also keep track for each $\delta \in \omega^G$ an element g such that $\omega^g = \delta$.

Problem: BLOCKS

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

Find: a nontrivial block (or a block system) for G, if one exists.

Definition: If (G is transitive and) there are no nontrivial blocks for G (in its action on Ω) then G is **primitive**.

Exercise: A transitive group $G \leq Sym(\Omega)$ is primitive $\iff$ for some $\omega \in \Omega$ the subgroup G_ω fixing ω is a maximal subgroup of G. From this, or otherwise, show that if G_ω is maximal for some $\omega \in \Omega$ then it is maximal for all $\omega \in \Omega$.

Proposition: If $\Sigma \subseteq \Omega$ then there is a unique minimal block containing Σ.

Proof: Suppose Δ_1, Δ_2 are both blocks containing Σ. Then $\Delta_1 \cap \Delta_2$ also is a block containing Σ. For, if $(\Delta_1 \cap \Delta_2)^g \cap (\Delta_1 \cap \Delta_2) \neq \emptyset$, then $({\Delta_1}^g \cap \Delta_1) \neq \emptyset$ and so ${\Delta_1}^g = \Delta_1$, and by the same reasoning, ${\Delta_2}^g = \Delta_2$. Hence $(\Delta_1 \cap \Delta_2)^g = (\Delta_1 \cap \Delta_2)$, so $\Delta_1 \cap \Delta_2$ is a block. Since the intersection of two blocks is again a block, consider now the intersection of all blocks containing Σ. This must be the unique minimal block containing Σ. □

Problem: MINIMAL BLOCK (MB)

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

Find: the minimal block containing $\{\alpha, \beta\}$.

Proposition: There is a polynomial time algorithm for MB.

Proof: [Sims] Form a graph $X = (V, E)$ where $V = \Omega$ and $E = \{\alpha, \beta\}^G$. Note that this graph may be easily formed in polynomial time (E is an orbit of the action of G on $\binom{\Omega}{2}$). The connected components of X form a block system, since $G \leq Aut(X)$ as each $g \in G$ induces a permutation of V such that $(\gamma, \delta) \in E \iff (\gamma^g, \delta^g) \in E$. Hence $g \in G$ permutes the connected components, and so they form a block system.

Moreover, $B =$ the component containing α and β is in fact the required minimal block. Suppose it isn't. Let $B_1 \subset B$ be the minimal block where α and $\beta \in B_1$. Since B is a connected component, there exists an edge connecting a vertex in B_1 to a vertex in $B \setminus B_1$. As every edge is the image of the edge $\{\alpha, \beta\}$, without loss of generality, $\exists g \in G$ such that $\alpha^g \in B_1$, and $\beta^g \notin B_1$. Now $\alpha^g \in B_1 \cap {B_1}^g$ implies that $B_1 \cap {B_1}^g \neq \emptyset$. Also $\beta^g \in {B_1}^g \setminus B_1$. Therfore $B_1 \neq {B_1}^g$. Hence B_1 is not a block, a contradiction.

The result now follows from the observation that the connected components of a graph may be found with a standard transitive closure algorithm in polynomial time.□

Proposition: BLOCKS $\leq_P$ MB

Proof: Fix a point $\alpha \in \Omega$. For each $\beta \in \Omega$ apply the algorithm for MB given above to the points α, β. □

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.

Schreier generators

Theorem: Given $G = \langle A \rangle$ and $H \leq G$, and R a right transversal for H in G. Let $B = \{r_1 a {r_2}^{-1} | r_1, r_2 \in R, a \in A\} \cap\ H$. Then B generates H.

Proof of Theorem: For each $r_1, r_2 \in R$ and $a \in A$, then $r_1 a {r_2}^{-1} \in H$ iff if r_2 is the coset representative of the coset $H(r_1 a)$ in R. (In particular, for finite G, $|B| \leq |R||A|$.)

For any $r_1 \in R$ and $a \in A$ if r_2 is chosen as above, then $(r_1 a {r_2}^{-1}) \in B$ and $r_1 a = (r_1 a {r_2}^{-1}) r_2$. Therefore, $RA \subseteq BR \subseteq \langle B \rangle R$. This implies that $(\langle B \rangle R)A \subseteq \langle B \rangle \langle B \rangle R = (\langle B \rangle R)$. That is $(\langle B \rangle R)$ is closed with respect to right multiplication by A.

Also, for any $r_2 \in R$ and $a \in A$ we can choose $r_1 \in R$ such that $(r_1 a {r_2}^{-1}) \in B$. (In this case we choose r_1 as the coset representative of the coset $H(r_2 a^{-1})$ in R.) Therefore $r_2 a^{-1} = (r_2 a^{-1} {r_1}^{-1}) r_1$ implies that $RA^{-1} \subseteq B^{-1} R \subseteq \langle B \rangle R$. Thus $(\langle B \rangle R) A^{-1} \subseteq \langle B \rangle \langle B \rangle R = (\langle B \rangle R)$. That is $(\langle B \rangle R)$ is closed with respect to right multiplication by A^{-1}. *Note:* this paragraph is needed only if $|G|$ is infinite.

The two closure properties above, imply that $\langle B \rangle R = G$. In particular, $H \subseteq \langle B \rangle R$. For $r \in R$ such that $r \notin H$, $H \cap \langle B \rangle r = \emptyset$, as $\langle B \rangle \leq H$ (by definition of B). Therefore if r_0 is the unique element of $R \cap H$, then $H \subseteq \langle B \rangle r_0$. Hence $H = H{r_0}^{-1} \subseteq \langle B \rangle$. This implies $H = \langle B \rangle$. □

Remark: The generators in B are called the **Schreier generators** for H. For $G \leq Sym(n)$ and H the subgroup fixing a point, the Schreier generators can be found in polynomial time, as $|R||A|$ elements need to be computed where $|R| \leq n$.

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 $[H^{(1)} : H^{(2)}] + [H^{(2)} : H^{(3)}] + \ldots + [H^{(n-2)} : H^{(n-1)}] < n^2$. □

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). □

Corollary: MEMBER is in P.

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

Problem: SUBGROUP?

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

Question: Is H a subgroup of G ?

Proposition: SUBGROUP? is in P.

Proof: If each $b \in B$ is a member of G (invoke the algorithm for MEMBER) then H is a subgroup of G. □

Problem: NORMAL?

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

Question: Is $H \triangleleft G$?

Definition: If $A, B \in Sym(\Omega)$ then $B^A = \{\, b^a = a^{-1}ba \mid a \in A, b \in B \,\}$.

Proposition: NORMAL? is in P.

Proof: It can be seen easily, that it is enough to check that $B^A \subseteq H$. Therefore invoke MEMBER $|B||A|$ times. □

Definition: Given $H \leq G$, the **normal closure of H in G** is the smallest normal subgroup of G containing H and is denoted by $\langle H^G \rangle$.

Problem: NORMAL CLOSURE

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

Find: $\langle H^G \rangle$ the normal closure of H in G.

Remark: In the above problem, and in future problem statements, we assume that finding a particular group means finding a set of generators for that group. Similarly, unless specified otherwise, we assume that groups are input via generators. (In the above instance it is convenient to have names for the generating sets.)

Proposition: There is a polynomial time algorithm for NORMAL CLOSURE.

Algorithm:

```
Let $B' \leftarrow B$ and $K \leftarrow \langle B' \rangle = H$
While $\exists b \in B', a \in A$ such that $b^a \notin K$ do
    $B' \leftarrow B' \cup \{b^a\}$ and $K \leftarrow \langle B' \rangle$
Return $(B')$
```

Each time a new generator is added to B', the size of the group is (at least) doubled. Therefore this algorithm takes polynomial time to complete. □

Remark: The observation that increasing subgroup chains in $Sym(n)$ have polynomially-bounded length is fundamental in establishing polynomial time in many algorithms. We will not always recall it explicitly. By the way, the naive $\log(n!)$ chain-length bound obtained from Lagrange's theorem has been improved to $O(n)$ by Babai.

Remark: For ease of notation, from now on, we sometimes write H^G for $\langle H^G \rangle$.

Definition: Let G be a group. For $g, h \in G$ the element $g^{-1}h^{-1}gh$ is denoted by $[g, h]$ and is called a **commutator of** G. The **commutator subgroup** of G or the **derived group** of G is the subgroup generated by all the commutators of G. It is denoted by G' or $[G, G] = \langle \{[g, h] \mid g, h \in G\} \rangle$.

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 $\langle [A, A] \rangle \leq G'$ and as $G' \triangleleft G$, therefore $\langle [A, A] \rangle^G \leq G'$. 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, $G' \leq \langle [A, A] \rangle^G$ (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.

