Permutation Groups

Max Neunhöffer

University of St Andrews

GAC 2010, Allahabad

Permutation Groups

Let Σ_n be the group of all permutations of $\{1, 2, \ldots, n\}$, the symmetric group on n points.

Permutation Groups

Let Σ_n be the group of all permutations of $\{1, 2, \ldots, n\}$, the symmetric group on n points.

Let A_n be the alternating group on n points, i.e. the even permutations.

Permutation Groups

Let Σ_n be the group of all permutations of $\{1, 2, \ldots, n\}$, the symmetric group on n points.

Let A_n be the alternating group on n points, i.e. the even permutations.

We use cycle notation:

$(1, 3, 4)(2, 5)$ maps $1 \mapsto 3 \mapsto 4 \mapsto 1$ and $2 \mapsto 5 \mapsto 2$.

Permutation Groups

Let Σ_n be the group of all permutations of $\{1, 2, \ldots, n\}$, the symmetric group on n points.

Let A_n be the alternating group on n points, i.e. the even permutations.

We use cycle notation:

$(1, 3, 4)(2, 5)$ maps $1 \mapsto 3 \mapsto 4 \mapsto 1$ and $2 \mapsto 5 \mapsto 2$.

We concatenate left before right:

$(1, 2)(2, 3) = (1, 3, 2)$.

Permutation Groups

Let Σ_n be the group of all permutations of $\{1, 2, \ldots, n\}$, the symmetric group on n points.

Let A_n be the alternating group on n points, i.e. the even permutations.

We use cycle notation:

$(1, 3, 4)(2, 5)$ maps $1 \mapsto 3 \mapsto 4 \mapsto 1$ and $2 \mapsto 5 \mapsto 2$.

We concatenate left before right:

$(1, 2)(2, 3) = (1, 3, 2)$.

Definition (Permutation group)

A permutation group on n points is a subgroup of Σ_n.

Permutation Groups

Let Σ_n be the group of all permutations of $\{1, 2, \ldots, n\}$, the symmetric group on n points.

Let A_n be the alternating group on n points, i.e. the even permutations.

We use cycle notation:

$(1, 3, 4)(2, 5)$ maps $1 \mapsto 3 \mapsto 4 \mapsto 1$ and $2 \mapsto 5 \mapsto 2$.

We concatenate left before right:

$(1, 2)(2, 3) = (1, 3, 2)$.

Definition (Permutation group)

A permutation group on n points is a subgroup of Σ_n.

Theorem

Every finite group is isomorphic to a subgroup of some symmetric group Σ_n.

GAP examples

see other window

Orbits and stabiliser cosets

Theorem (Orbit-Stabiliser)

Let G act on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$.
Let $S \setminus G := \{Sg \mid g \in G\}$.

Orbits and stabiliser cosets

Theorem (Orbit-Stabiliser)

Let G act on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$. Let $S \setminus G := \{Sg \mid g \in G\}$. Then $|G| = |xG| \cdot |S|$

Orbits and stabiliser cosets

Theorem (Orbit-Stabiliser)

Let G act on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$. Let $S \setminus G := \{Sg \mid g \in G\}$. Then $|G| = |xG| \cdot |S|$ and

$$F: \begin{array}{ccc} S \setminus G & \longrightarrow & xG \\ Sg & \longmapsto & xg \end{array}$$

is well-defined and is a bijection.

Orbits and stabiliser cosets

Theorem (Orbit-Stabiliser)

Let G act on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$. Let $S \setminus G := \{Sg \mid g \in G\}$. Then $|G| = |xG| \cdot |S|$ and

$$F : \begin{array}{ccc} S \setminus G & \longrightarrow & xG \\ Sg & \longmapsto & xg \end{array}$$

is well-defined and is a bijection.

Proof:

- If $Sg = Sg'$ then $g' = sg$ for some $s \in S$, thus $xg = xsg = xg'$. So F is well-defined.

