LMS/EPSRC Course
Computational Group Theory
St Andrews 2013

# Permutation Groups 1: Orbits and Stabilizers

Alexander Hulpke
Department of Mathematics
Colorado State University
Fort Collins, CO, 80523, USA
http://www.math.colostate.edu/~hulpke

# Elements

Work in $G \leq S_\Omega$, typically $\Omega=\{1,...,n\}$.

Can represent elements of $G$ on the computer, each permutation requires (roughly) $n \cdot \log_2(n)$ bits. Obvious multiplication/inverse.

$n=10^5$, 1GB memory: 5000 elements

As $|S_\Omega|=n!$ we cannot assume to store many elements.

# Ground Rules

Instead represent/store subgroups by generators. At most $\log_2(|U|)$ needed. Often in praxis $<10$.

We can store *some* extra elements for data structures.

Utilize (natural) group actions:

- Permutations on points
- Matrices on Vectors
- On group elements

# Group Actions

We now assume that the group $G$ acts on the set $\Omega$ from the right: $g: \omega \to \omega^g$. (Here and in GAP always from the right.) The natural questions are to find:

ORBIT: $\omega^G$ of $\omega \in \Omega$. (Length, Elements).

STAB: Stabilizer of $\omega \in \Omega$ as subgroup of $G$.

TRANSPORTER: For $\omega, \delta \in \Omega$ find element $g \in G$ such that $\omega^g = \delta$ (or confirm that no such an element exists).

# Basic Orbit Algorithm

$G$ acts on $\Omega$. Orbit $\omega^G$ of $\omega \in \Omega$ consists of all images $\omega^g$ for $g \in G$.

Each $g$ is a product of generators (and inverses if $G$ is infinite – assume included)

Take iteratively images of all points obtained under all generators.

Cost: $|\omega^G| \cdot$#gens.

<u>**Input**</u>: $G = \langle g_1,\ldots,g_m \rangle$, acting on $\Omega$. Seed $\omega \in \Omega$.

<u>**Output**</u>: The orbit $\omega^G$.

```
begin
  Δ := [ω];
  for δ ∈ Δ do
    for i ∈ {1,...,m} do
      γ := δ^{g_i};
      if γ ∉ Δ then
        Append γ to Δ;
      fi;
    od;
  od;
  return Δ;
end.
```

# Modification: Transporters

Keeping track, we also get a *Transversal* of transporters $T[\delta]$, such that $\omega^{T[\delta]} = \delta$.

These $T[\delta]$ are also are reps. for (right) cosets of $\mathrm{Stab}_G(\omega)$ in $G$.

If $\omega^g = \delta$ and $\omega^h = \gamma$, then $\delta^x = \gamma$ for $x = g^{-1}h$, solving the general

```
begin
  Δ := [ω]; T:=[1];
  for δ ∈ Δ do
    for i ∈ {1,...,m} do
      γ := δ^{g_i};
      if γ ∉ Δ then
        Append γ to Δ;
        Append T[δ] · g_i to T;
      fi;
    od;
  od;
  return Δ;
end.
```

# Schreier's Lemma

If $\omega^a = \delta$, $\delta^b = \gamma$, $\omega^c = \gamma$, then $a \cdot b/c \in \mathrm{Stab}_G(\omega)$.

By SCHREIER's lemma (→ Max's lecture 3) such elements, formed in all ways, generate $\mathrm{Stab}_G(\omega)$:

**Lemma:** $G = \langle \mathbf{g} \rangle$ finitely gen., $S \leq G$ with $[G:S] < \infty$.
Suppose $\mathbf{r} = \{r_1, \ldots, r_n\}$ set of representatives for cosets of $S$ in $G$, such that $r_1 = 1$.
For $h \in G$ write $\underline{h} := r_i$ for the chosen representative such that $Sr_i = Sh$. Let

$$U := \{r_i g_j (\underline{r_i g_j})^{-1} \mid r_i \in \mathbf{r},\ g_j \in \mathbf{g}\}$$

Then $S = \langle U \rangle$.
U is called a set of *Schreier generators* for S.

# Modification: Stabilizer

The transversal gives coset representatives, the image of $\omega$ identifies cosets.

Thus we can form Schreier generators.

At this point $S := \langle S, T[\delta] \cdot g_i / T[\gamma] \rangle$ just produces a generating set.

```
begin
  Δ := [ω]; T:=[1]; S:=⟨1⟩;
  for δ ∈ Δ do
    for i ∈ {1,...,m} do
      γ := δ^{g_i};
      if γ ∉ Δ then
        Append γ to Δ;
        Append T[δ] · g_i to T;
      else
        S:=⟨S, T[δ] · g_i / T[γ]⟩;
      fi;
    od;
  od;
  return Δ;
end.
```

# Remarks on Performance

- Need to store whole orbit – Available memory limits scope.
- Store transversal $T$ in factored form to save memory – *Schreier vector*. (Issue: balanced tree of low depth)
- Cost of basic algorithm is dominated by test $\gamma \in \Delta$? to check for new points – Data structures.
- There is a **huge** number of Schreier generators: Index of stabilizer $\times$ # group generators.

  Usually many of them are redundant (or even trivial).

# Schreier generators

The number of Schreier generators cannot be reduced in general, as in free groups (→ Max's lectures) all are needed.

- If there is a membership test in the partial stabilizer, test each new generator whether it is already in. (Still does not produce minimal sets!)
- Let $S=\{s_1,s_2,\ldots,s_n\}$ be a generating set. A ***random subproduct*** of $S$ is a product $x=\prod_i s_i^{\epsilon_i}$ with the $\epsilon_i$ chosen independently by random from $\{0,1\}$.

# Using Random Subproducts

**Lemma:** Let $U=\langle S\rangle$ have subgroup chains of maximum length $m\leq\log_2(|U|)$. Then for every $\delta > 0$ there exists a constant $c$, such that $c\cdot m$ random subproducts generate U with probability $1-\delta$.

**Theorem** (BABAI, COOPERMAN, FINKELSTEIN, LUKS, SERESS, '95):

There is an algorithm that computes for all $\delta > 0$ in $\mathcal{O}(|S|\log m)$ operations a set of size $\mathcal{O}(m)$ that generates $U$ with probability $1-\delta$.

# Dictionaries

To test $\gamma \in \Delta$? (and to determine $T[\gamma]$) one can

- Search linearly though $\Delta$. (Orbit length $n$ requires $\mathcal{O}(n^2)$ element comparisons)
- Keep a sorted copy of $\Delta$. (needs $<$ test, $\mathcal{O}(n \log(n))$
- Determine index number for $\gamma$. (bit list, $\mathcal{O}(n)$ )

# Search in a hash table. (Hash key, almost $\mathcal{O}(n)$ )

In GAP, the `Dictionary` data type provides a uniform interface.

All nontrivial approaches require dedicated handling for each data type. (Many objects do not have unique representations!)

# Variants: Stabilizer Order

If storage or time requirements are an issue the following variants might help if $|G|$ is known:

- Known orbit length, partial stabilizer order can give early termination. If we can calculate subgroup orders, can stop if the largest proper divisor of $[G:S]$ is smaller than the orbit length.
- Use of Birthday paradox to estimate orbit length – indicate that full stabilizer is known.

More often than not I end up re-implementing an orbit algorithm instead of using a generic default...

# Consequences / Summary

The orbit algorithm and its variants let us solve

ORBIT, STABILIZER and TRANSPORTER as long as the orbit fits into memory.

By keeping track of the transversal, we write transversal elements as product of generators.

If we let $G$ act on itself this allows for element lists, centralizer, normalizer *in small groups*.

To deal with larger cases, we need to use more group theory!

# Variant: Spinning Algorithm

Take as $\Omega$ an algebraic structure, return the smallest substructure containing a seed.

Instead of an orbit, $\Delta$ is generating set for the closure. Map all elements of $\Delta$ under all group generators.

Add to $\Delta$ if image $\gamma$ not in $\langle\Delta\rangle$. (Group action preserves closure.)

Applications are e.g. normal closure, submodule.

# Group Actions in GAP

In GAP group actions are done by the operations:

- `Orbit`, `Orbits`
- `Stabilizer`, `RepresentativeAction` (Orbit/Stabilizer algorithm, sometimes backtrack, → lecture 2).
- `Action` (Permutation image of action) and `ActionHomomorphism` (homomorphism to permutation image with image in symmetric group)

The arguments are in general are:

- A group $G$. (Will act by its `GeneratorsOfGroup`.)
- A domain $\Omega$ (may be left out for `Orbit`, `Stabilizer`, but may improve performance).
- Point $\omega$, or list of point seeds for `Orbits`.