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

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 $N_G(H) = \{g \in G \mid g^{-1}Hg = H\}$.

Exercise: Let P be a Sylow p-subgroup of G, and H be any subgroup containing $N_G(P)$ then $N_G(H) = H$.

Solution: Given $P \leq N_G(P) \leq H$, we want to show that $N_G(H) = H$. Clearly $H \leq N_G(H)$. Let $g \in N_G(H)$, then $g^{-1}Hg = H$. Therefore $P, g^{-1}Pg \leq H$ are Sylow p-subgroups of H. By the Sylow theorem, there exists $h \in H$ such that $g^{-1}Pg = h^{-1}Ph$. So, $(gh^{-1})^{-1}P(gh^{-1}) = P$, implies $gh^{-1} \in N_G(P) \leq H$. Hence $g \in H$. □

Exercise: Show the following are equivalent for a finite group G.

(i) G is nilpotent.

(ii) G has a central series.

(iii) Every subgroup of G is subnormal.

(iv) Every proper subgroup is properly contained in its normalizer.

(v) Every maximal subgroup is normal.

(vi) The Sylow subgroups of G are normal in G.

(vii) G is a direct product of p-groups.

(viii) The upper central series terminates at G.

Solution:

$(i) \Rightarrow (ii)$ Clear from definitions.

$(ii) \Rightarrow (iii)$ Let $G = G_0 \geq G_1 \geq \ldots \geq G_r = 1$ be a central series for G. Consider the chain $G = G_0H \geq G_1H \geq \ldots \geq G_rH = H$. It suffices to show G_iH is normal in $G_{i-1}H$ for each i. Since H normalizes both G_i and H, it suffices to show G_{i-1} normalizes G_iH, for which it suffices to show $h^x \in G_iH$ for $h \in H$, $x \in G_{i-1}$. But $h^x = h[h,x] \in h[G, G_{i-1}] \subset hG_i = G_ih$.

$(iii) \Rightarrow (iv)$ Let $H < G$. Then H is subnormal in G and there is a series $H = H_0 \triangleleft H_1 \triangleleft \ldots \triangleleft H_n = G$ If i is the least positive integer such that $H \neq H_i$, then $H = H_{i-1} \triangleleft H_i$ and $H_i \leq N_G(H)$.

$(iv) \Rightarrow (v)$ If M is a maximal subgroup of G, then $M < N_G(M)$, so by maximality $N_G(M) = G$ and $M \triangleleft G$.

$(v) \Rightarrow (vi)$ Let P be a Sylow p-subgroup. If P is not normal then $N_G(P) < G$, and there exists a maximal subgroup M of G such that $N_G(P) < M$. M is normal by hypothesis. By the above exercise, $N_G(M) = M$ which contradicts the normality of M.

$(vi) \Rightarrow (vii)$ There is exactly one Sylow p-subgroup for each prime p since all such are conjugate. The product of all the Sylow subgroups is clearly direct and it must equal G.

$(vii) \Rightarrow (viii)$ It suffices to show (vii) holds for a finite p-group G for $Z^i(G_1 \times G_2) = Z^i(G_1) \times Z^i(G_2)$. But then it suffices to show that $Z(P) \neq 1$ for a finite p-group $P \neq 1$ (P acts on underlying set P by conjugation; orbit sizes must divide $|P|$; at least one orbit has size 1, namely $\{1\}$; conclude that some other orbit $\{x\}$ has size 1; $x \in Z(P)$).

$(viii) \Rightarrow (i)$ Suppose $1 = Z^0(G) < Z^1(G) < Z^2(G) < \ldots < Z^r(G) = G$ is the upper central series. From the definitions, and by induction on i, it is immediate that $L^i(G) \leq Z^{r-i}(G)$, for $i = 0, 1, \ldots, r$. In particular, $L^r(G) = 1$.

Exercise: Use condition (vii) to find a polynomial time test of nilpotency.

Solution: Given $G = \langle A \rangle$. First compute $|G|$. Let $|G| = {p_1}^{n_1}{p_2}^{n_2} \dots {p_r}^{n_r}$, where the p_i's are distinct primes. Let $q_i = |G|/{p_i}^{n_i}$ for $1 \leq i \leq r$.

If $r = 1$ then condition (vii) is satisfied (by Sylow's theorem) and G is nilpotent.

$G = \langle A \rangle$ is a product of its Sylow p_i subgroups $\iff$ each Sylow p_i-subgroup is normal $\iff$ $P_i = \langle \{ a^{q_i} \mid a \in A \} \rangle$ is a normal subgroup and $|P_i| = {p_i}^{n_i}$.

Since, NORMAL? and ORDER are in P, the above paragraph gives another polynomial time test of nilpotency. □

Problem: SUBNORMAL?

Given: H, G groups with $H \leq G$

Question: Is $H \triangleleft \triangleleft G$?

Problem: SUBNORMAL SERIES (SS)

Given: $H \triangleleft \triangleleft G$

Find: Generators for each group L_i in a normal tower from H to G.

Fact: $H \triangleleft \triangleleft G \iff H \triangleleft \triangleleft H^G$.

Proof: ($\Leftarrow$) Immediate, since $H^G \triangleleft G$. ($\Rightarrow$) Let $H \triangleleft L_1 \triangleleft \dots \triangleleft G$ be a subnormal series for H in G. Then intersect each group in this series with H^G and obtain a subnormal series for H in H^G.□

Claim: SUBNORMAL? and SS are in P.

Proof: The above observation suggests the following subnormality test and construction of a subnormal series. Let $L_1 = H^G$. Inductively, let $L_{i+1} = H^{L_i}$. Stop when $L_{i+1} = L_i$. If $L_i = H$ then $H \triangleleft \triangleleft G$, otherwise not.

Definition: A group $H \leq Sym(n)$ will called **recognizable** if there is polynomial-time test (for some fixed polynomial in n) for membership in H.

Example: H could be the subgroup of G that stabilizes some subset.

Definition: For $H \leq G \leq Sym(n)$, we shall say that H has **small index** in G if $[G : H]$ is polynomially bounded (for some fixed polynomial in n).

Problem: GENERATORS FOR A RECOGNIZABLE SUBGROUP OF SMALL INDEX (GRS)

Given: Generators for G, and a specification of H, a recognizable subgroup of small index.

Find: Generators for H.

Claim: GRS is in P.

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: $G \cap H$.

Proposition: STAB $\leq_P$ INTERSECTION

Proof: As, $G_{\{\Delta\}} = Sym(\Omega)_{\{\Delta\}} \cap G$ and we can find generators for $Sym(\Omega)_{\{\Delta\}}$, where $\Delta \subseteq \Omega$, easily. □

Definition: Let $G, H \leq Sym(\Omega)$, then G **normalizes** H if $\forall g \in G, H^g = g^{-1}Hg = H$.

Note: Given $G = \langle A \rangle$ and $H = \langle B \rangle$, G normalizes $H \iff B^A = \{ b^a \mid b \in B, a \in A \} \subseteq H$. Therefore we can check if G normalizes H in polynomial time.

Exercise:

(i) If L normalizes H and $M \leq L$ then $[LH : MH] \leq [L : M]$.

(ii) If $M \leq L$ and N any group, then $[L \cap N : M \cap N] \leq [L : M]$.

Solution:

(i) If $L = \cup x_i M$ then $LH = \cup x_i MH$, therfore $[L : M] \geq [LH : MH]$

(ii) Distinct cosets of $M \cap N$ in $L \cap N$ correspond to distinct cosets of M in L.

Problem: INTERSECTION-N

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

Find: $G \cap H$.

Proposition: INTERSECTION-N is in P.

Proof: We have the following series of subgroups.

$$G = G^{(1)} \geq G^{(2)} \geq \ldots \geq G^{(n-1)} = 1$$

$$GH = G^{(1)}H \geq G^{(2)}H \geq \ldots \geq G^{(n-1)}H = H \text{ (since } G \text{ normalizes } H\text{).}$$

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

We'll find $G \cap H$, by successively finding $G \cap G^{(i)}H$ for $i = 1 \ldots n-1$. Now, $G \cap G^{(1)}H = G$ and we have generators for it. Assume we have generators for $G \cap G^{(i)}H$. Then, by the above algorithm for GRS, we can find generators for $G \cap G^{(i+1)}H$, since $G \cap G^{(i+1)}H$ is a recognizable subgroup of $G \cap G^{(i)}H$ of small index. (For this, note that $[G \cap G^{(i)}H : G \cap G^{(i+1)}H] \leq [G^{(i)}H : G^{(i+1)}H] \leq [G^{(i)} : G^{(i+1)}] \leq n$ [recall the above exercise], and $G \cap G^{(i+1)}H$ has a polynomial time membership test [simply test membership in G *and* test membership in $G^{i+1}H$, the latter being generated by generators of G^{i+1} together with generators of H].) □

The above Proposition can be restated as: if $H \triangleleft \langle G, H \rangle$, then $G \cap H$ can be found in polynomial time. The following easy extension will be useful

Problem: INTERSECTION-SUBN

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

Find: $G \cap H$.