Orbits and stabiliser cosets

Theorem (Orbit-Stabiliser)

Let G act on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$. Let $S \setminus G := \{Sg \mid g \in G\}$. Then $|G| = |xG| \cdot |S|$ and

$$F: \begin{array}{ccc} S \setminus G & \longrightarrow & xG \\ Sg & \longmapsto & xg \end{array}$$

is well-defined and is a bijection.

Proof:

- If $Sg = Sg'$ then $g' = sg$ for some $s \in S$, thus $xg = xsg = xg'$. So F is well-defined.
- F is surjective, because the image is the orbit xG.

Orbits and stabiliser cosets

Theorem (Orbit-Stabiliser)

Let G act on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$. Let $S \setminus G := \{Sg \mid g \in G\}$. Then $|G| = |xG| \cdot |S|$ and

$$F : \begin{array}{ccc} S \setminus G & \longrightarrow & xG \\ Sg & \longmapsto & xg \end{array}$$

is well-defined and is a bijection.

Proof:

- If $Sg = Sg'$ then $g' = sg$ for some $s \in S$, thus $xg = xsg = xg'$. So F is well-defined.
- F is surjective, because the image is the orbit xG.
- If $xg = xg'$, then gg'^{-1} fixes x and thus lies in S. Thus $g' = sg$ for some $s \in S$ and F is injective.

Orbits and stabiliser cosets

Theorem (Orbit-Stabiliser)

Let G act on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$. Let $S \setminus G := \{Sg \mid g \in G\}$. Then $|G| = |xG| \cdot |S|$ and

$$F : \begin{array}{ccc} S \setminus G & \longrightarrow & xG \\ Sg & \longmapsto & xg \end{array}$$

is well-defined and is a bijection.

Proof:

- If $Sg = Sg'$ then $g' = sg$ for some $s \in S$, thus $xg = xsg = xg'$. So F is well-defined.
- F is surjective, because the image is the orbit xG.
- If $xg = xg'$, then gg'^{-1} fixes x and thus lies in S. Thus $g' = sg$ for some $s \in S$ and F is injective.

Fact

We can read off in which S-coset g lies by looking at xg.

Idea

Let $G \leq \Sigma_n$, that is, G acts on $\{1, 2, \ldots, n\}$.

Idea

Let $G \leq \Sigma_n$, that is, G acts on $\{1, 2, \ldots, n\}$.

1. $i := 1, S_0 := G$

Idea

Let $G \leq \Sigma_n$, that is, G acts on $\{1, 2, \ldots, n\}$.

1. $i := 1, S_0 := G$
2. Take orbit $x_i S_{i-1}$ with $|x_i S_{i-1}| > 1$

Idea

Let $G \leq \Sigma_n$, that is, G acts on $\{1, 2, \ldots, n\}$.

1. $i := 1, S_0 := G$
2. Take orbit $x_i S_{i-1}$ with $|x_i S_{i-1}| > 1$
3. Compute $x_i S_{i-1}$ and $S_i := \mathrm{Stab}_{S_{i-1}}(x_i)$

Idea

Let $G \leq \Sigma_n$, that is, G acts on $\{1, 2, \ldots, n\}$.

1. $i := 1, S_0 := G$
2. Take orbit $x_i S_{i-1}$ with $|x_i S_{i-1}| > 1$
3. Compute $x_i S_{i-1}$ and $S_i := \mathrm{Stab}_{S_{i-1}}(x_i)$
4. This “controls” the cosets $S_i \setminus S_{i-1}$

Idea

Let $G \leq \Sigma_n$, that is, G acts on $\{1, 2, \ldots, n\}$.

1. $i := 1$, $S_0 := G$
2. Take orbit $x_i S_{i-1}$ with $|x_i S_{i-1}| > 1$
3. Compute $x_i S_{i-1}$ and $S_i := \mathrm{Stab}_{S_{i-1}}(x_i)$
4. This “controls” the cosets $S_i \setminus S_{i-1}$
5. If $S_i \neq \{1\}$ then $i := i + 1$ and go to step 2

Idea

Let $G \leq \Sigma_n$, that is, G acts on $\{1, 2, \ldots, n\}$.