# Action functions

The last argument is an *action function* actfun($\omega$,$g$):=$\omega^g$. This is a GAP function that implements the actual action. Some predefined actions are:

- `OnPoints`: action via ^. The default if not given.
- `OnTuples`, `OnSets`: Lists or sets (i.e. sorted lists) of points.
- `OnSetsSets`, `OnSetsTuples`, etc.
- `OnRight`: right multiplication *. (e.g. on cosets)
- `OnLines`: Projective action on vectors scaled to have first nonzero entry 1.
- `OnSubspacesByCanonicalBasis`: Subspaces, given as list of RREF basis vectors.
- `Permuted`: Permuting list entries.

# Optional Arguments

$G$ may act via a homomorphism $\varphi$. (Say, a matrix group acting on enumerated vectors.) One can compute (in particular stabilizers) by giving two further list arguments:

**gens** A list of group generators,
**imgs** Images of these generators under $\varphi$.

Action Homomorphisms by default have codomain $S_n$. For large $n$ this is inefficient. Append the string argument "`surjective`" to force the codomain equal to the image.

**Action on cosets:** Internal use of `PositionCanonical` (position of a *standard* equivalent object) allows:

`ActionHomomorphism(G,RightTransversal(G,U),OnRight, "surjective")`; to get the action on the cosets of $U$ in $G$.

`FactorCosetAction` produces the same result.

LMS/EPSRC Course
Computational Group Theory
St Andrews 2013

# Permutation Groups 2: Stabilizer Chains

Alexander Hulpke
Department of Mathematics
Colorado State University
Fort Collins, CO, 80523, USA
http://www.math.colostate.edu/~hulpke

# Blocks (AKA imprimitivity)

G acting (transitively) on $\Omega$, a ***block system*** is a $G$-invariant partition $\mathscr{B}$ of $\Omega$, i.e. $\Omega = \cup_{B \in \mathscr{B}} B$ but $B_i \cap B_j = \emptyset$. Thus G also acts on $\mathscr{B}$.

**Basic facts:**

- All blocks have the same size.
- Trivial block systems: $\{\Omega\}$ and singleton sets. If only these two: ***primitive*** (otherwise ***imprimitive).***
- $b$ blocks of size $a$ ($n=a \cdot b$) iff $G \leq S_a \wr S_b$.
- Block systems are in bijection with subgroups $\mathrm{Stab}_G(\omega) \leq S \leq G$, $S$ is stabilizer of block with $\omega$.
- If $N \triangleleft G$ the orbits of N form a block system.

# Orbit with Normal Subgroup

If we know $N \triangleleft G$, we can reduce:

- Determine the orbit $\Delta$ of $\omega$ under $N$.
- Determine the orbit of the set $\Delta$ under $G$.

  The image of a single point determines whether an image is new, if so whole block image is new.

  Cost is that of two smaller orbit algorithms: $|\Delta|+|\Delta^G|$ instead of $|\Omega|=|\Delta|\cdot|\Delta^G|$.
- For Stabilizer, take $\mathrm{Stab}_N(\omega)$ and correct $g \in \mathrm{Stab}_G(\Delta)$ with $n \in N$: $\omega^g = \omega^n$, $\omega^{g/n} = \omega$.

More in Bettina's lecture.

# Some Fundamental Tasks

For groups of permutations of degree up to a few $10^6$, order easily $10^9$ (so the orbit approach is infeasible), we want to solve:

ORDER: find the order of a group. (Implies element membership test.)

HOMOMORPHISM: decompose element as generator product. (Rewriting problem.)

We want to identify the group STRUCTURE, possibly find isomorphisms.

Also centralizers, normalizers if index is huge.

# Use Subgroups

The principal idea now is to use subgroups/cosets to factor the problem: As $|G|=|U|\cdot[G:U]$ this logarithmizes the problem.

Suitable subgroups: Point stabilizers $U=\mathrm{Stab}_G(\omega)$, index at most $|\Omega|$.

We can iterate this process for $U$.

**Caveat:** This works for any group with a natural action (matrix, automorphism, etc.) but often the problem is that $[G:\mathrm{Stab}_G(\omega)]$ is not small.

Case in point: $\mathrm{GL}_n(q)$, orbit length $q^n$.

# Stabilizer Chains

Let $G \leq S_\Omega$. A list of points $B=(\beta_1,\ldots,\beta_m)$, $\beta_i \in \Omega$ is called a ***base***, if the identity is the only element $g \in G$ such that $\beta_i^g = \beta_i$ for all $i$.

The associated ***Stabilizer Chain*** is the sequence

$$G = G^{(0)} > G^{(1)} > \ldots > G^{(m)} = \langle 1 \rangle$$

defined by $G^{(0)} := G$, $G^{(i)} := \mathrm{Stab}_{G^{(i-1)}}(\beta_i)$. (Base guarantees that $G^{(m)} = \langle 1 \rangle$.)

Note that every $g \in G$ is defined uniquely by base images $\beta_1^g,\ldots,\beta_m^g$. (If $g,h$ have same images, then $g/h$ fixes base.)

# Base Length

The base length $m$ often is short ($m \leq \log_2(|G|)$). In practice often $m < 10$.

We say that $G$ is ***short-base*** if $\log|G| \leq \log^c |\Omega|$

Bounds on base length have been studied in theory. If there is no short base the groups must be essentially $A_n$ and relatives.

Same concept also possible for other kinds of groups and mixed actions, but then no good orbit length/base length estimates.

# Data structure

We will store for a stabilizer chain:

- The base points $(\beta_1,\ldots,\beta_m)$.
- Generators for all stabilizers $G^{(i)}$. (Union of all generators is ***strong generating set***, as it permits reconstruction of the $G^{(i)}$.) Data structure thus is often called **B**ase and **S**trong **G**enerating **S**et.
- The orbit of $\beta_i$ under $G^{(i-1)}$ and an associated transversal for $G^{(i)}$ in $G^{(i-1)}$ (possibly as *Schreier tree*).

Storage cost thus is $\mathcal{O}(m \cdot |\Omega|)$

# Consequences

- Group order: $G = [G^{(0)}:G^{(1)}]\cdot[G^{(1)}:G^{(1)}]\cdot\ldots [G^{(m-1)}:G^{(m)}]$ and thus $G=\prod_i |\beta_i^{G(i-1)}|$.
- Membership test in G for $x \in S_\Omega$:

1. Is $\omega = \beta_1^x \in \beta_1^G$? If not, terminate.
2. If so, find transversal element $t \in G^{(0)}$ such that $\beta_1^t=\beta_1^x$.
3. Recursively test membership of x/t (stabilizing $\beta_1$) in $G^{(1)}$.
   (Or test x/y=( ) in last step.)

# More Consequences

Bijection $g \in G \Leftrightarrow$ base image $(\beta_1^g, \beta_2^g, \ldots)$.

- Enumerate $G$, equal distribution random elements.
- Write $g \in G$ as product in transversal elts.
- Write $g \in G$ as product in strong generators.
- Write $g \in G$ as product in generators of $G$. (Caveat: Long words)
- Chosen base: Find stabilizers, transporter elements, for point tuples.

# Schreier-Sims algorithm

SIMS' (1970) primary idea is to use a membership test in a partial stabilizer chain to reduce on the number of Schreier generators.

Basic structure is a partial stabilizer, i.e. a subgroup $U \leq G^{(i-1)}$ given by generators and a base-point orbit $\beta^U$ with transversal elements (products of the generators of U).

The basic operation now is to pass an element $x \in G^{(i-1)}$ to this structure and to consider the base point image $\omega = \beta^x$.

# Base point image $\omega = \beta^x$

- If $\omega \in \beta^U$, transversal element $t \in U$ such that $\beta^t = \omega$. Pass $y = x/t$ to the next lower partial stabilizer $\leq G^{(i)}$.
- If $\omega \notin \beta^U$, add $x$ to the generating set for $U$ and extend the orbit of $\beta$. All new Schreier generators $y \in \mathrm{Stab}_U(\beta)$ are passed to next partial stab. $\leq G^{(i)}$.
- If no lower stabilizer was known, test whether the $y$ was the identity. If so just return. (Successful membership test.)
- Otherwise start new stabilizer for generator $y$ and the next base point. (Pick a point moved by $y$).