Proposition: INTERSECTION-SUBN is in P.

Proof: Exercise.

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

Lecture #3 **17 April 1990**

Algorithms for intersection and set stabilizer problems in nilpotent groups, with application to trivalent graph isomorphism

Recall from the last lecture: $H \triangleleft\triangleleft \langle G, H\rangle \Rightarrow$ can find $G \cap H$.

Also recall: G is nilpotent $\iff$ every subgroup of G is subnormal.

Problem: INTERSECTION-NIL

Given: G, H, subgroups of a nilpotent group N.

Find: $G \cap H$.

Remark: A last reminder that, unless specified otherwise, groups are input and output via generators.

Problem: STAB-NIL

Given: Nilpotent $G \leq Sym(\Omega)$, $\Delta \subset \Omega$

Find: $G_{\{\Delta\}}$.

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

Proof: $\langle G, H\rangle$ nilpotent $(\Rightarrow)$ $H \triangleleft\triangleleft \langle G, H\rangle$ $(\Rightarrow)$ can find $G \cap H$. $\square$

This algorithm will give us an application to ISO, but before we can use it, we'll have to develop some related comments.

It is promising to be able to do some group intersection, because of the following reductions:

ISO $\leq_P$ STAB $\leq_P$ INTERSECTION.

The first of these reductions, ISO $\leq_P$ STAB, we proved in the first lecture. The second reduction, STAB $\leq_P$ INTERSECTION, follows from observing that $G_{\{\Delta\}} = G \cap Sym(\Omega)_{\{\Delta\}}$, and noting that it is easy to give generators for $Sym(\Omega)_{\{\Delta\}}$, (*Exercise!*).

In fact, STAB and INTERSECTION are polynomial time equivalent. This follows from the following reduction:

Claim: INTERSECTION $\leq_P$ STAB.

Proof: For this reduction, suppose that both $G, H \leq Sym(\Omega)$. Then $G \cap H$ acts on $\Omega \times \Omega$. Let $Diag(\Omega \times \Omega) = \{(\omega, \omega) \mid \omega \in \Omega\}$. It is easy to check that $(G \times H)_{\{Diag(\Omega\times\Omega)\}} = \{(a, a) \mid a \in G \cap H\}$. $\square$

One of our short term goals is a polynomial time algorithm for STAB in nilpotent groups. Note, though, that our polynomial time algorithm for INTERSECTION for nilpotent groups is not sufficient for this, since the above reduction of STAB to INTERSECTION transforms an instance of STAB-NIL to an instance of INTERSECTION of a nilpotent group and a non-nilpotent group, which is *not* an instance of INTERSECTION-NIL.

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!*).

Recall the original motivation: we wanted to obtain $G_{\{\Delta\}}$, for G a 2-group in $Sym(\Omega)$. We noted that $G_{\{\Delta\}} = G \cap Sym(\Omega)_{\{\Delta\}}$. The above observation suggests obtaining $G_{\{\Delta\}}$ as the intersection of G with the Δ-stabilizer of a Sylow 2-subgroup of $Sym(\Omega)$, i.e. $G_{\{\Delta\}} = G \cap P_{\{\Delta\}}$, where P is a Sylow 2-subgroup of $Sym(\Omega)$ containing G.

Claim: Let $P \leq Sym(\Omega)$ be a Sylow 2-subgroup of $Sym(\Omega), \Delta \subseteq \Omega$. Then we can find $P_{\{\Delta\}}$ in polynomial time.

Proof: (sketch) Form the structure forest for P. Distinguish in some way the leaves which are points of Δ (by marking the leaves that are in Δ, for example). Now take the automorphism group of this modified (marked) tree to obtain $P_{\{\Delta\}}$.

We can now combine these ideas into an algorithm for finding set stabilizers in 2-groups:

Proposition: Let $G \leq Sym(\Omega)$ be a 2-group, and $\Delta \subseteq \Omega$. There is a polynomial time algorithm for finding $G_{\{\Delta\}}$.

Proof: *Algorithm:* 1. Embed G in a Sylow 2-subgroup P. 2. Find $P_{\{\Delta\}}$. 3. Form $G \cap P_{\{\Delta\}}$, which of course is $G_{\{\Delta\}}$. (For 3. note that G and $P_{\{\Delta\}}$ are subgroups of a common nilpotent group (P), so we may apply the algorithm for INTERSECTION-NIL.)

Generalization from 2-groups to p-groups

Essentially, all the above discussion for 2-groups extends naturally to p-groups. Instead of complete binary trees, complete p-ary trees will now arise. In a p-group, G, the stabilizer G_v of a node v in the structure forest for G will act primitively on the children of v, necessarily a set of size p. Of course, now, just having the tree isn't enough for our purposes, we will require some cyclic orientation of the children of each node v in the tree (i.e. obtained via a generator for G_v).

Problem: STAB-P

Given: $G \leq Sym(\Omega)$, a p-group, $\Delta \subseteq \Omega$

Find: $G_{\{\Delta\}}$

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

Proof: Same as algorithm given above for 2-group case, with the necessary modifications (e.g. we're now using p-ary trees with cyclic orientations on the children of each interior node).

Since, as noted above, STAB-NIL reduces to STAB-P, we now have a polynomial time algorithm for STAB-NIL.

Corollary: If G, H are nilpotent, we can find $G \cap H$.

Proof: The same technique used for the reduction of INTERSECTION to STAB works here: (Work with $G \times H$, a nilpotent group, acting on $\Omega \times \Omega$.)

Isomorphism of Trivalent Graphs

The reduction of trivalent graph isomorphism to STAB for 2-groups is given in see section 2 of [E.M. Luks, *Isomorphism of graphs of bounded valence can be tested in polynomial-time*, J. Comp. Sys. Sci., v.25 (1982) pp. 42–65]. (In that original paper, instead of STAB, one finds the equivalent 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).

Lecture #4 **23 April 1990**

Algorithms for computing centers, centralizers with application to solvable normal subgroup

Remarks: In the previous lecture we reduced TRIVALENT GRAPH ISOMORPHISM to STAB for 2-groups. We also solved the latter, more generally STAB in nilpotent groups. The method offered herein was included because of its easy dependence on two basic, attractive ideas: the subnormality of subgroups of nilpotent groups and the nature of Sylow subgroups of $Sym(n)$. Actually, the first announcement of a polynomial-time algorithm for finding set stabilizers in 2-groups followed a somewhat different approach [E.M. Luks, *Isomorphism of graphs of bounded valence can be tested in polynomial-time*, J. Comp. Sys. Sci., v.25 (1982) pp. 42–65]. That technique follows more directly a divide-and-conquer based upon the orbit and imprimitivity structure. It has the additional advantage of extending to an algorithm for finding $G \cap H$ where there is *no* assumption made about H, while G is only assumed to have bounded noncyclic composition factors (e.g., G could be solvable) [ibid, section 4.2].

Groups that arise in the (general) bounded valence d-case are not necessarily p-groups, not even solvable for $d \geq 6$. They do have the property that the primes in the order of the group are bounded. In fact the composition factors of the group are embeddable in $Sym(d-1)$. It has since been observed that just the fact that the noncyclic composition are bounded implies the primitive groups that arise in the course of the algorithm have polynomially bounded order [Babai, Cameron, Palfy]. This plays an important role in simplifying the algorithm (see comments in [Luks, ibid]).

Note: A nilpotent primitive group (acting on Ω) has order $= |\Omega|$.

Center and Centralizer

Problem: CENTER

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

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

Problem: CENTRALIZER

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

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

Note: CENTER $\leq_P$ CENTRALIZER. Cut G down in stages. At each stage centralize an additional generator of G. Therefore, repeated application of an algorithm for CENTRALIZER (once for each generator of G), gives an algorithm for CENTER.

Proposition: STAB $\leq_P$ CENTRALIZER.

Proof: Want to find $G_{\{\Delta\}} = \{g \in G \mid \Delta^g = \Delta\}$, where $\Delta \subseteq \Omega$.

Let G act naturally on the disjoint union $\bar{\Omega} = \Omega \dot{\cup} \Omega'$ of two copies of Ω. Let $h \in Sym(\bar{\Omega})$ switch the corresponding elements in the two copies of Δ while it fixes every other point. Then $C_G(h) = G_{\{\Delta\}}$ (*Exercise:* Verify this.). □

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 $\bar{\Pi}$ of Π that is compatible with the action of G, i.e., such that $\bar{\Pi}^x = \bar{\Pi}$ for $x \in G$.