1. $i := 1, S_0 := G$
2. Take orbit $x_i S_{i-1}$ with $|x_i S_{i-1}| > 1$
3. Compute $x_i S_{i-1}$ and $S_i := \mathrm{Stab}_{S_{i-1}}(x_i)$
4. This "controls" the cosets $S_i \setminus S_{i-1}$
5. If $S_i \neq \{1\}$ then $i := i + 1$ and go to step 2

$\Longrightarrow$ This computes

$$G = S_0 > S_1 > S_2 > \cdots > S_k = \{1\}$$

with the orbits $O_i := x_i S_{i-1}$ and lengths $\ell_i := |x_i S_{i-1}|$.

Idea

Let $G \leq \Sigma_n$, that is, G acts on $\{1, 2, \ldots, n\}$.

1. $i := 1, S_0 := G$
2. Take orbit $x_i S_{i-1}$ with $|x_i S_{i-1}| > 1$
3. Compute $x_i S_{i-1}$ and $S_i := \mathrm{Stab}_{S_{i-1}}(x_i)$
4. This "controls" the cosets $S_i \setminus S_{i-1}$
5. If $S_i \neq \{1\}$ then $i := i + 1$ and go to step 2

$\Longrightarrow$ This computes

$$G = S_0 > S_1 > S_2 > \cdots > S_k = \{1\}$$

with the orbits $O_i := x_i S_{i-1}$ and lengths $\ell_i := |x_i S_{i-1}|$.

All groups are given by generators.

The group order

Recall $H \setminus G := \{Hg \mid g \in G\}$ and $[G : H] := |H \setminus G|$.

The group order

Recall $H \setminus G := \{Hg \mid g \in G\}$ and $[G : H] := |H \setminus G|$.

Use inductively:

Theorem (Lagrange)

Let $H < G$ then $|G| = [G : H] \cdot |H|$.

The group order

Recall $H \setminus G := \{Hg \mid g \in G\}$ and $[G : H] := |H \setminus G|$.

Use inductively:

Theorem (Lagrange)

Let $H < G$ then $|G| = [G : H] \cdot |H|$.

Since we know all orbit lengths, we know the indices $[S_{i-1} : S_i] = |S_{i-1}|/|S_i|$.

The group order

Recall $H \setminus G := \{Hg \mid g \in G\}$ and $[G : H] := |H \setminus G|$.

Use inductively:

Theorem (Lagrange)

Let $H < G$ then $|G| = [G : H] \cdot |H|$.

Since we know all orbit lengths, we know the indices $[S_{i-1} : S_i] = |S_{i-1}|/|S_i|$.

Fact

We know the group order

$$\begin{array}{ccccccccc} |G| & = & [S_0 : S_1] & \cdot & [S_1 : S_2] & \cdots & [S_{k-1} : S_k] \\ & & \ell_1 & \cdot & \ell_2 & \cdots & \ell_k \end{array} .$$

Transversals

Even better, the Orbit-Stabiliser-Algorithm provides the Schreier tree and thus words in the generators to reach the orbit points.

Transversals

Even better, the Orbit-Stabiliser-Algorithm provides the Schreier tree and thus words in the generators to reach the orbit points.

These words form transversals: We have elements $t_j^{(i)}$ for $1 \leq i \leq k$ and $1 \leq j \leq \ell_i$ with:

$$S_{i-1} = \bigcup_{j=1}^{\ell_i} S_i t_j^{(i)} \qquad \text{(disjoint union).}$$

Transversals

Even better, the Orbit-Stabiliser-Algorithm provides the Schreier tree and thus words in the generators to reach the orbit points.

These words form transversals: We have elements $t_j^{(i)}$ for $1 \leq i \leq k$ and $1 \leq j \leq \ell_i$ with:

$$S_{i-1} = \bigcup_{j=1}^{\ell_i} S_i t_j^{(i)} \qquad \text{(disjoint union).}$$

Therefore, $g \in G$ can be written uniquely in the form

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for some numbers $j_1, j_2, \ldots, j_k$ with $1 \leq j_i \leq \ell_i$ for all i.

Transversals

Even better, the Orbit-Stabiliser-Algorithm provides the Schreier tree and thus words in the generators to reach the orbit points.