# Homomorphisms

Embed permutation group $G$ into direct product $D=G\times H$. A homomorphism $\varphi:G\rightarrow H$ can be represented as $U\leq D$ via

$$U=\{(g,h) \in G\times H \mid g^{\varphi}=h\}$$

Build a stabilizer chain for $U$ using only the $G$-part.

Then decomposing $g \in G$ using this chain produces an $H$-part that is $g^{(\varphi-1)}$.
Use this to evaluate arbitrary homomorphisms.

# Kernels, Relators

Vice versa, let $\varphi: H \to G$ and
$U = \{(g,h) \in G \times H \mid h^{\varphi} = g\}$.

Form a stabilizer chain from generators of $U$, using the $G$-part.
The elements sifting through this chain (trivial $g$-part) are generators for $\ker \varphi$.

If $H$ is a free group, this yields a presentation for $G$.

# Quandry

Deterministic Algorithm. Polynomial (in the degree $n=|\Omega|$) runtime, but larger exponent ($n^3$ if Schreier tree used).

The cause is the processing of all (mostly redundant) Schreier generators.

In practice not feasible if $n$ is big ( $>1000$). For short base ($\log|G| \leq \log^c n$) we would like nearly linear time $\mathcal{O}(n \log^c n)$, best possible

# Wrong Results are Cheap

Use only *some* generators (random subset, better: random subproducts). Wrong data structure. But:

- Error results in chain that claims to be too small - can detect if group order is known.
- Error analysis: A random element of $G$ fails sifting in wrong chain with probability 1/2 - guarantee arbitrary small error probability.
- But we can verify that a chain is correct:
  - Combinatorial Verification (Sims, see SERESS' book)
  - Presentation from stabilizer chain. Verify that group fulfills it. If too small, some relators fail to be. (Todd-Coxeter-Schreier-Sims; or Recognition see lecture 3.)

# Other Actions

Every finite group is a permutation group in suitable actions. (E.g. matrices on vectors.) Same methods apply there.

It is possible to use different actions (e.g. matrix group on subspaces and on vectors)

**But:** Orbit lengths can be unavoidably huge if there are no subgroups of small index.

Approach can be useful for well-behaved groups. Not a panacea, but part of matrix group recognition.

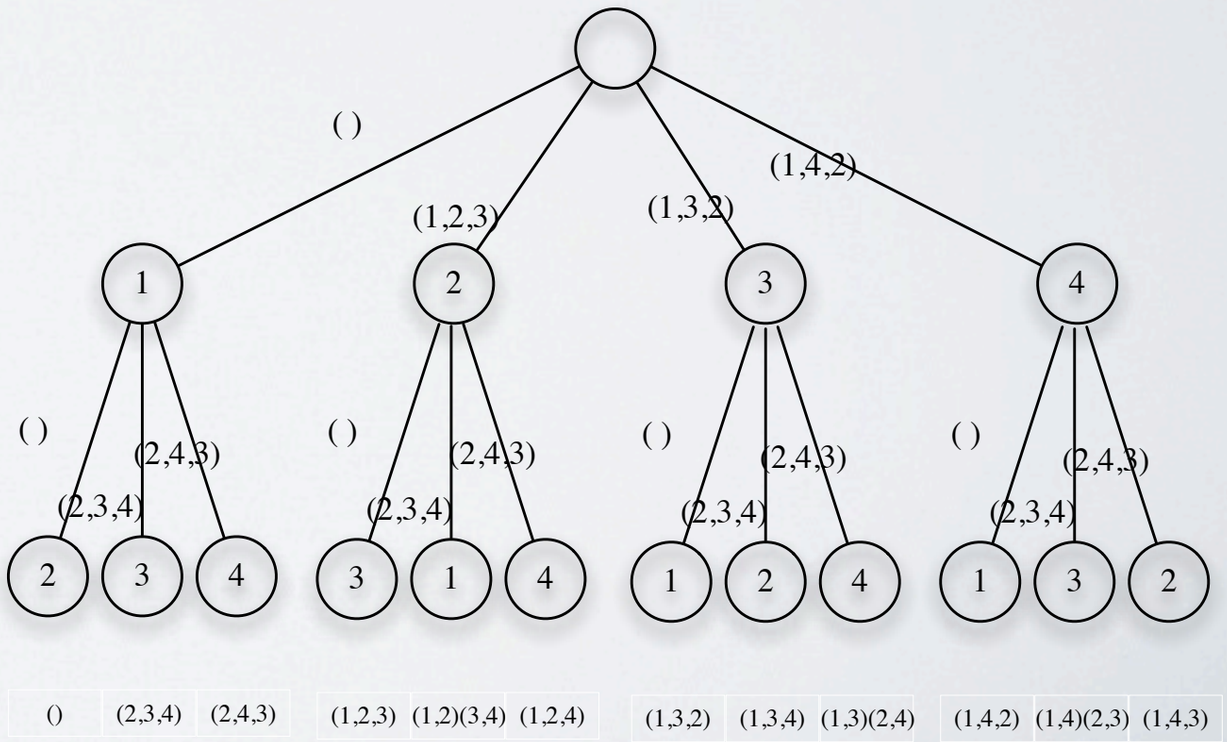
# Backtrack

A stabilizer chain lets us consider the elements of $G$ as leafs on a tree, branches corresponding to base point images.

Traverse the tree (depth first) by enumerating all possible base images. Find group elements with particular desired property.

Exponential run time but good in practice.

E.g.: Centralizer, Normalizer, Set Stab., Intersection, Conjugating elts., ...



# Search Tree Pruning

It is crucial to reduce the search space down from $|G|$ to a manageable size. Tools:

**Algebraic structure:** Solution set forms a subgroup (if stabilizer) or double coset (if transporter). E.g., all elements mapping $\omega$ to $\delta$ lie in $\mathrm{Stab}_G(\omega)\cdot g\cdot\mathrm{Stab}_G(\delta)$ where $\omega^g=\delta$.

The closure properties of the structure mean that the existence of some elements implies existence of others.

For simplicity, assume that we are aiming to find $S=\mathrm{Stab}_G(\omega)$.

# Double Coset Pruning

Assume we have found (or were given) some elements of $S$, generating subgroup $U$. (Hard part is to prove there are no further ones.)

If $g \in G$, then either all or no elements of $UgU$ will be in S. Sufficient to test one.

Criterion: Only test $g$ if it is minimal in $UgU$. (lexicographically as lists of base images.)

Minimal in $UgU$ is hard. Instead use minimal in $Ug$ and in $gU$ (necessary, not sufficient). Restrict choice of possible base images.

# Problem-specific Pruning

The real power of backtrack comes with pruning methods that are specific to the problem to be solved. For example:

- An element centralizing a permutation must map cycles to cycles of the same length. Images of the first cycle point thus are limited. Once the image $\omega^g$ of a first cycle point is chosen, the images of **all** other points in the cycle are given.
- An element normalizing a subgroup *U* must preserve the orbits of *U*. When also fixing the point $\omega$, one must preserve the orbits of $\mathrm{Stab}_U(\omega)$ (these are called orbitals).

# Base Change

For efficiency , it is helpful to use a base that causes problem-specific prunings to apply early.

E.g. when centralizing an element, choose the first base point in a cycle of longest length (as the choice of one point image determines all others).

There used to be algorithms that performed a ***base change***, i.e. computed a new stabilizer chain from an old one but with different base.

Modern, randomized, Schreier-Sims algorithms are so fast that is is usually easiest to just compute a new chain for the desired base.

# Partition backtrack

Partition Backtrack (McKay, Leon, Theissen,...) is a convenient way to process the different kinds of pruning.

The algorithm maintains a partition (list of points) of Ω, indicating possible images of the base points. Tree root=(Ω), leaves=1point cells.

Selection of base images and pruning conditions are partition refinements, done by intersecting with particular partitions, such as (img, rest) or orbits of a subgroup.

LMS/EPSRC Course
Computational Group Theory
St Andrews 2013

# Permutation Groups 3: Composition Series

Alexander Hulpke
Department of Mathematics
Colorado State University
Fort Collins, CO, 80523, USA
http://www.math.colostate.edu/~hulpke

# Towards Structure

A crucial tool on the way towards determining a permutation group's structure is the composition series.

Purposes include:

- Decomposing the group
- As a Tool for other tasks
- Showcase of new class of structural methods
- Verification of stabilizer chains.

# Homomorphisms

As a basic tool we want to be able for a permutation group $G$ to either:

- Find a homomorphism $\varphi$ on $G$ with "*smaller*" image. Or:
- Prove that $G$ is simple.