It is easy to obtain this refinement: while $\exists a \in A$ such that $\Pi \neq \Pi^a$, replace Π by the common refinement of Π and Π^a (in particular, increasing the number of cells). When done, G acts on the collection of cells in the partition Π, and the kernel of this action is $C_G(H)$. Finding generators for the kernel of an action reduces to finding pointwise set stabilizers, which can be found in polynomial time.□

Note: The problem of finding the coarsest partition compatible with the action of G is closely related to the problem of finding minimum-state finite automata and fast techniques for the latter may be applied in this setting.

Corollary: If $G \leq Sym(n)$, then $G/Z(G) \hookrightarrow Sym(n^2)$.

Remark: $G/Z(G) \cong Inn(G)$, the inner automorphism group of G.

Remark: Repeating the algorithm to find $Z(G/Z(G))$, would seem to involve an additional squaring of the set size ($n^2 \to n^4$). Repeating the process to find the upper central series, will result in an exponential blow up in the size of the set. Therefore computing the upper central series seems to be difficult. Nevertheless, it can be computed in polynomial time [Kantor-Luks], although the algorithm depends upon the classification of finite simple groups.

Exercise: Given $G \leq Sym(\Omega)$, find $C_{Sym(\Omega)}(G)$. (The above method cannot be used as $Sym(\Omega)$ need not normalize G.)

Note: An algorithm to find $C_{Sym(\Omega)}(G)$, gives another algorithm for CENTRALIZER-N. For, if G normalizes H, then G normalizes $C_{Sym(\Omega)}(H)$. As $C_G(H) = G \cap C_{Sym(\Omega)}(H)$ and INTERSECTION-N is in P, this gives the required algorithm. (This method, too, does not seem to be of use in finding the upper central series.)

Application: Solvable normal subgroup

Problem: SOLVABLE NORMAL SUBGROUP (SNS)

Given: $G \leq Sym(\Omega)$

Question: Does G have a non-trivial solvable normal subgroup, $1 \neq H \triangleleft G$? If so find H.

Remark: We will show that SNS is in P. But eventually we want to find the maximal solvable normal subgroup. The maximal normal subgroup is unique, as H_1, H_2 solvable, normal $\Rightarrow \langle H_1, H_2 \rangle$ is solvable, normal.

Proposition: SNS is in P

Proof: For now, assume we know a proper normal subgroup $N, 1 \neq N \triangleleft G$, in polynomial time.

Procedure to solve SNS for G given $1 \neq N \triangleleft G$

```
    Solve SNS for $N$
    If some solvable $1 \neq H \triangleleft N$ is found then output $H^G$
    Else Solve SNS for $C_G(N)$
        If some solvable $1 \neq H \triangleleft C_G(N)$ is found then output $H^G$
        Else output "$G$ does not have a non-trivial solvable normal subgroup".
```

Correctness: Suppose G has a non-trivial solvable normal subgroup (say) H. If $H \cap N \neq 1$ then SNS for N will return some solvable group H_1 and ${H_1}^G$ will be solvable (${H_1}^G$ is solvable as it is generated by solvable normal subgroups of N, namely the G-conjugates of H_1). If $H \cap N = 1$, then $H \leq C_G(N)$ in which case SNS for $C_G(N)$ will return some solvable group.

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.

Lecture #5 **25 April 1990**

Algorithm for a special case of proper normal subgroup

In the previous lecture we saw that SNS can be solved in polynomial time if we can find a proper normal subgroup in polynomial time. We would like to find proper normal subgroups, but to solve SNS we need to do this only for the case when the group has non-trivial solvable normal subgroups (the algorithm for the general case is more involved and uses the classification of finite simple groups).

Problem: PROPER NORMAL SUBGROUP FOR SNS (PNS-S)

Given: $G = \langle A \rangle \leq Sym(\Omega), |G|$ is not prime.

Find: A proper normal subgroup of G or report that G does not have a non-trivial solvable normal subgroup.

Note: (i) If $|G|$ is prime, then G is solvable, and SNS can output G. (ii) In PNS-S, the "or" in the find statement is not exclusive, i.e. even if G has no non-trivial solvable normal subgroup PNS-S may return a proper normal subgroup.

Our approach will be to create a set of actions with the property that *if* a proper normal subgroup exists, then at least one of these actions will have a nontrivial kernel.

If Δ is a non-trivial orbit (i.e. $|\Delta| > 1$), then find G_Δ, the kernel of the action of G on Δ. If $G_\Delta \neq 1$, then we're done (output G_Δ). Otherwise, we may assume G acts transitively on Ω (G acts *faithfully* on Δ, so we may replace Ω with Δ). If G is not primitive, then let $\mathcal{B}$ be a block system. If $G_{\mathcal{B}} \neq 1$ then we're done (output $G_{\mathcal{B}}$). Otherwise, we may assume G acts primitively on Ω (G acts *faithfully* on the blocks $\mathcal{B}$ so we may replace Ω with $\mathcal{B}$).

Remark: Suppose $\phi_1 : G \hookrightarrow Sym(\Omega)$, and we have some auxiliary action $\phi_2 : G \to Sym(\Delta)$. We can find the kernel of the action of G on Δ by viewing $g \in G \leq Sym(\Omega \cup \Delta)$ as the product of two permutations $g = \phi_1(g)\phi_2(g)$ and take the pointwise set stabilizer G_Δ in this action. (This idea has already been implicit in earlier lectures).

Subgroups described in terms of an induced action

In response to queries, we expand a bit on the last remark. More generally, whenever we are computing a subgroup H of G in its action on Δ, we can also obtain the "pullback" of this subgroup on the original action on the set Ω (i.e. $\{\phi_1(g) \mid \phi_2(g) \in H\}$) as follows. View group elements as ordered pairs $(\phi_1(g), \phi_2(g))$. As our algorithm for computing H with respect to Δ directs us to form products, take inverses, etc. of our generators in the second coordinate, we simply duplicate the identical computations in the first coordinate as well. Our "answer" back on the original set Ω, then, will be generated by the set of resultant first coordinates, *together with* generators for the kernel of the action of G on Δ.

Alternatively, we could simply compute H on Δ, (computing with the second coordinates, ignoring the first coordinates), then "lift" these permutations on Δ to permutations on $\Omega \cup \Delta$, (as outlined in the proof of the following claim) and restrict these permutations to Ω.

Claim: (*Membership of Partial Permutations*) Let $G \leq Sym(\Omega)$, $\Delta \subset \Omega$. Let $f : \Delta \hookrightarrow \Omega$ be given. Then it is possible to find (in polynomial time) an extension of f to an element $g \in G$, (in fact, to find *all* such extensions) if one exists, or to determine that there are none.

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_Δg.

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. $\Box$

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.

In fact, however, we have only generators for G and a "promise" that G has an abelian (and therefore regular) normal subgroup H. To find a proper normal subgroup of G it suffices to produce a nontrivial primitive action of G for which H is not regular. Such an action cannot be faithful, so the kernel of that action will be a proper normal subgroup.

It suffices to produce a transitive action $\pi : G \longrightarrow Sym(\Delta)$ in which $\exists \delta \in \Delta$ such that $H_\delta \neq 1$. For, if G is transitive on Δ and $H_\delta \neq 1$ for some $\delta \in \Delta$, then the same holds for any block system on Δ, so a primitive action of G for which H is not regular can be obtained.

It is sufficient to find a cycle Γ of any $1 \neq h \in H$. For, let $\Delta = \{\Gamma^g \mid g \in G\}$, then Γ^g is a cycle for $h^g = g^{-1}hg \in H$, and any edge in Γ^g determines the unique element of H inducing it (as H is regular on Ω, and by (i) of exercise above), and hence determines the whole cycle. Therefore $|\Delta| < n^2$, and we can use a transitive closure algorithm to form all the Γ^g's. So G acts transitively on Δ, and $\exists\ 1 \neq h \in H$ such that h fixes the point $\Gamma \in \Delta$.

Note: If we knew $h \in H$, then all we need to output is $\langle h \rangle^G$ and we would be done.

We need to find a cycle for some $1 \neq h \in H$ (but don't have h). Take any $\alpha, \beta \in \Omega, \alpha \neq \beta$. Then $\exists!\ h \in H$ such that $\alpha^h = \beta$. We want to complete the cycle $(\alpha \to \beta \to \ldots)$ of that h. Observe that it is sufficient to know β^h (i.e. three points are sufficient to complete the cycle). For, let $\gamma = \beta^h$. Find $g \in G$ such that $(\alpha, \beta)^g = (\beta, \gamma)$. Since $\beta^{g^{-1}hg} = \gamma = \beta^h \Rightarrow g^{-1}hg = h$ (h is the unique element in H that maps β to γ), $\Rightarrow \gamma^h = \gamma^{g^{-1}hg} = \gamma^g$. An easy induction argument shows that the cycle of g containing α is the same as that of h.