These words form transversals: We have elements $t_j^{(i)}$ for $1 \leq i \leq k$ and $1 \leq j \leq \ell_i$ with:

$$S_{i-1} = \bigcup_{j=1}^{\ell_i} S_i t_j^{(i)} \qquad \text{(disjoint union)}.$$

Therefore, $g \in G$ can be written uniquely in the form

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for some numbers $j_1, j_2, \ldots, j_k$ with $1 \leq j_i \leq \ell_i$ for all i.

Yet better, the stabiliser chain allows us to read off these numbers!

Sifting

Assume $g \in G$, thus:

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for $j_1, j_2, \ldots, j_k$ with $1 \le j_i \le \ell_i$ for $1 \le i \le k$.

Sifting

Assume $g \in G$, thus:

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for $j_1, j_2, \ldots, j_k$ with $1 \leq j_i \leq \ell_i$ for $1 \leq i \leq k$.

Since the first $k-1$ of these lie in S_1, they all fix x_1.

Sifting

Assume $g \in G$, thus:

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for $j_1, j_2, \ldots, j_k$ with $1 \leq j_i \leq \ell_i$ for $1 \leq i \leq k$.

Since the first $k-1$ of these lie in S_1, they all fix x_1.

Thus $x_1 g \in O_1$ depends only on j_1 and not on the other j_i!

Sifting

Assume $g \in G$, thus:

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for $j_1, j_2, \ldots, j_k$ with $1 \leq j_i \leq \ell_i$ for $1 \leq i \leq k$.

Since the first $k - 1$ of these lie in S_1, they all fix x_1.

Thus $x_1 g \in O_1$ depends only on j_1 and not on the other j_i!

So, we compute $x_1 g \in O_1$, look it up and determine j_1.

Sifting

Assume $g \in G$, thus:

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for $j_1, j_2, \ldots, j_k$ with $1 \leq j_i \leq \ell_i$ for $1 \leq i \leq k$.

Since the first $k-1$ of these lie in S_1, they all fix x_1.

Thus $x_1 g \in O_1$ depends only on j_1 and not on the other j_i!

So, we compute $x_1 g \in O_1$, look it up and determine j_1.

Now

$$g_1 := g {t_{j_1}^{(1)}}^{-1} = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_2}^{(2)}$$

fixes x_1 and thus lies in S_1.

Sifting

Assume $g \in G$, thus:

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for $j_1, j_2, \ldots, j_k$ with $1 \leq j_i \leq \ell_i$ for $1 \leq i \leq k$.

Since the first $k-1$ of these lie in S_1, they all fix x_1.

Thus $x_1 g \in O_1$ depends only on j_1 and not on the other j_i!

So, we compute $x_1 g \in O_1$, look it up and determine j_1.

Now

$$g_1 := g {t_{j_1}^{(1)}}^{-1} = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_2}^{(2)}$$

fixes x_1 and thus lies in S_1.

We can now compute $x_2 g_1 \in O_2$ and determine $j_2, \ldots, j_k$ inductively.

Sifting

Assume $g \in G$, thus:

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}$$

for $j_1, j_2, \ldots, j_k$ with $1 \leq j_i \leq \ell_i$ for $1 \leq i \leq k$.

Since the first $k-1$ of these lie in S_1, they all fix x_1.

Thus $x_1 g \in O_1$ depends only on j_1 and not on the other j_i!

So, we compute $x_1 g \in O_1$, look it up and determine j_1.

Now

$$g_1 := g {t_{j_1}^{(1)}}^{-1} = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_2}^{(2)}$$

fixes x_1 and thus lies in S_1.

We can now compute $x_2 g_1 \in O_2$ and determine $j_2, \ldots, j_k$ inductively.

This procedure is called sifting.