**Natural Source of Homomorphisms:** Group Actions, in particular from permutation action.

If $G$ is intransitive on $\Omega$: Action on Orbit

If $G$ is transitive, imprimitive on $\Omega$: Action on blocks. (Find block systems by starting with block seed, Union of images.)

# Primitive Groups

Otherwise $G$ is ***primitive***. The O'NAN-SCOTT theorem describes the possible structure.

Key component: The ***Socle*** $\mathrm{Soc}(G)$, subgroup generated by all minimal normal subgroups.

**Lemma** $\mathrm{Soc}(G)$ is direct product of minimal normal subgroups.

<u>Proof:</u> Take $M \leq \mathrm{Soc}(G)$, $M \triangleleft G$ maximal with this property. If $M \neq \mathrm{Soc}(G)$ there exists $N \triangleleft G$, minimally normal, $N \not\leq M$. Thus $M \cap N = \langle 1 \rangle$ and $\langle M, N \rangle = M \times N \leq \mathrm{Soc}(G)$ is larger, Contradiction.

# Socle of a Primitive Group

Let $N \triangleleft G$ minimal normal.

**Remember:** Orbits of $N$ are blocks, thus in primitive $G$ we have that $N$ is transitive.

Nontrivial $C_G(N) \triangleleft G$ will be normal, transitive.

**Lemma** $N \leq S_\Omega$ transitive. Then $C = C_{S_\Omega}(N)$ is *semiregular* (i.e. for all $\omega \in \Omega$: $\mathrm{Stab}_C(\omega) = 1$.)

<u>Proof:</u> Let $c \in \mathrm{Stab}_C(\omega)$, $\delta \in \Omega$. Then there is $g \in N$ such that $\delta = \omega^g = \omega^{(cg)} = \omega^{(gc)} = \delta^c$, thus $c \in \mathrm{Stab}_C(\delta)$ for every $\delta$. Thus $c = 1$.

# Socle Structure

**Theorem** Let $G$ primitive on $\Omega$. $S=\mathrm{Soc}(G)$. Then either

- a) $S$ is minimally normal, ***or***
- b) $S=N\times M$ with $N,M\lhd G$ minimal, $N\cong M$ nonabelian.

**Proof:** If $S$ is not minimally normal then $S=N\times M$, $M\leq C_G(N)$ and $N\leq C_G(M)$. Both groups are transitive, semiregular, thus $|N|=|\Omega|=|M|$, both nonabelian.

For $n\in N$ exists unique $m(n)\in M$ such that $(1^n)^{m(n)}=1$. Then $\varphi:N\to M$, $n\to m(n)$ is isomorphism, as for $k,n\in N$:

$$1^{(k\cdot n\cdot m(k)\cdot m(n))}=1^{k\cdot m(k)\cdot n\cdot m(n)}=((1^k)^{m(k)})^{n\cdot m(n)}=1^{n\cdot m(n)}=1 \qquad \square$$

We thus have that $\mathrm{Soc}(G)\cong T^{\times m}$ with $T$ simple. We say that $\mathrm{Soc}(G)$ is ***homogeneous of type*** $T$.

# Abelian Socle

If $S$=Soc($G$ ) is abelian, it is an elementary abelian regular normal subgroup.

A point stabilizer $U$ =Stab$_G$(ω) intersects trivially with $S$, thus $G \leq \mathrm{AGL}_n(p)$ is an affine group (linear+translation).

Submodules yield blocks, thus $S$ is irreducible under conjugation by $U$ (or $G$ ).

(Finding $S$ requires some work, algorithm exists.)

Vice versa irreducible action of a group $U$ yields primitive group $U \ltimes C_p{}^n$.

# Nonabelian Socle

If the Socle $S = \mathrm{Soc}(G) \cong T^{\times m}$ is not abelian then $C_G(S) = \langle 1 \rangle$.

The action of $G$ on $S$ thus is faithful. Therefore (up to isomorphism) $G \leq \mathrm{Aut}(\mathrm{Soc}(G))$.

$T$ is simple nonabelian, $\mathrm{Aut}(T^{\times m}) = \mathrm{Aut}(T) \wr S_m$.

The action on the $m$ direct factors of $S$ is a homomorphism with nontrivial kernel.

More detailed description of the possible actions is given by the O'NAN-SCOTT Theorem. (Cf: ASCHBACHER's theorem, → Derek's lectures)

# O'Nan-Scott Theorem

$G$ primitive, $|\Omega|=n$. Let $S=\mathrm{Soc}(G)= T^{\times m}$. Then:

**Affine:** $G \leq \mathrm{AGL}_n(q)$.

**Almost simple:** $m=1$ and $H \triangleleft G \leq \mathrm{Aut}(S)$.

**Diagonal:** $m \geq 2$ and $n=|T|^{m-1}$. Further, $G \leq V=(T \wr S_m).\mathrm{Out}(T)$ in diagonal action.

**Product Action:** $m=rs$ with $s>1$. $G \leq W=A \wr B$ in product action, $A \leq S_d$ primitive, not regular, $B \leq S_s$ transitive. Thus $n=ds$.

**Twisted wreath:** S regular and $n=|T|^m$. $G_\omega$ isomorphic transitive subgroup of $S_m$. ($n \geq 60^6$.)

# O'Nan-Scott Theorem

$G$ primitive, $|\Omega|=n$. Let $S=\mathrm{Soc}(G)=T^{\times m}$. Then:

**Affine:** $G \leq \mathrm{AGL}_n(q)$.

**Almost simple:** $m=1$ and $H \triangleleft G \leq \mathrm{Aut}(S)$.

**Diagonal:** $m \geq 2$ and $n=|T|^{m-1}$. Further, $G \leq V=(T \wr S_m).\mathrm{Out}(T)$ in diagonal action.

**Product Action:** $m=rs$ with $s>1$. $G \leq W=A \wr B$ in product action, $A \leq S_d$ primitive, not regular, $B \leq S_s$ transitive. Thus $n=ds$.

**Twisted wreath:** S regular and $n=|T|^m$. $G_\omega$ isomorphic transitive subgroup of $S_m$. ($n \geq 60^6$.)

# O'Nan-Scott Theorem

$G$ primitive, $|\Omega|=n$. Let $S=\mathrm{Soc}(G)=T^{\times m}$. Then:

**Affine:** $G \leq \mathrm{AGL}_n(q)$.

**Almost simple:** $m=1$ and $H \triangleleft G \leq \mathrm{Aut}(S)$.

**Diagonal:** $m \geq 2$ and $n=|T|^{m-1}$. Further, $G \leq V=(T \wr S_m).\mathrm{Out}(T)$ in diagonal action.

**Product Action:** $m=rs$ with $s>1$. $G \leq W=A \wr B$ in product action, $A \leq S_d$ primitive, not regular, $B \leq S_s$ transitive. Thus $n=ds$.

**Twisted wreath:** S regular and $n=|T|^m$. $G_\omega$ isomorphic transitive subgroup of $S_m$. ($n \geq 60^6$.)

# O'Nan-Scott Theorem

$G$ primitive, $|\Omega|=n$. Let $S=\mathrm{Soc}(G)=T^{\times m}$. Then:

**Affine:** $G \leq \mathrm{AGL}_n(q)$.

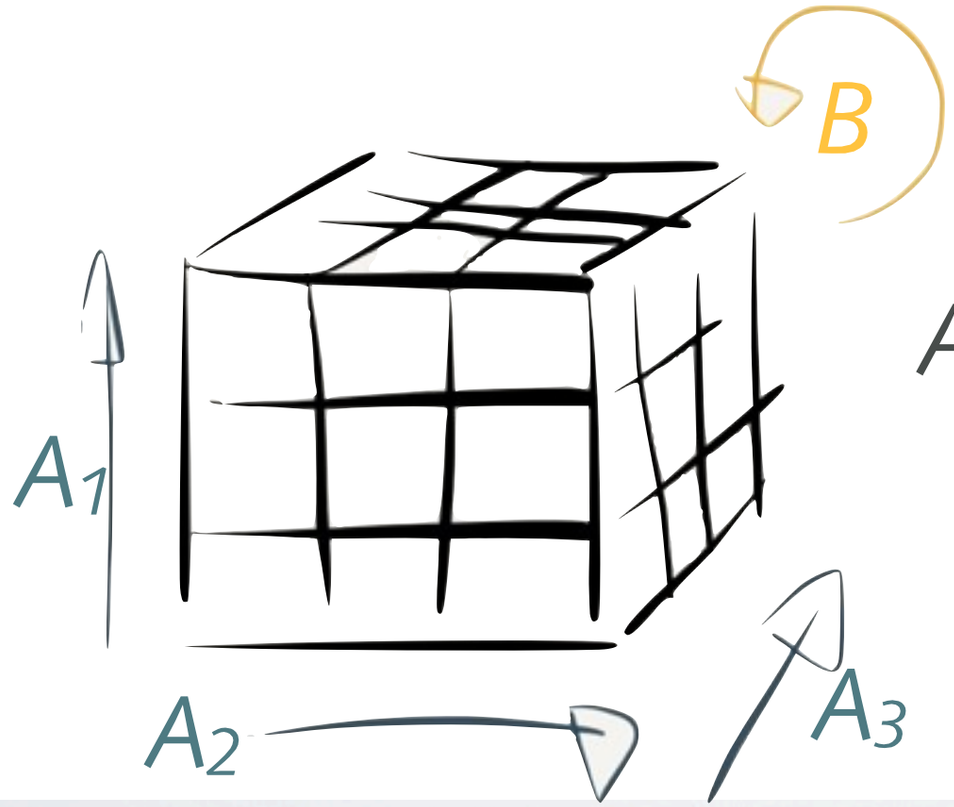
**Almost simple:** $m=1$ and $H \triangleleft G \leq \mathrm{Aut}(S)$.

**Diagonal:** $m \geq 2$ and $n=|T|^{m-1}$. Further, $G \leq V=(T \wr S_m).\mathrm{Out}(T)$ in diagonal action.

$$\mathrm{Stab}_S(\omega)=\{ (t,t,\ldots,t) \mid t \in T \}$$

isomorphic transitive subgroup of $S_m$. ($n \geq 60^6$.)

# O'Nan-Scott Theorem




$A \wr B = (A \times A \times \ldots \times A) \rtimes B$

**Product Action:** $m = rs$ with $s > 1$. $G \leq W = A \wr B$ in product action, $A \leq S_d$ primitive, not regular, $B \leq S_s$ transitive. Thus $n = ds$.

**Twisted wreath:** S regular and $n = |T|^m$. $G_\omega$ isomorphic transitive subgroup of $S_m$. ($n \geq 60^6$.)

# O'Nan-Scott Theorem

$G$ primitive, $|\Omega|=n$. Let $S=\mathrm{Soc}(G)= T^{\times m}$. Then:

**Affine:** $G \leq \mathrm{AGL}_n(q)$.

**Almost simple:** $m=1$ and $H \triangleleft G \leq \mathrm{Aut}(S)$.

**Diagonal:** $m \geq 2$ and $n=|T|^{m-1}$. Further, $G \leq V=(T \wr S_m).\mathrm{Out}(T)$ in diagonal action.

**Product Action:** $m=rs$ with $s>1$. $G \leq W=A \wr B$ in product action, $A \leq S_d$ primitive, not regular, $B \leq S_s$ transitive. Thus $n=ds$.

**Twisted wreath:** S regular and $n=|T|^m$. $G_\omega$ isomorphic transitive subgroup of $S_m$. ($n \geq 60^6$.)

# O'Nan-Scott Theorem

$G$ primitive, $|\Omega|=n$. Let $S=\mathrm{Soc}(G)= T^{\times m}$. Then:

**Affine:** $G \leq \mathrm{AGL}_n(q)$.

**Almost simple:** $m=1$ and $H \triangleleft G \leq \mathrm{Aut}(S)$.

**Diagonal:** $m \geq 2$ and $n=|T|^{m-1}$. Further, $G \leq V=(T \wr S_m).\mathrm{Out}(T)$ in diagonal action.

**Product Action:** $m=rs$ with $s>1$. $G \leq W=A \wr B$ in product action, $A \leq S_d$ primitive, not regular, $B \leq S_s$ transitive. Thus $n=ds$.

**Twisted wreath:** S regular and $n=|T|^m$. $G_\omega$ isomorphic transitive subgroup of $S_m$. ($n \geq 60^6$.)

# Use in Classifications

Reduces to maximal subgroups of simple groups (**C**lassification of **F**inite **S**imple **G**roups).

Information allows for explicit lists:

- ≦50 (Sims, 1970s)
- ≦1000 nonaffine (Dixon & Mortimer, 1989)
- ≦255 affine (Theissen, 1997)
- ≦1000 affine (Roney-Dougal & Unger, 2000)
- ≦$3^8$ solvable (Eick & Hoefling, 2004)
- ≦4095 (Roney-Dougal, Quick, Coutts, 2012)

# Proof Sketch

[Dixon, Mortimer: Permutation Groups, GTM163]

Assume Socle $S = T \times \ldots \times T$ nonabelian.

$S$ acts transitively. Let $U = \mathrm{Stab}_S(1)$. $\alpha : U \to T_1$

- If $U$ trivial then twisted wreath. Degree $\geqq 60^6$
- If $U^\alpha \neq T$, then $U = U^\alpha \times \ldots \times U^\alpha$, product action.
- Otherwise $U$ is subdirect product (thus direct product) of $T$'s. Consider $V = U \cap \ker \alpha$. If $V$ trivial then diagonal type.
- Otherwise product action of almost simple or diagonal type.

# Finding the Socle

To find Soc($G$) for a primitive group we use

**Schreier's Conjecture:** $T$ finite, simple, nonabelian. Then $\mathrm{Out}(T)=\mathrm{Aut}(T)/T$ is solvable of derived length at most 3. Proof by inspection of all cases (CFSG).

**Lemma** Let $U \leq G$ be a 2-Sylow subgroup and $N=\langle Z(U)\rangle_G$ (normal closure). Then $S=N'''$.

<u>Proof:</u> As $2 \mid |T|$, $U$ has elements in each copy of $T$. So $Z(U)$ cannot move any $T$, thus $Z(U) \leq \mathrm{Aut}(T)^{\times m}$. As $1 \neq Z(U)$, also $T^{\times m} \leq \langle Z(U)\rangle_G \leq \mathrm{Aut}(T)^{\times m}$. But then the derived series of $\langle Z(U)\rangle_G$ ends in $T^{\times m}$.

# Almost Simple Case

In the almost simple case $m$=1 and the action on the socle factors does not give any reduction.

However Out($T$) is small and solvable, so it is easy to construct a homomorphism with kernel $T$.

Remaining case is that of simple group $T$. In this case use constructive recognition (Effective isomorphism to *natural* copy, → Derek's lectures) to identify $T$ as a simple group.

In many cases order/degree of a primitive group can establish simplicity or identify the isomorphism type if simple.

# Composition Series

Given a permutation group $G$, we search for a homomorphism $\varphi$ with smaller image if one exists. Recurse to Image and Kernel (if nontrivial).

Pulling the kernels back through previous homomorphisms gives a composition series of $G$.

We can combine presentations of the simple factors to obtain a presentation of $G$.

(Respectively, presentations of the images to obtain kernel generators.)

# Combining Presentations

Let $N \triangleleft G$ with presentations

$N \cong \langle a_1,\ldots,a_l \mid r_1(\underline{a}),r_2(\underline{a}),\ldots\rangle$

$G/N \cong \langle b_1,\ldots,b_m \mid s_1(\underline{b}),s_2(\underline{b}),\ldots\rangle$

$$\begin{tikzcd}
F(\underline{\boldsymbol{b}}) \arrow[r] \arrow[d, dash] & G/N \arrow[d, dash] & G \arrow[l] \arrow[d, dash] & & \\
\langle s_j(\underline{\boldsymbol{b}})\rangle_j \arrow[r] \arrow[d, dash] & \langle 1\rangle \arrow[d, dash] & N \arrow[l] & & F(\underline{\boldsymbol{a}}) \arrow[ll] \arrow[d, dash] \\
 & & & & \langle r_j(\underline{\boldsymbol{a}})\rangle_j \arrow[d, dash] \\
\langle 1\rangle & & & & \langle 1\rangle
\end{tikzcd}$$

$$\begin{tikzcd}
b_i \arrow[r] & Ng_i & g_i \arrow[l] \\
s_j(\underline{\boldsymbol{b}}) \arrow[r] & w_j(\underline{\boldsymbol{n}}) & n_i \arrow[l] & a_i \arrow[l] \\
 & & \langle 1\rangle & \langle r_j(\underline{\boldsymbol{a}})\rangle_j \arrow[l] \\
 & & 1 & r_j(\underline{\boldsymbol{a}}) \arrow[l]
\end{tikzcd}$$

Let $n_i \in N$ image of $a_i$ in $N$.