Although we don't know such a γ, we consider all points of Ω as potential candidates, and try them all:

```
**Algorithm**
Fix $\alpha, \beta \in \Omega$, $\alpha \neq \beta$.
For each $\gamma \in \Omega$ do
    If there exists $g \in G$ such that $\alpha \xrightarrow{g} \beta \xrightarrow{g} \gamma$, Then
        $\Gamma \leftarrow$ cycle of $g$ containing $\alpha$.
        $\Delta \leftarrow$ $G$-images of $\Gamma$ (if $\geq n^2$ such, reject $\gamma$)
        $\mathcal{B} \leftarrow$ a minimal block system of $G$'s action on $\Delta$.
        $K \leftarrow$ the kernel of the primitive action of $G$ on $\mathcal{B}$.
            If $K \neq 1$ Output $K$.
            Else reject $\gamma$.
    Else reject $\gamma$.
If all $\gamma \in \Omega$ are rejected, Output "$G$ does not have a non-trivial solvable normal subgroup".
```

Note that *if* G has an abelian normal (and regular) subgroup, then one of the actions considered in this algorithm *will* have a nontrivial kernel, and the algorithm will succeed in finding it. If no infaithful action is found, G couldn't have had an abelian normal subgroup, hence it couldn't have had a solvable normal subgroup.

Comment: If G is primitive and has a regular normal subgroup H which is not abelian, then the algorithm above could fail to find a nontrivial kernel, since the fact that H is abelian guarantees that H is regular in any faithful primitive action of G. Of course, if on Δ, G is faithful and primitive, but H isn't regular, then $|\Delta| < |\Omega|$, in fact $|\Delta| = \frac{|H|}{|H_\delta|} \leq \frac{1}{2}|H| = \frac{1}{2}|\Omega|$.

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 $g^{-1}h_i g$ 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.

Problem: CORE-p

Given: $G = \langle A \rangle \leq Sym(\Omega)$, and a prime p.

Find: $O_p(G)$.

Problem: FITTING

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

Find: $Fit(G)$.

Note: FITTING $\leq_P$ CORE-p, as $Fit(G) = \prod_{p\,prime} O_p(G)$.

Problem: CORE-p-SOLVABLE

Given: A solvable group $G = \langle A \rangle \leq Sym(\Omega)$, and a prime p.

Find: $O_p(G)$.

Claim: CORE-p $\leq_P$ CORE-p-SOLVABLE

Proof: Let $N \triangleleft G$ and $O_p(G) \leq N$, then $O_p(G) = O_p(N)$ since $O_p(N)$ is characteristic in N and hence it is normal in G. (e.g $N = Rad(G)$). Therefore $O_p(G) = O_p(Rad(G))$, where $Rad(G)$ is solvable, and can be found in polynomial time. □

The rest of this lecture indicates some polynomial time algorithms for CORE-p.

Version 1

Claim: There is a polynomial time algorithm for CORE-p-SOLVABLE.

Proof:

```
Algorithm
    $P \leftarrow$ Sylow $p$-subgroup of $G$ (see next Remark)
    While $P$ is not normal in $G$ do
        Find $g \in A$ such that $P^g \neq P$
        $P \leftarrow P \cap P^g$
    Return $P$
```

The algorithm will return $O_p(G)$ (as $O_p(G) = \cap_{g \in G} P^g$). □

Remark: Sylow subgroups of solvable groups can be found by "elementary" methods [Kantor-Taylor] (later simplified further by Kantor) in distinction to known polynomial-time methods for Sylow subgroups in general groups, which require the classification of finite simple groups [Kantor]. If the reader is not concerned about this distinction then he/she can ignore the reduction to the reduction to solvable G and accept the above as a direct algorithm for CORE-p.

Version 2

A more direct algorithm for CORE-p can be obtained by modifying the algorithm for RADICAL. Our ability to locate solvable normal subgroups in polynomial time played a crucial role in the algorithm for RADICAL. Similarly, for CORE-p, we need to find normal p-subgroups in polynomial time. The SNS algorithm, with some minor modifications, can be used to find normal p-subgroups. The only tricky point is showing this algorithm runs in polynomial time. Recall that in the algorithm for SNS, we recursively called SNS with input N and $C_G(N)$ where N was a proper normal subgroup of G, and we noted that if the second recursive call is invoked then $|G| \geq |N||C_G(N)|$. If we modify SNS to find just p-normal subgroups, then the above inequality may not hold.

Exercises: (1) Verify that the modified SNS algorithm to find p-normal subgroups runs in polynomial time. (Hint: Always check first that p divides the group order).

(2) Complete the algorithm for CORE-p.

Version 3

Algorithm

```
    $H \leftarrow$ an abelian normal subgroup of $G$
    If $p \not| \, |H|$ then
        $G \leftarrow C_G(H)$
        $\{$ From now on, we can assume that $H \leq Z(G)\}$
        $\Sigma \leftarrow \{\Delta_1, \Delta_2, \ldots, \Delta_r\}$, the set of orbits of $H$
        Let $G \overset{\pi}{\to} Sym(\Sigma)$ denote the action of $G$ on $\Sigma$
        $K \leftarrow Ker(\pi)$
        $G \leftarrow C_G(K)$
        $Q/K \leftarrow O_p(\pi(G))$
        $\{$ Then $Q = O_p(G) \times K\}$
        Output the generators of $Q$ raised to the $p'$ part of $|G|$
    Else $\{p \,|\, |H|\}$
        $H \leftarrow$ minimal normal $p$-subgroup of $G$ (details to follow)
        $G \leftarrow C_G(H)$
        $\Sigma \leftarrow \{\Delta_1, \Delta_2, \ldots, \Delta_r\}$, the set of orbits of $H$
        Let $G \overset{\pi}{\to} Sym(\Sigma)$ denote the action of $G$ on $\Sigma$
        $K \leftarrow Ker(\pi)$
        $Q/K \leftarrow O_p(\pi(G))$
        Output generators for $Q$
```

Correctness: We can assume we have $1 \neq H \triangleleft G$, with H abelian (if N is the solvable normal subgroup returned by SNS with input G, then let H be the last nontrivial term in the derived series for N). Let $\Delta_1, \Delta_2, \ldots, \Delta_r$ be the orbits of H. Let $G \overset{\pi}{\to} Sym(\Sigma)$ denote the action of G on $\Sigma = \{\Delta_1, \Delta_2, \ldots, \Delta_r\}$, and $K = Ker(\pi)$. If p does not divide $|H|$, then $O_p(G) \leq C_G(H)$ (as H, $O_p(G) \triangleleft G$ and $H \cap O_p(G) = 1$). Note that $H \leq K$ and K has the same constituents as H (as K commutes with H), so K is an abelian, normal p-prime subgroup of G. $\pi(O_p(G)) \leq O_p(\pi(G)) = Q/K \Rightarrow O_p(G) \leq Q \triangleleft G \Rightarrow O_p(G) = O_p(Q)$. Let P be any Sylow p-subgroup of Q, then $|P||K| \geq |Q|$ as Q/K is a p-group, so $Q = PK$. Since $P \leq C_G(K)$ so $Q = P \times K$ and P is normal in Q. Hence $O_p(G) = O_p(Q) = P$, $Q = O_p(Q) \times K$.

If p divides $|H|$ we don't know that $O_p(G) \leq C_G(H)$.

Exercise: If H is a minimal normal p-subgroup of G then $O_p(G) \leq C_G(H)$ and H is elementary abelian.

Solution: $[H, H]$ is a characteristic subgroup of H, so it is normal in G. By minimality of H, $[H, H] = H$ or 1. Since H is solvable (H is a p-group) $[H, H] = 1$, so H is abelian. The subgroup H_p of H generated by all the elements of order p, is characteristic in H and hence normal in G. Since $H_p \neq 1$ therefore $H_p = H$ and hence H is elementary abelian.

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

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.

Lecture #8 **7 May 1990**

The structure of primitive permutation groups

The next two lectures focus on the structure of minimal normal subgroups and socles. We'll look first at general (finite) groups, then concentrate on primitive permutation groups. Primitive groups arise naturally as the base case in certain divide-and-conquer algorithms. In a sense, this base case occurs when naive combinatorial divide-and-conquer has been exhausted. Additional decomposition of the problem is often possible but now uses the structure of the group itself. To get at this structure, we develop a portion of the O'Nan-Scott Theorem for permutation groups. Studying particularly the nature and action of the socles of primitive groups provides the key to obtaining further divide-and-conquer decompositions. In these lectures, our immediate goal is to reveal the composition factors of a group. The machinery also plays a key role in the parallelization of (most of) the algorithms we've seen so far in the course.