Membership test

If we sift an element $g \notin G$ (for example, if $G \leq \Sigma_n$ and $g \in \Sigma_n \setminus G$), then something will go wrong:

Membership test

If we sift an element $g \notin G$ (for example, if $G \leq \Sigma_n$ and $g \in \Sigma_n \setminus G$), then something will go wrong:

- if $x_1 g \notin O_1 = x_1 G$: then we proved that $g \notin G$,

Membership test

If we sift an element $g \notin G$ (for example, if $G \leq \Sigma_n$ and $g \in \Sigma_n \setminus G$), then something will go wrong:

- if $x_1 g \notin O_1 = x_1 G$: then we proved that $g \notin G$,
- if $x_1 g = x_1 t_j^{(1)}$ for some $1 \leq j \leq \ell_1$, then:

  since $t_j^{(1)} \in G$ and $x_1 g {t_j^{(1)}}^{-1} = x_1$, we have

  $$g \in G \qquad \iff \qquad g {t_j^{(1)}}^{-1} \in S_1.$$

  $\Longrightarrow$ Inductively, we test membership in the stabiliser.

Membership test

If we sift an element $g \notin G$ (for example, if $G \le \Sigma_n$ and $g \in \Sigma_n \setminus G$), then something will go wrong:

- if $x_1 g \notin O_1 = x_1 G$: then we proved that $g \notin G$,
- if $x_1 g = x_1 t_j^{(1)}$ for some $1 \le j \le \ell_1$, then:

  since $t_j^{(1)} \in G$ and $x_1 g {t_j^{(1)}}^{-1} = x_1$, we have
  $$g \in G \qquad \Longleftrightarrow \qquad g {t_j^{(1)}}^{-1} \in S_1.$$

  $\Longrightarrow$ Inductively, we test membership in the stabiliser.

Theorem (Stabiliser chain and sifting)

Given $G \le \Sigma_n$, an element $g \in \Sigma_n$ and a stabiliser chain for G with its orbits and Schreier trees, we can sift it, and

Membership test

If we sift an element $g \notin G$ (for example, if $G \leq \Sigma_n$ and $g \in \Sigma_n \setminus G$), then something will go wrong:

- if $x_1 g \notin O_1 = x_1 G$: then we proved that $g \notin G$,
- if $x_1 g = x_1 t_j^{(1)}$ for some $1 \leq j \leq \ell_1$, then:

since $t_j^{(1)} \in G$ and $x_1 g {t_j^{(1)}}^{-1} = x_1$, we have

$$g \in G \quad \Longleftrightarrow \quad g {t_j^{(1)}}^{-1} \in S_1.$$

$\Longrightarrow$ Inductively, we test membership in the stabiliser.

Theorem (Stabiliser chain and sifting)

Given $G \leq \Sigma_n$, an element $g \in \Sigma_n$ and a stabiliser chain for G with its orbits and Schreier trees, we can sift it, and

- *either prove that $g \notin G$,*

Membership test

If we sift an element $g \notin G$ (for example, if $G \leq \Sigma_n$ and $g \in \Sigma_n \setminus G$), then something will go wrong:

- if $x_1 g \notin O_1 = x_1 G$: then we proved that $g \notin G$,
- if $x_1 g = x_1 t_j^{(1)}$ for some $1 \leq j \leq \ell_1$, then:

  since $t_j^{(1)} \in G$ and $x_1 g {t_j^{(1)}}^{-1} = x_1$, we have

$$g \in G \qquad \Longleftrightarrow \qquad g {t_j^{(1)}}^{-1} \in S_1.$$

$\Longrightarrow$ Inductively, we test membership in the stabiliser.

Theorem (Stabiliser chain and sifting)

Given $G \leq \Sigma_n$, an element $g \in \Sigma_n$ and a stabiliser chain for G with its orbits and Schreier trees, we can sift it, and

- *either prove that $g \notin G$,*
- *or write g constructively in a unique way as product of transversal elements*

$$g = t_{j_k}^{(k)} \cdot t_{j_{k-1}}^{(k-1)} \cdots t_{j_1}^{(1)}.$$

Base and strong generators

Let $G \leq \Sigma_n$ and $G = S_0 > S_1 > \cdots > S_k = \{1\}$ be a stabiliser chain. Let S be the set of all generators of all the S_i for $0 \leq i < k$.

Base and strong generators

Let $G \leq \Sigma_n$ and $G = S_0 > S_1 > \cdots > S_k = \{1\}$ be a stabiliser chain. Let S be the set of all generators of all the S_i for $0 \leq i < k$.

