Wreath Product Decompositions

Friedrich Rober

Algebra and Representation Theory

RWTH AACHEN UNIVERSITY

Computational Group Theory

Groups encode symmetry

What is a group?

Groups encode symmetry

symmetries of square

rotations of tetrahedron

shuffles of cards

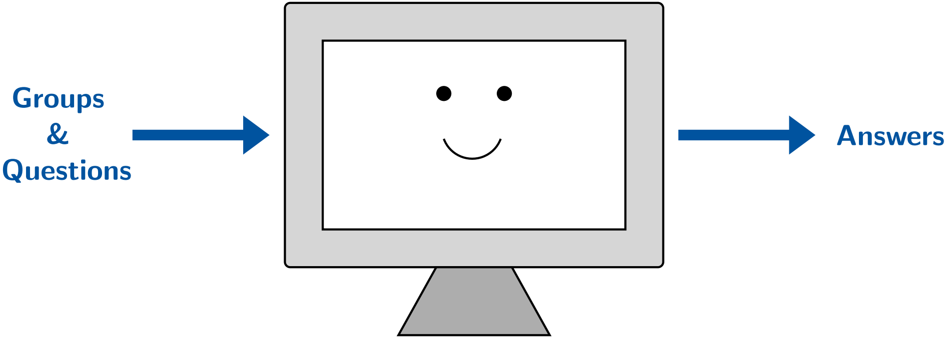
Groups encode symmetry

$\mathrm{Dih}(2 \cdot 4)$

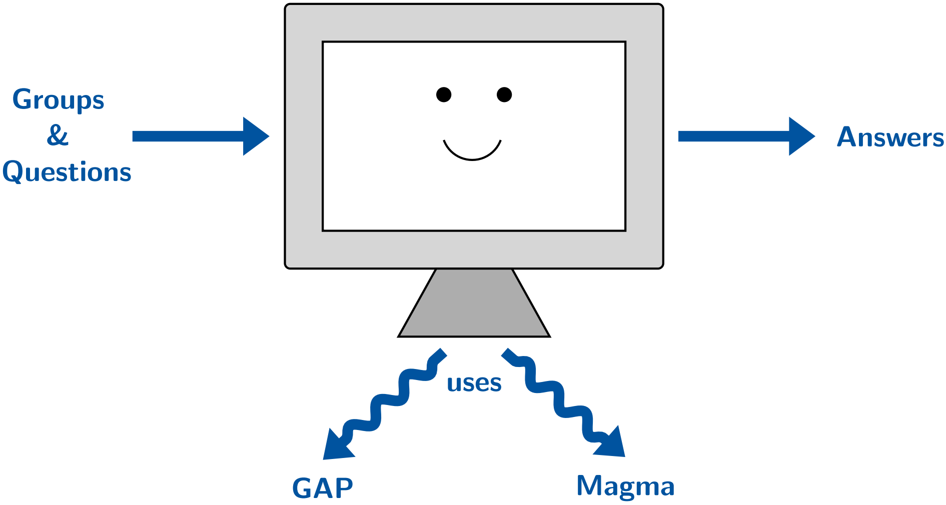
$\mathrm{Alt}(4)$

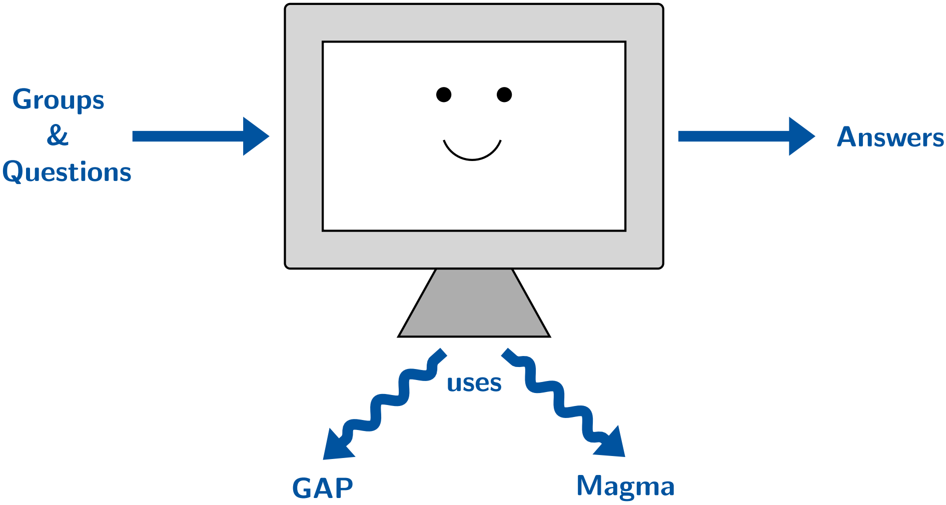
$\mathrm{Sym}(4)$

The computer is our friend!



The computer is our friend!





How does it work?

There are numerous ways to represent a (finite) group.

Our setting

$$\underbrace{P = \mathrm{Sym}(n)}_{\text{permutation group}} \quad \text{or} \quad \underbrace{P = \mathrm{GL}(d, q)}_{\text{matrix group}}$$

There are numerous ways to represent a (finite) group.

Our setting

$$\underbrace{P = \mathrm{Sym}(n)}_{\text{permutation group}} \quad \text{or} \quad \underbrace{P = \mathrm{GL}(d,q)}_{\text{matrix group}}$$

Represent a group G via a **generating set** $X \subseteq P$ as