Definition: The **socle** of a finite group G, denoted $Soc(G)$, is the subgroup generated by all minimal normal subgroups of G.

Proposition: The group generated by some collection of minimal normal subgroups of G is a direct product of a subcollection of them.

Proof: Let $N = \langle \mathcal{B} \rangle$. Let $\mathcal{M} = \{M_1, \ldots, M_r\}$ be a maximal subcollection of $\mathcal{B}$ such that $K = \langle \mathcal{M} \rangle = M_1 \times \ldots \times M_r$. Our goal is to show $N = K$. It suffices to show that for *every* minimal normal subgroup $M \in \mathcal{B}$, $M \leq K$. Suppose, to the contrary, there is some minimal normal subgroup $M \in \mathcal{B}$ such that $M \not\leq K$. Then $M \cap K = 1$ (M is minimal normal), and $K \times M$ contradicts our choice of K. □

Corollary: $Soc(G)$ is a direct product of minimal normal subgroups of G.

Definition: A group H is **characteristically simple** if it has no proper characteristic subgroups.

Claim: H is characteristically simple $\iff$ H is a direct product of isomorphic simple groups.

Proof: ($\Leftarrow$) Let H be a direct product of isomorphic simple groups $M_1 \times \ldots \times M_r$. Suppose first that the M_i are abelian: then H is a vector space over a finite field, subgroups are subspaces and $Aut(H)$ is the group of linear transformations of H; give any proper subspace N, there is a linear transformation that does not preserve N; hence N is not characteristic. Suppose next that the M_i are nonabelian: clearly $Aut(H)$ induces all permutations of the factors $\{M_i\}$; let N be a proper normal subgroup of H, then $N = M_{i_1} \times \ldots \times M_{i_s}$ for some $1 \leq i_1 \leq \ldots \leq i_s \leq r$ (*Exercise:* prove this!); since N is proper, i.e., $s \neq 0, r$, there exists a permutation of $\{M_i\}$ that does not stabilize $\{M_{i_1}, \ldots, M_{i_s}\}$; hence N is not characteristic.

($\Rightarrow$) If H is characteristically simple, then $H = Soc(H)$ (the socle is always a characteristic subgroup). It follows that $H = M_1 \times \ldots \times M_r$, the direct product of minimal normal subgroups of H. Moreover, the M_i's are simple, since if $K \triangleleft M_i$, then $K \triangleleft H$ (since M_i is a direct factor of H), which contradicts the minimality of M_i. Let $\mathcal{M}$ be the collection of all minimal normal subgroups of H that are isomorphic to M_1, and $K = \langle \mathcal{M} \rangle$. Then K is characteristic in H, and so (by the characteristic simplicity of H) $K = H$. By the earlier proposition, H is the direct product of some of the M's in $\mathcal{M}$. □

Corollary: If M is a minimal normal subgroup of a group G, then M is the direct product of isomorphic simple subgroups.

Proof: Minimal normal subgroups are characteristically simple. □

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 $[M_1, M_2] \lhd G$, 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**

action of G: $\lambda : G \to Sym(G), g^\lambda : h \mapsto g^{-1}h$. *Exercise:* verify that ρ and λ are actions, (i.e. homomorphisms), and that these actions commute.

When a group G acts regularly on a set Ω, we may identify Ω with G by distinguishing some point $\omega \in \Omega$ and for each $g \in G$, identifying g with ω^g. It is clear that with this identification, G's action on Ω is precisely the right regular action on itself. If there is some other group H that acts regularly on Ω and commutes with G, we know by the Proposition that it *must* be the left regular action. (The left regular action is *some* commuting action, the remark tells us it is the *only* one.) Thus, $G \cong H$.

In particular, distinct minimal normal subgroups of a primitive group are isomorphic.

We note also that an abelian normal subgroup of primitive group is necessarily regular and self-centralizing (apply the proposition - it commutes with itself).

We summarize some important implications of the foregoing discussion that we'll need later in the following:

Theorem: If G is primitive, then $Soc(G) = M_1$ or $Soc(G) = M_1 \times M_2$, where $M_1 \cong M_2$ and M_1, M_2 are the minimal normal subgroups of G. In either case, $Soc(G) = T_1 \times \ldots \times T_r$ where the T_i's are all isomorphic simple subgroups.

Remark: If $Soc(G)$ is abelian, then $Soc(G)$ is an elementary abelian p-group, each T_i is cyclic of order p, and every minimal normal subgroup of $Soc(G)$ is also cyclic of order p (it is a "subspace" of $Soc(G)$). In this case, $\{T_1, \ldots, T_r\}$ is a subset of the set of all minimal normal subgroups of $Soc(G)$. If $Soc(G)$ is nonabelian, $\{T_1, \ldots, T_r\}$ actually comprises *all* minimal normal subgroups of $Soc(G)$. Moreover, the only normal subgroups of $Soc(G)$ are direct products of some of the T_i's.

We now turn our attention to analyzing the structure of point stabilizers of socles of primitive groups. This structure plays a key role in the algorithm for finding composition factors, and is an important ingredient in the machinery for parallelizing much permutation group machinery.

Point stabilizers of socles of primitive permutation groups

First consider the case where G is a primitive *solvable* group. $N = Soc(G)$ is abelian. Since N is a direct product of isomorphic simple subgroups, it must in fact be an elementary abelian p-group, $N \cong {Z_p}^d$. Since N is transitive and abelian, it is also regular, and $|\Omega| = |N| = p^d$, and we may identify Ω with N via the right regular action. (Recall we fix a point ω and identify $n \in N$ with $\omega^n \in \Omega$.)

This endows Ω with a vector space structure. Since N is abelian, we'll use additive notation for the group operation in N. N's (right regular) action on Ω induces the full group of vector space translations on Ω: for a point $\omega^{n_1} \in \Omega$, and an element $n_2 \in N, \rho(n_2) : \omega^{n_1} \mapsto (\omega^{n_1})^{n_2} = \omega^{n_1+n_2}$.

Since ω is identified with $1 \in N$, it is natural to view ω as the origin of the vector space Ω. Consider the point stabilizer G_ω. Since G is primitive, we know $G = G_\omega N$ (in fact this is true whenever N is normal and transitive). Since N is regular, $N_\omega = G_\omega \cap N = 1$, so G is the *semidirect product* of N and G_ω. Therefore, to understand G's action on Ω, it suffices to know N's action and G_ω's action on Ω. We've already noted that N acts on Ω as the full group of vector space translations.

G_ω acts on Ω (via the original given action of G on Ω); it also acts on N via conjugation, and hence on Ω via the identification we've made between elements of N and points of Ω. *These two actions are in fact the same!* Consider $n \in N$ and its associated point $\omega^n \in \Omega$, and let $g \in G_\omega$. We need to verify that the point g maps ω^n to (in the original given action $G \to Sym(\Omega)$), and the point in Ω

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 congugation 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 $H = N$. □

Lecture #9 **9 May 1990**

The structure of primitive permutation groups: II

Case I

We'll define a case I group to be a primitive group G with a regular normal subgroup. This includes as a subcase the situation described in detail in the last lecture: $Soc(G)$ abelian, hence regular.—It also includes the case where G has more than one (therefore exactly two) minimal normal subgroups, each of which is regular. In addition, it includes the case where G has a regular, nonabelian socle, which we do not need to analyze further.

In both case II and case III, G has no regular normal subgroup. Assume now that G has no regular normal subgroup.

Here $N = Soc(G), G$ acts primitively on Ω. $N = T_1 \times \ldots \times T_r$, where the T_i's are nonabelian simple isomorphic groups. $G_\omega, (\omega \in \Omega)$ acts transitively (by conjugation) on $\{T_1, \ldots, T_r\}$. Our interest again focuses on the structure of point stabilizers of N. $N_\omega \leq N = T_1 \times \ldots \times T_r$. Let π_i be the i-th projection function $\pi_i : N \to T_i$ given by $n \mapsto n_i$ where $n = n_1 \ldots n_r$, and $n_i \in T_i \; \forall i = 1, \ldots r$.

Let $S_i = \pi_i(N_\omega)$. Note that while S_i is clearly a subgroup of T_i, S_i may not be a subgroup of N_ω.

Claim: G_ω acts transitively on $\{S_1, \ldots, S_r\}$.

Proof: To see that G_ω acts on $\{S_1, \ldots, S_r\}$, we verify that, for $g \in G_\omega$ such that ${T_i}^g = T_j$, we additionally have ${S_i}^g = S_j$; that G_ω acts *transitively* on $\{S_1, \ldots, S_r\}$ will then be clear since G_ω acts transitively on $\{T_1, \ldots, T_r\}$.