Definition (Base and strong generators)

The points $x_1, x_2, \ldots, x_k$ together are called a base for G, since the only element $g \in G$ fixing them all is the identity.

Base and strong generators

Let $G \leq \Sigma_n$ and $G = S_0 > S_1 > \cdots > S_k = \{1\}$ be a stabiliser chain. Let S be the set of all generators of all the S_i for $0 \leq i < k$.

Definition (Base and strong generators)

The points $x_1, x_2, \ldots, x_k$ together are called a base for G, since the only element $g \in G$ fixing them all is the identity. The set S is called a set of strong generators for G, since $\langle S \cap S_i \rangle = S_i$ for $0 \leq i \leq k$.

Base and strong generators

Let $G \leq \Sigma_n$ and $G = S_0 > S_1 > \cdots > S_k = \{1\}$ be a stabiliser chain. Let S be the set of all generators of all the S_i for $0 \leq i < k$.

Definition (Base and strong generators)

The points $x_1, x_2, \ldots, x_k$ together are called a base for G, since the only element $g \in G$ fixing them all is the identity.
The set S is called a set of strong generators for G, since $\langle S \cap S_i \rangle = S_i$ for $0 \leq i \leq k$.
A stabiliser chain provides a base and a strong generating set.

Base and strong generators

Let $G \leq \Sigma_n$ and $G = S_0 > S_1 > \cdots > S_k = \{1\}$ be a stabiliser chain. Let S be the set of all generators of all the S_i for $0 \leq i < k$.

Definition (Base and strong generators)

The points $x_1, x_2, \ldots, x_k$ together are called a base for G, since the only element $g \in G$ fixing them all is the identity. The set S is called a set of strong generators for G, since $\langle S \cap S_i \rangle = S_i$ for $0 \leq i \leq k$.

A stabiliser chain provides a base and a strong generating set.

Fact

The sifting procedure expresses a $g \in G$ as a product of the strong generators S.

How to compute a stabiliser chain

Theorem (Schreier-Sims)

Let $G = \langle T \rangle \leq \Sigma_n$. A base and strong generating set for G can be computed in time bounded by

$$C \cdot (n^2 \log^3 |G| + |T| n^2 \log |G|)$$

for some constant $C > 0$.

How to compute a stabiliser chain

Theorem (Schreier-Sims)

Let $G = \langle T \rangle \leq \Sigma_n$. A base and strong generating set for G can be computed in time bounded by

$$C \cdot (n^2 \log^3 |G| + |T| n^2 \log |G|)$$

for some constant $C > 0$.

Theorem (Babai, Cooperman, Finkelstein, Seress)

Let $G = \langle T \rangle \leq \Sigma_n$ and d an arbitrary constant. A guess for a strong generating set for G can be computed in time bounded by

$$C \cdot (n \log n \log^4 |G| + |T| n \log |G|)$$

for some constant $C > 0$ with error probability $\leq 1/n^d$.

Nearly linear time

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed in time less than $C \cdot n \cdot |T| \cdot \log^c |G|$ for some constants C and c:

Nearly linear time

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed in time less than $C \cdot n \cdot |T| \cdot \log^c |G|$ for some constants C and c:

- a base and strong generating set,

Nearly linear time

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed in time less than $C \cdot n \cdot |T| \cdot \log^c |G|$ for some constants C and c:

- a base and strong generating set,
- images under homomorphisms,

Nearly linear time

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed in time less than $C \cdot n \cdot |T| \cdot \log^c |G|$ for some constants C and c:

- a base and strong generating set,
- images under homomorphisms,
- pointwise stabilisers,

Nearly linear time

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed in time less than $C \cdot n \cdot |T| \cdot \log^c |G|$ for some constants C and c:

- a base and strong generating set,
- images under homomorphisms,
- pointwise stabilisers,
- closure and normal closure,

Nearly linear time

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed in time less than $C \cdot n \cdot |T| \cdot \log^c |G|$ for some constants C and c:

- a base and strong generating set,
- images under homomorphisms,
- pointwise stabilisers,
- closure and normal closure,
- a composition series,

Nearly linear time

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed in time less than $C \cdot n \cdot |T| \cdot \log^c |G|$ for some constants C and c:

- a base and strong generating set,
- images under homomorphisms,
- pointwise stabilisers,
- closure and normal closure,
- a composition series,
- the center $C(G)$ of G.