$g_i \in G$, $Ng_i$ image of $b_i$ in $G/N$.

Find words $v_{i,j}$, $w_j$ such that
$w_j(n_1,\ldots,n_l)=n_i{}^{g_j} \in N$ and
$w_j(n_1,\ldots,n_l)=s_j(g_1,\ldots,g_m) \in N$. Then

$\langle a_1,\ldots,a_l,b_1,\ldots,b_m \mid r_1(a),r_2(a),\ldots,\ s_j(\underline{b})=w_j(\underline{a}),\ a_i{}^{b_i}=v_i(\underline{a})\rangle$

is a presentation for $G$.

**Proof:** Relators define $G$ with normal $N$, factor $G/N$.

# Verification of Chain

The resulting presentation for $G$ is based on the composition factors recognized.

**Back to Random Schreier-Sims:**
Composition series, with randomized stabilizer chains for $G$ and factors.

If any randomized calculation failed, the resulting presentation will describe a *smaller* group. Detect this by evaluating the presentation on $G$. Otherwise we know $|G|$.

So we can certify a stabilizer chain for $G$.

# Randomized Algorithms

A *Monte Carlo* algorithm can give wrong result (with selectable probability $\epsilon$),

A **Las Vegas** algorithm, in addition tests for correctness, never returns a wrong result but failure (or unbounded run time) possible.

The randomized stabilizer chain calculation is Monte Carlo. Verification makes it Las Vegas. The only question is run time.

Randomized stabilizer chain is nearly linear $\mathcal{O}(n \log^c n)$. Can one sustain this?

# Upgrade to Las Vegas

To maintain good run time for the verification, we need algorithms for

- Homomorphisms for decomposing to primitive factors and for splitting primitive factors.
- Constructive recognition of simple permutation groups.
- Write down a presentation for the simple factors. (Also used in recognition)

To maintain complexity, runtime for simple $T$ must be $\mathcal{O}(\log^c |T|)$. This means presentations must be of length $\mathcal{O}(\log^c |T|)$.

# Short Presentations

Such short presentations are known for:

- Cyclic Groups (trivial)
- Sporadic Groups (trivial)
- Alternating Groups (Coxeter, Moser 1972)
- Lie Type of rank >1 (Steinberg 1962, Babai, Goodman, Kantor, Luks, Pálfy, 1997)
- $PSL_2(q)$ (Todd 1936)
- Suzuki groups (Suzuki 1964)
- $PSU_3(q)$ (H., Seress 2001)

*Only* the Ree groups ${}^2G_2(q)$ remain ...

# Permutation Group Recognition

The decomposition of a permutation representation is exactly the analog of matrix group recognition, decomposing with Aschbacher's theorem.

As some reductions of matrix groups reduce to permutation groups, one can really consider this recognition as the *same* process working on permutation groups and matrix groups.

The `recog` package in GAP (NEUNHÖFFER, SERESS) does exactly this.

LMS/EPSRC Course
Computational Group Theory
St Andrews 2013

# Permutation Groups 4: The Solvable Radical/ Trivial Fitting Method

Alexander Hulpke
Department of Mathematics
Colorado State University
Fort Collins, CO, 80523, USA
http://www.math.colostate.edu/~hulpke

# Structural Computations

We typically don't just want to determine group order or composition structure, but higher level information such as conjugacy classes, subgroups, isomorphisms, etc.

The algorithms built for this over the last 15-20 years use the *Solvable Radical* (or *Trivial Fitting*) method. (HOLT, CANNON, H.) Generalizing earlier algorithms for solvable groups this tries to do as much as possible by linear algebra, i.e. with elementary abelian normal subgroups.

The crucial ingredient is the *Solvable Radical*, $R$=Rad($G$) the largest solvable normal subgroup.

# Structure Analysis

Let $R$=Rad($G$). Then $G/R$ has no solvable normal subgroup. (So it has no nilpotent normal subgroups and thus a trivial Fitting subgroup.)

Every minimal normal subgroup of $G/R$ is a direct product of simple groups. So is $S^*/R = \mathrm{Soc}(G/R)$.

The action of $G/R$ on $S^*/R$ is faithful, thus (up to isomorphism) $G/R \leq \mathrm{Aut}(S^*/R) = \mathrm{Aut}(T_1 \times \ldots \times T_k)$ with $T_i$ simple, nonabelian.

If all $T$ are isomorphic, then $\mathrm{Aut}(T \times \ldots \times T)$ $= \mathrm{Aut}(T) \times \ldots \times \mathrm{Aut}(T) : S_k = \mathrm{Aut}(T_i) \wr S_k$.

Otherw. direct product of such wreath products.

# Permutation Images

$G/R < \mathrm{Aut}(S^*/R) = \times_i \mathrm{Aut}(T_i) \wr S_{k_i}$.

The $S_k$-part (permutation of socle factors) yields a homomorphism $G/R \to S_k$ with kernel $Pker := K \geq S^*$. By (the proof of) SCHREIER's conjecture $K/S^*$ is solvable.

Action on $S^*/R$ induces an action homomorphism $G \to \mathrm{Aut}(S^*/R)$ that we can use to represent $G/R$ as (permutation) group of moderate degree. Methods similar as with composition series.

# Using Solvable-Radical

A Solvable-Radical algorithm now first computes the result in $G/R$ using its special structure.

Then choose a series of $G$-normal subgroups

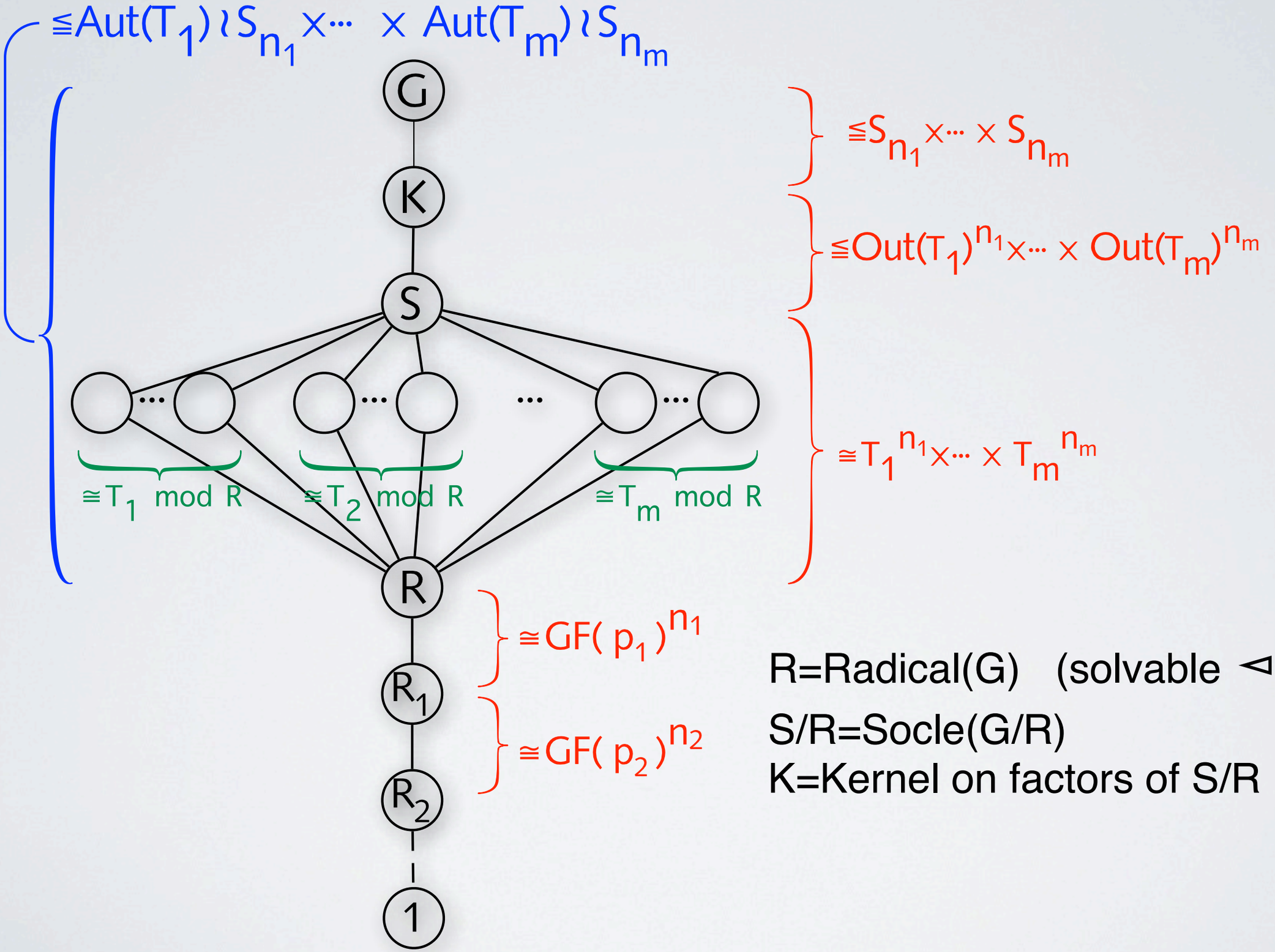
$$R = R_0 > R_1 > R_2 > \ldots > R_l = \langle 1 \rangle.$$

with $R_i / R_{i+1} \cong \mathrm{GF}(p_i)^{n_i}$.