Let $s_i = \pi_i(s)$, where $s = s_1 \ldots s_r \in N_\omega$. For an element $g \in G_\omega$, $s^g \in N_\omega$ because $N_\omega \lhd G_\omega$. Since G_ω acts on $\{T_1, \ldots, T_r\}$, and $s_i \in S_i \leq T_i$, we have ${s_i}^g \in T_j$ (where $T_j = T_i^g$). It follows that $\pi_j(s^g) = \pi_j({s_1}^g \ldots {s_r}^g) = {s_i}^g$, so ${S_i}^g \leq S_j$; similarly $S_j^{g^{-1}} \leq S_i$, so ${S_i}^g = S_j$. □

Note: Since G_ω acts transitively on $\{S_1, \ldots, S_r\}$, if $S_i = T_i$ for any i, then $S_i = T_i$ for all $i = 1, \ldots, r$.

Case II

We define a case II group to be a primitive group G with no regular normal subgroup and $\pi_i(N_\omega) = S_i < T_i$ for each i. Furthermore, $S_i \neq 1$, otherwise N would be regular.

Claim: $N_\omega = (T_1)_\omega \times \ldots \times (T_r)_\omega = S_1 \times \ldots \times S_r$.

Proof: Let $H = \langle S_1, \ldots, S_r \rangle = S_1 \times \ldots \times S_r$. We know $N_\omega \leq H \leq N$. The latter inclusion is proper, since $S_i < T_i$. Since H is a G_ω-invariant subgroup containing N_ω, a *maximal* G_ω-invariant subgroup (see the last claim of the last lecture), $N_\omega = H$. In particular, $S_i \leq N_\omega$ for each i. Thus, $(T_i)_\omega \leq S_i \leq N_\omega \cap T_i = (T_i)_\omega$, proving the claim. □

Note: Since N is transitive, we may identify points of Ω with cosets of N_ω. Since $N_\omega = (T_1)_\omega \times \ldots \times (T_r)_\omega$, we may identify a coset $N_\omega n$ with an r-tuple $((T_1)_\omega t_1, \ldots, (T_r)_\omega t_r))$, where $n = t_1 \ldots t_r$. If we define $\Omega_i = \{(T_i)_\omega t_i \mid t_i \in T_i\}$, then the set Ω may be identified as a product: $\Omega = \Omega_1 \times \ldots \times \Omega_r$ and $|\Omega| = |\Omega_1|^r = \left(\frac{|T_1|}{|(T_1)_\omega|}\right)^r$. We don't explicitly use this result but it does lead to an alternative (perhaps more intuitive) motivation of a step (specifically Step 6) in the simplicity-test algorithm.

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. □

Lecture #10 **17 May 1990**

The structure of primitive permutation groups: III and an algorithm for testing simplicity

Case III

Recall from lecture 9 the definition of a case III group: A primitive group G with no regular normal subgroup where $N = Soc(G) = T_1 \times \ldots \times T_r$, T_i nonabelian simple and $\pi_i(N_\omega) = T_i$ for each i.

N_ω is a subdirect product, so the lemma from lecture 9 applies and we may assume:

$$N_\omega = diag(T_1 \times \ldots \times T_k) \times diag(T_{k+1} \times \ldots \times T_{2k}) \times \ldots \times diag(T_{(l-1)k+1} \times \ldots \times T_{lk}) = D_1 \times \ldots \times D_l.$$

Additionally, (as suggested by the notation), each diagonal block is composed of the same number of T's. This follows from the observation that the equivalence relation on $\{1, \ldots, r\}$ (defined in the last lecture), is G_ω-invariant, i.e. it's respected by the conjugation action of G_ω, (because the diagonal groups D_i are the minimal normal subgroups of N_ω, so G_ω must be permuting them).

Here in case III, $|\Omega| = \frac{|N|}{|N_\omega|} = \frac{|T_1|^r}{|T_1|^l} = |T_1|^{(k-1)l} = |T_1|^{r-l}$. Recall that in case II, $|\Omega| = \left(\frac{|T_1|}{|(T_1)_\omega|}\right)^r$.

Claim: If G's action on Ω falls into case III then the size of G is $\mathcal{O}(n^{cloglogn})$ where $n = |\Omega|$.

Proof: Observe $|N| = |T_1|^{kl} \leq |T_1|^{2(k-1)l} = n^2$. Since $C_G(N) \lhd G$ and $\{T_i\}_{1 \leq i \leq r}$ is the set of all minimal normal subgroups of G, $C_G(N) = 1$ (for if not, then $T_j \leq C_G(N) \leq C_G(T_j)$ for some j, which implies that T_j is abelian, a contradiction). Therefore $G \hookrightarrow Aut(N)$. Now, $|Aut(N)| \leq |Aut(T_1)|^r r!$, and $|Aut(T_1)| \leq |T_1|^2$ (using the fact from the classification of simple groups, that every simple group is generated by two elements). So we get $|Aut(T_1)|^r \leq |T_1|^{2r} \leq |N|^2 \leq n^4$. To estimate $r!$, we observe that $n \geq |T_1|^{r/2} \geq (\sqrt{60})^r$ (since the smallest nonabelian simple group is A_5), which implies $r \in \mathcal{O}(\log(n))$, and therefore $r! \in \mathcal{O}(n^{\log(\log n)})$. Hence $|G| \leq |Aut(N)| \leq |Aut(T_1)|^r r! \in \mathcal{O}(n^{\log(\log n)+4})$. □

Remarks: This estimate will not be used in these lectures, having arisen only as a digression in answer to a query about the lecturer's suggestion that Case III groups are "small" in relation to the permutation domain. However, arguments of this sort have proved useful in other algorithmic studies.

The intuition that was actually being offered was the fact that, if a given abstract group G with $Soc(G) = T_1 \times \ldots \times T_r$ acts primitively in a case II fashion on Ω_1 and in case III fashion on Ω_2, then it seems most likely that $|\Omega_1| < |\Omega_2|$ will usually hold (comparing the expressions preceding the claim). The simplicity test will take advantage of this (in Step 7).

Normal subgroups

Problem: PROPER NORMAL SUBGROUP (PNS)

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

Find: Generators for a proper normal subgroup of G or report that G is simple.

Remark : Testing for simplicity is in Co-NP, for a subgroup can be verified to be a normal subgroup in polynomial time. Our goal is to show that PNS can be solved in polynomial time.

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.

Step 4:

```
Else $|G| > |\Omega| = n$
    $\{g_0, g_1, \ldots, g_n\} \leftarrow n+1$ distinct elements of $G$.
    For $0 \leq i \leq j \leq n$ do
        $H_{ij} \leftarrow \langle g_i g_j^{-1} \rangle^G$.
        If $H_{ij} \neq G$ output $H_{ij}$.
```

Since $|G| > |\Omega| = n$, we can find $n+1$ distinct elements, $g_0, g_1, \ldots, g_n$, of G. $H_{ij} = \langle g_i g_j^{-1} \rangle^G \neq 1$ since the g_i's are distinct. If $H_{ij} \neq G$ then we have found a proper normal subgroup and we are done. Otherwise we know that G does not have a proper normal subgroup of index $\leq n$. For suppose $N \triangleleft G$ with $|G : N| \leq n$, then (by pigeon hole) $\exists\ 0 \leq i \neq j \leq n$ such that $Ng_i = Ng_j$ and so H_{ij} would have been a proper normal subgroup ($\langle g_i g_j^{-1} \rangle^G \leq N < G$).

Note : The above does not give us an algorithm to find a proper normal subgroup of small index. It merely produces *some* proper normal subgroup *if* there is a proper normal subgroup of small index.

We may now assume that G does not have a normal subgroup of index $\leq n$.

Remark : If $G' < G$ then G has a normal subgroup of index $\leq n$ (since G/G' is abelian, the pullback of a maximal normal subgroup of G/G' would be of prime index in G). If we had tested for $G' = G$ earlier we would have either found a proper normal subgroup G' or would be in this case. Hence testing for a proper G' is superflous.