Slower algorithms

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed:

Slower algorithms

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed:

- centraliser in Σ_n (still polynomial time),

Slower algorithms

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed:

- centraliser in Σ_n (still polynomial time),
- centraliser $C_G(g)$ of an element $g \in G$,

Slower algorithms

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed:

- centraliser in Σ_n (still polynomial time),
- centraliser $C_G(g)$ of an element $g \in G$,
- setwise stabiliser in G of a set $M \subseteq \{1, 2, \ldots, n\}$,

Slower algorithms

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed:

- centraliser in Σ_n (still polynomial time),
- centraliser $C_G(g)$ of an element $g \in G$,
- setwise stabiliser in G of a set $M \subseteq \{1, 2, \ldots, n\}$,
- the intersection of two such groups,

Slower algorithms

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed:

- centraliser in Σ_n (still polynomial time),
- centraliser $C_G(g)$ of an element $g \in G$,
- setwise stabiliser in G of a set $M \subseteq \{1, 2, \ldots, n\}$,
- the intersection of two such groups,
- a conjugating element $g \in G$ with $a^g = b$ for $a, b \in G$,

Slower algorithms

For $\langle T \rangle = G \leq \Sigma_n$ the following can be computed:

- centraliser in Σ_n (still polynomial time),
- centraliser $C_G(g)$ of an element $g \in G$,
- setwise stabiliser in G of a set $M \subseteq \{1, 2, \ldots, n\}$,
- the intersection of two such groups,
- a conjugating element $g \in G$ with $a^g = b$ for $a, b \in G$,
- the normaliser in Σ_n.

Large base

One problem with stabiliser chains is Σ_n itself:

Fact

The smallest base for Σ_n itself contains $n-1$ points. Not surprising, since $\log|G| = \log(n!) \approx n \log n - n$.

Large base

One problem with stabiliser chains is Σ_n itself:

Fact

The smallest base for Σ_n itself contains $n - 1$ points. Not surprising, since $\log|G| = \log(n!) \approx n \log n - n$.

Thus, time $n \log|T| \log^c |G|$ becomes $n^{(c+1)} \log^c n \log|T|$.

Large base

One problem with stabiliser chains is Σ_n itself:

Fact

The smallest base for Σ_n itself contains $n-1$ points. Not surprising, since $\log|G| = \log(n!) \approx n \log n - n$.

Thus, time $n \log |T| \log^c |G|$ becomes $n^{(c+1)} \log^c n \log |T|$.

Definition (Small-base family)

A family $\mathcal{F}$ of permutation groups is called a family of small-base groups, if there is a constant c such that each $G \in \mathcal{F}$ of degree n satisfies $\log |G| \leq \log^c n$.

Large base

One problem with stabiliser chains is Σ_n itself:

Fact

The smallest base for Σ_n itself contains $n-1$ points.
Not surprising, since $\log|G| = \log(n!) \approx n\log n - n$.

Thus, time $n \log|T| \log^c |G|$ becomes $n^{(c+1)} \log^c n \log|T|$.

Definition (Small-base family)

A family $\mathcal{F}$ of permutation groups is called a family of small-base groups, if there is a constant c such that each $G \in \mathcal{F}$ of degree n satisfies $\log|G| \leq \log^c n$.

Example

All permutation representations of all finite simple groups except the alternating groups form a family of small-base groups.

The End

Bibliography

G. Butler.
Fundamental algorithms for permutation groups, volume 559 of *Lecture Notes in Computer Science*.
Springer-Verlag, Berlin, 1991.

Ákos Seress.
Permutation group algorithms, volume 152 of *Cambridge Tracts in Mathematics*.
Cambridge University Press, Cambridge, 2003.