Step by step, lift the result from $G/R_i$ to $G/R_{i+1}$. In each step use elementary abelian normal subgroup. Usually orbit calculations on this.

Work with subgroups, elements of $G$ to represent factor groups.

# Structure summary




R=Radical(G) (solvable $\lhd$ )

S/R=Socle(G/R)

K=Kernel on factors of S/R

# A PCGS for the Radical

A variant of Schreier-Sims (also by SIMS, 1990) for solvable groups finds a stabilizer chain *and* a PCGS compatible with such a series:

- Find $g \in R$ in an elementary abelian normal $N \triangleleft G$ as suitable commutators/powers of generators.
- Find $N$ as normal closure under $G$ (this will be the next $R_i$ from the bottom) by adding conjugates of $g$.
- Repeat for $G/N$ if $G$ is not reached.

The elements $g$ (+ conjugates) found form PCGS for $R$.

In each step we add one *normalizing* element to an existing group increasing the order by $p$.

Thus an existing stabilizer chain extends the orbit in exactly one layer by $p$, using normal subgroup orbit variant.

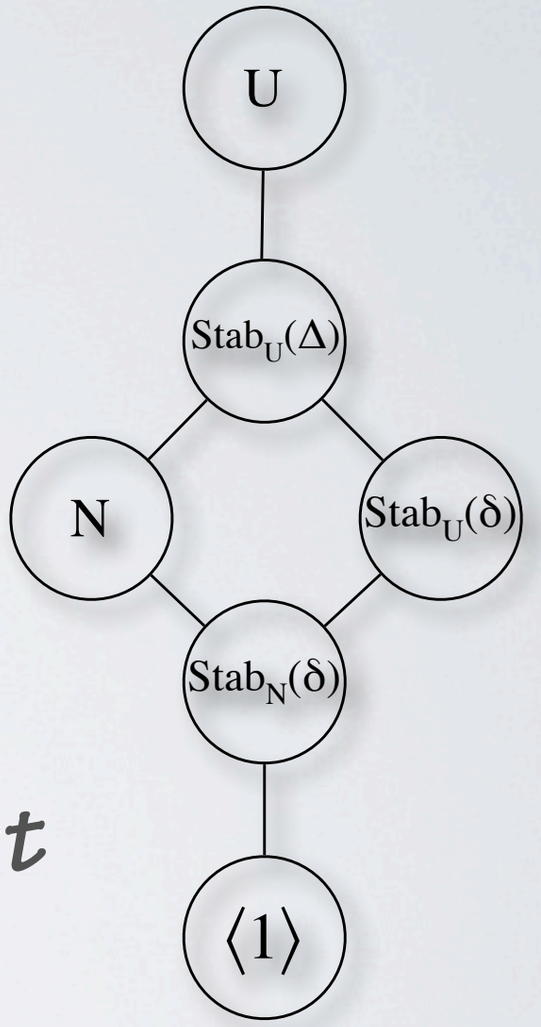
Decomposition with stabilizer chain yields PCGS exponents.

# $N$-orbits

When computing group actions, one can use the the normal subgroup $N = R_i / R_{i+1}$ whose orbits form blocks.

Instead of stabilizing a point $\omega$ under $U$, first stabilize the corresponding block $\Delta = \omega^N$. Let $A = \mathrm{Stab}_U(\Delta)$ be this stabilizer.

Then for every $a \in A$, there exists $n \in N$ such that $\omega^a = \omega^n$, so the ***corrected element*** $a / n \in \mathrm{Stab}_U(\omega)$ is in the same $N$-coset. Thus $\mathrm{Stab}_U(\omega)$ is generated by $\mathrm{Stab}_N(\omega)$ with corrections of generators of $A$.



# Example: Complements

If $N \triangleleft G$, a ***complement*** to $N$ in $G$ is a subgroup $C$ such that $G = NU$ and $N \cap U = \langle 1 \rangle$. (Semidirect product.)

If $C$ is a complement, so is $C^g$; in general there can be multiple conjugacy classes of complements to one $N$.

If two complements are conjugate under $G$, they are conjugate under $N$.

As $C \cong G/N$, view complements as homomorphisms $G \to G$ with kernel $N$. Every $g \in G$ is mapped to $g \cdot n_g$ with the ***cofactor*** $n_g \in N$. Clearly sufficient to find these $n_g$ for generators of $G$ modulo $N$.

Necessary+sufficient condition is that the elements $g \cdot ng$ fulfill a presentation for $G/N$ in the generators $Ng$.

# Equations

If $N$ is elementary abelian then one can *collect* (→ Bettina's lectures) the equations into a $G/N$-part and an $N$-part. E.g., suppose $G/N=\langle Na,Nb\rangle$ for a,b$\in G$ and relator $Na\cdot Nb\cdot Na$. Now write this relators in $a\cdot n_a$ , $b\cdot n_b$

$$a\cdot n_a \cdot b\cdot n_b\cdot a\cdot n_a = a\cdot b\cdot n_a^b\cdot n_b\cdot a\cdot n_a = a\cdot b \cdot a\cdot n_a^{ba}\cdot n_b^a\cdot n_a$$

Thus the group element $(a\cdot b \cdot a)^{-1}\in N$ equals the (linearly written!) vector space element $n_a\cdot(M_{ba}+1)+ n_b\cdot M_a$ where $M_x$ is the matrix for the action on $N$ induced by $x$. Compute the $M_x$, the $n_y$ are variables.

These equations for all relators yield a linear inhomogeneous system of equations. (The field is that of the prime whose order divides $N$.)

# Cohomology

Each setting of values is a set of cofactors $n_x$ for the generators. Solutions correspond to complements.

Associated homogeneous system describes complements, if chosen generators for $G/N$ already describe a complement - solutions are 1-cocycles $Z^1$.

Conjugation by $m \in N$: $g \rightarrow g^m = m^{-1}gm = g\,[g,m]$ gives cofactors $n_g = [g,m]$. These generate subspace $B^1 \leq Z^1$ of 1-coboundaries, classes of complements correspond to the 1-cohomology group $H^1 = Z^1/B^1$.