Step 5:

```
$\{G$ has no proper normal subgroup of index $\leq n.\}$
Fix $\alpha, \beta \in \Omega$, $\alpha \neq \beta$.
For each $\gamma \in \Omega$ do
    If there exists $g \in G$ such that $\alpha \xrightarrow{g} \beta \xrightarrow{g} \gamma$, then
        $\Gamma \leftarrow$ cycle of $g$ containing $\alpha$.
        $\Delta \leftarrow G$-images of $\Gamma$ (if $\geq n^2$ such reject $\gamma$).
        Find a minimal block system $\mathcal{B} = \{B_i\}$ for this transitive action of $G$ on $\Delta$.
            Let $G \xrightarrow{\pi} Sym(\mathcal{B})$ be this primitive action.
            $K \leftarrow ker(\pi)$.
                If $K \neq 1$ output the proper normal subgroup $K$.
                Else if $|\mathcal{B}| \leq |\Omega|/2$
                    output $G \overset{\pi}{\hookrightarrow} Sym(\mathcal{B})$.
                    Else reject $\gamma$.
    Else reject $\gamma$.
```

If all γ are rejected then we know that G has no regular normal subgroup. For if there was a regular normal subgroup N then for some γ we would find a $g \in G$ such that $\alpha \xrightarrow{g} \beta \xrightarrow{g} \gamma$, (and as seen in lecture 5) we would find a primitive action of G in which N does not acting regularly. If this action is faithful then $|\mathcal{B}| = |\{B_i\}| = |N|/|N_{\{B_1\}}| \leq |N|/2 = n/2$ and this gives a new primitive action of G on a set of size $\leq n/2$.

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.

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.

where

in Case II: $m = r$, and $N_i = T_i$, and $n = (|T_1|/|(T_1)_\alpha)^r$

in Case III: $m = l$, and $N_i = T_{(i-1)k+1} \times \ldots \times T_{ik}$, $(N_i)_\alpha = diag(N_i)$, and $n = |T_1|^{r-l}$.

Fix some $\alpha \in \Omega$. In running step 6 we will try some $\beta = \alpha^{n_2} \neq \alpha$ for some $n_2 \in N_2$ (by (d)). Then $(N_1)_\beta = (N_1)_{\alpha^{n_2}} = {n_2}^{-1}(N_1)_\alpha n_2 = (N_1)_\alpha$ (since N_1, N_2 commute). Hence $(N_1)_\alpha = (N_1)_\beta \leq G_{\alpha\beta}$. Since G acts on $\{N_i\}_{1\leq i\leq r}$, so does $G_{\alpha\beta}$. *Subclaim:* $G_{\alpha\beta}$ normalizes N_2. *Proof:* Let $g \in G_{\alpha\beta}$ and ${N_2}^g = g^{-1}N_2g = N_j$. Then $g^{-1}n_2g \in N_j$, and $\alpha^{g^{-1}n_2g{n_2}^{-1}} = \alpha$. By (c) we have $N_\alpha = (N_1)_\alpha \times \ldots \times (N_r)_\alpha$. Therefore $g^{-1}n_2g{n_2}^{-1} \in (N_1)_\alpha \times \ldots \times (N_r)_\alpha$ where $g^{-1}n_2g \in N_j$ and ${n_2}^{-1} \in N_2$. By the unique factorization in the direct product, we must have $j = 2$ (otherwise $g^{-1}n_2g \in (N_j)_\alpha$ and ${n_2}^{-1} \in (N_2)_\alpha$ which contradicts $\beta = \alpha^{n_2} \neq \alpha$). This proves the subclaim.

By (d) again, there exists $\gamma \in \Omega$ such that $(N_1)_\gamma \neq (N_1)_\alpha$. Corresponding to this γ there exists $\delta \in \Omega$ such that $(N_1)_\gamma = (N_1)_\delta$ (so $(N_1)_\gamma \leq G_{\gamma\delta}$) and $G_{\gamma\delta}$ normalizes N_2 (by the same argument as in the last paragraph).

Since $|(N_1)_\gamma| = |(N_1)_\alpha|$ (by (c)) but $(N_1)_\gamma \neq (N_1)_\alpha$, we have $(N_1)_\alpha < \langle (N_1)_\alpha, (N_1)_\gamma \rangle$. Also, $\langle (N_1)_\alpha, (N_1)_\gamma \rangle \leq H = \langle G_{\alpha\beta}, G_{\gamma\delta} \rangle$. Since $G_{\alpha\beta}$ and $G_{\gamma\delta}$ normalize N_2, $H \leq N_G(N_2)$ and $N_G(N_2) < G$ (because G acts transitively on the N_i's). Therefore for this $\{\beta, \gamma, \delta\}$, $H \neq G$ and the non-trivial action of G on a minimal block system Δ for G's action on the cosets of H is constructed. Let $h \in \Delta$ be the block containing the coset H. If this action is not faithful then the kernel is a proper normal subgroup. Otherwise we have a (faithful) primitive action of G in which $1 \neq \langle (N_1)_\alpha, (N_1)_\gamma \rangle \leq H = G_H \leq G_h$.

If the action of G on Ω is in Case II then the primitive action of G on Δ is also in Case II (since $1 \neq (N_1)_\alpha \leq (N_1)_h = (T_1)_h$ and in the other cases (Case I and III) $(T_1)_h = 1$ (Note: we only have to exclude subcases of Case 1 where there is a unique minimal normal subgroup but this relation holds in the third subcase as well). In this case the size of the new set $\Delta = (|T_1|/|(T_1)_h|)^r < (|T_1|/2|(T_1)_\alpha|)^r \leq (1/2)|\Omega|$. □

If the action of G on Ω is in Case III then the primitive action of G on Δ is also in Case III, for $(N_1)_\alpha$ projects onto T_1 and $(N_1)_h \geq (N_1)_\alpha$ implies $(N_1)_h$ also projects onto T_1, which shows that it is not in Case II. It is not in the subcases of Case I with unique minimal normal subgroup since $(N_1)_h \neq 1$. Since $N_1 = T_1 \times \ldots \times T_k$, and $(N_1)_h > (N_1)_\alpha$, $(N_1)_h$ must be a product of diagonal subgroups corresponding to a proper partition of $\{1, \ldots, k\}$ into cells of size $k' < k$. Hence if $l' = r/k'$, then $l' > l$ and $|\Delta| = |T_1|^{r-l'} = |T_1|^{r-l}/|T_1|^{l'-l} = |\Omega|/|T_1|^{l'-l} \leq |\Omega|/2$. □

Step 7:

```
For all $\alpha, \beta \in \Omega$ do
    $\Gamma \leftarrow \{\alpha, \beta\}^G$.
    $\Delta \leftarrow$ a minimal block system for $G$'s action on $\Gamma$.
    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 $\{\alpha, \beta\}$.
```

Claim: Suppose the action of G on Ω falls into Case III with $l = 1$. If Step 7 is reached the algorithm will halt there.

Proof: Let $t_1 \in T_1$ have order 2 (such a t_1 exists by the Feit-Thompson Theorem). Therefore 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 $|T_1| \geq |(T_1)_h|^r/2 \geq 2^{r-1}$. 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. □

Lecture #12 **25 May 1990**

An algorithm for composition factors

Problem: MAXIMAL NORMAL

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

Find: A maximal normal subgroup N of G and a faithful permutation group representation for G/N or the report "G is simple".

Remark: We can invoke the algorithm for PNS to find a proper normal subgroup N of G, but N may not be maximal. To find a maximal normal subgroup we can invoke PNS with input G/N provided we have a (reasonable) permutation representation for G/N (even an infaithful one). Neumann has an example for which G/N can be represented only on a exponential size set.

Claim: MAXIMAL NORMAL can be solved in polynomial time.

Proof:

Algorithm

```
$N \leftarrow$ PNS($G$)
If $N$ is the report "G is simple" then output $N$.
Else $\{N$ is a proper normal subgroup of $G$.$\}$
    Repeat
        $j \leftarrow \min\{i \mid G^{(i)}N > G^{(i+1)}N\}$.
        $\{$So $G = G^{(j)}N)\}$
        $\Delta \leftarrow \{$cosets of $G^{(j+1)}N$ in $G = G^{(j)}N\}$.
        $\{|\Delta| \leq [G^{(j)} : G^{(j+1)}] \leq |\Omega| - j\}$
        Let $\pi : G \rightarrow Sym(\Delta)$ be the action of $G$ on $\Delta$.
        $K \leftarrow Ker(\pi)$. $\{$ So $N \leq K\}$.
        $N \leftarrow K$.
        $L \leftarrow$ PNS($G/N$).
        $\{G/N \hookrightarrow Sym(\Delta)\}$
        If $L$ is a proper normal subgroup of $G/N$ then
            $M/N \leftarrow L$.
            $N \leftarrow M$.
    Until $L$ is the report that $G/N$ is simple.
    Output $N$ and $\pi : G/N \hookrightarrow Sym(\Delta)$.
```

The above algorithm runs in polynomial time since PNS does. It easy to see that it solves MAXIMAL NORMAL.□

Problem: COMPOSITION SERIES

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

Find: A composition series $G = N_0 \triangleright N_1 \ldots \triangleright N_k = 1$ and faithful representations of the quotients $N_i/N_{(i+1)}$.

Claim: COMPOSITION SERIES 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).