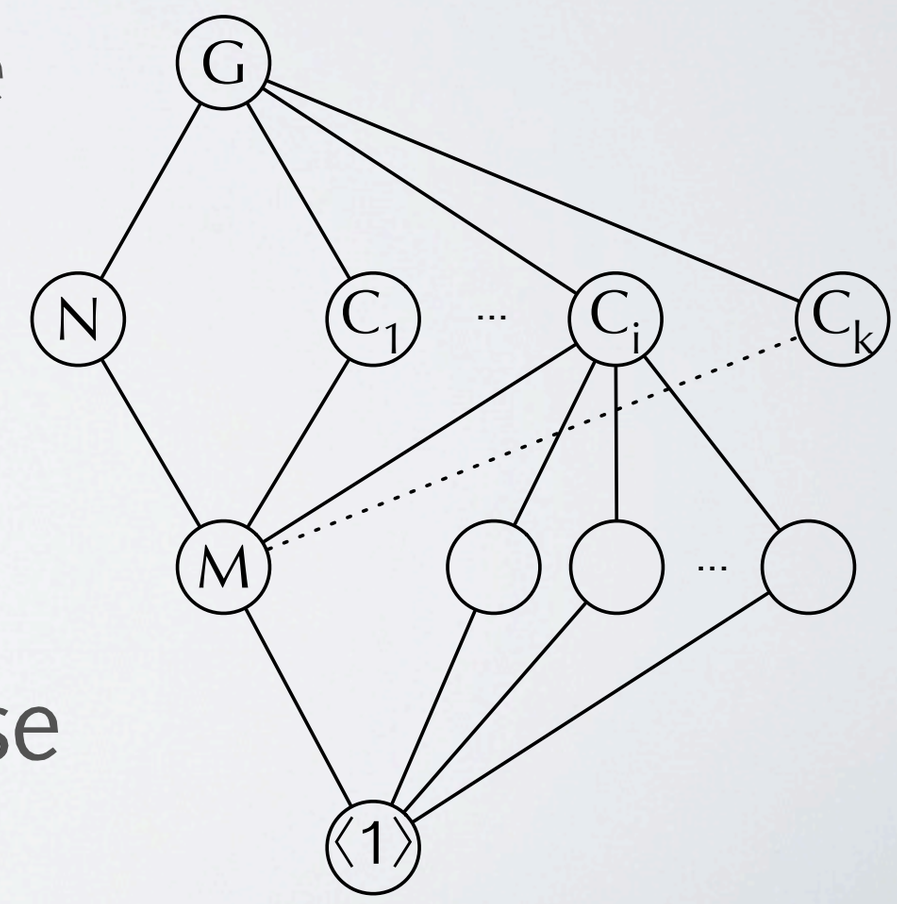
We thus get representatives of the conjugacy classes of complements from one particular solution together with representatives of $Z^1$ modulo $B^1$.

# General Case

For the general case, take a $G$-normal series for $N$ with elementary abelian factors.

In each step assume (in appropriate factor) that $N \geq M \triangleleft G$, $M$ is elementary abelian and we found representatives $C_i$ for the complements to $N/M$ in $G/M$.

Then for each $C_i$ find representatives for the complements to $M$ in $C_i$ and fuse these under action of $N_G(C_i)$.



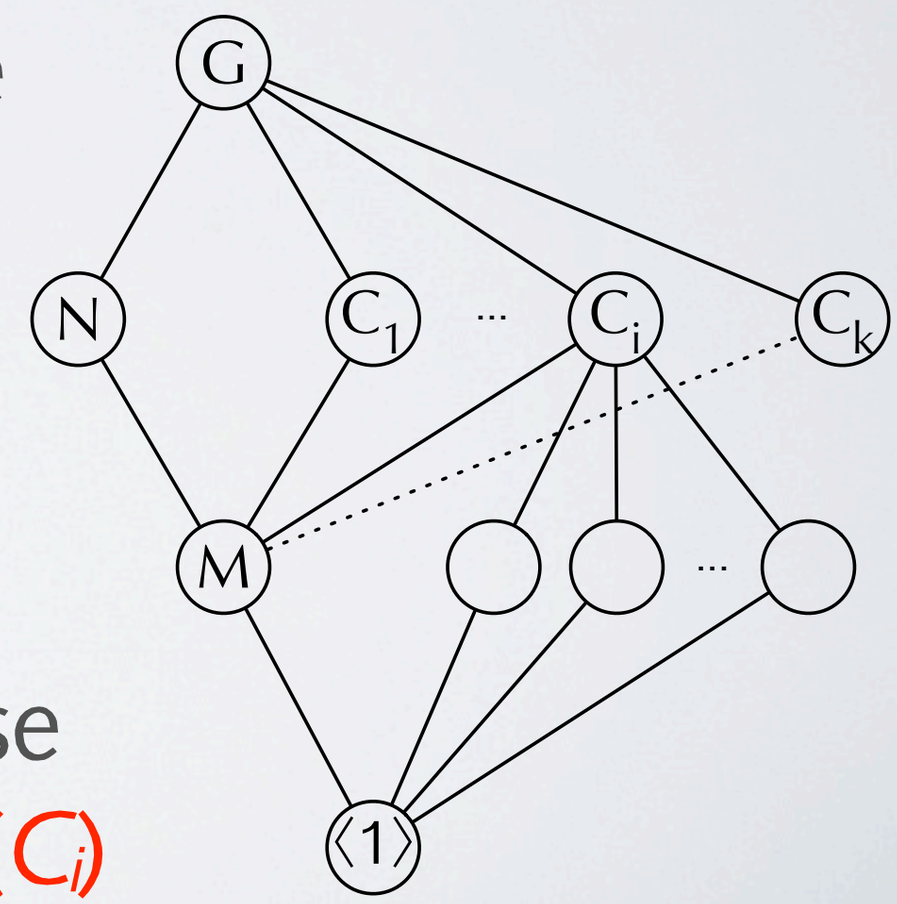
# General Case

For the general case, take a $G$-normal series for $N$ with elementary abelian factors.

In each step assume (in appropriate factor) that $N \geq M \triangleleft G$, $M$ is elementary abelian and we found representatives $C_i$ for the complements to $N/M$ in $G/M$.

Then for each $C_i$ find representatives for the complements to $M$ in $C_i$ and fuse these under action of ~~$N_G(C_i)$~~. $C_G(C_i)$

As $G/N=C$, elements of $G$ induce no outer automorphisms of $C_i$



# Subgroups

Complements are the key ingredient for determining all subgroups (up to conjugacy). Again assume $N \triangleleft G$ is elementary abelian.

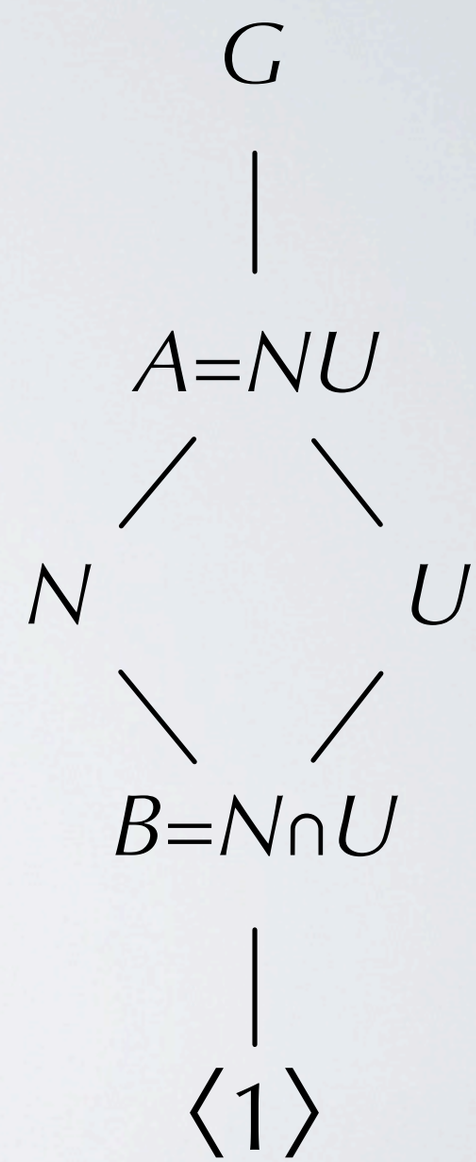
Let $U \leq G$. Then $A = \langle N, U \rangle \geq N$ and $B = N \cap U \leq N$.

Assume that we know all candidates for $A$ (as subgroups of $G/N$) and for $B$ (subspaces).

Also $B \triangleleft U$ and $B \triangleleft N$ (vector space), so $B \triangleleft A$ and $U/B$ is complement to $N/B$ in $A/B$.

Fuse under conjugation by $N_G(A) \cap N_G(B)$.

In initial step $G/Rad$, take $N = \mathrm{Soc}(G/Rad)$ as direct product, for each B find candidates for $A$ in $N_G(B)$.



# Initial Step

In the initial step, $F = G/Rad$, take as normal subgroup $N = \mathrm{Soc}(F)$.

This is a direct product of simple groups.

Subgroups of each direct factor from library (or older method, ***cyclic extension***). Combine into ***subdirect products*** to get subgroups of $N$.

For each $U \leq N$ then consider $A = \mathrm{N}_F(U)$ and determine subgroups of $A/\mathrm{N}_N(U)$.

Subgroups of $Q$ then come from complements to $\mathrm{N}_N(U)/U$ (may be non-solvable!) in the factor group $A/U$.

# If you only want some ...

Often we don't want all subgroups (or even can't hope to store all of them). In this case it might be possible to adapt the subgroup lattice computation.

Consider how the reductions to factor groups, normal subgroups relate to your desired subgroups.

GAP's functions provide some hooks for doing so, but in general one might need to adapt the code. This is easier than it may sound. **Ask**!