$$G = \langle X \rangle := \{\underbrace{x_1 \cdots x_\ell}_{\text{word}} : x_i \in X \cup X^{-1}, \ell \in \mathbb{N}\}.$$

There are numerous ways to represent a (finite) group.

Our setting

$$\underbrace{P = \mathrm{Sym}(n)}_{\text{permutation group}} \quad \text{or} \quad \underbrace{P = \mathrm{GL}(d,q)}_{\text{matrix group}}$$

Represent a group G via a **generating set** $X \subseteq P$ as

$$G = \langle X \rangle := \{\underbrace{x_1 \cdots x_\ell}_{\text{word}} : x_i \in X \cup X^{-1}, \ell \in \mathbb{N}\}.$$

For example, $\mathrm{Sym}(n) = \langle\, (1,2), (1,2,\ldots,n) \,\rangle \rightsquigarrow \mathcal{O}(n \log(n))$ bits.
However, $|\mathrm{Sym}(n)| = n!$ is comparably very large.

There are various questions/problems posed on groups.

Fundamental problems

- **Group order:** Determine $|G|$.
- **Membership:** Given $x \in P$, is x an element of G?
- **Rewriting:** Given $g \in G$, write g as a word in X.

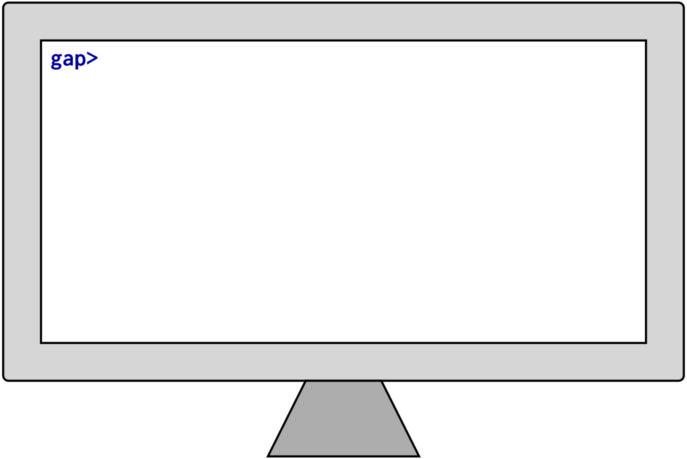
There are various questions/problems posed on groups.

Fundamental problems

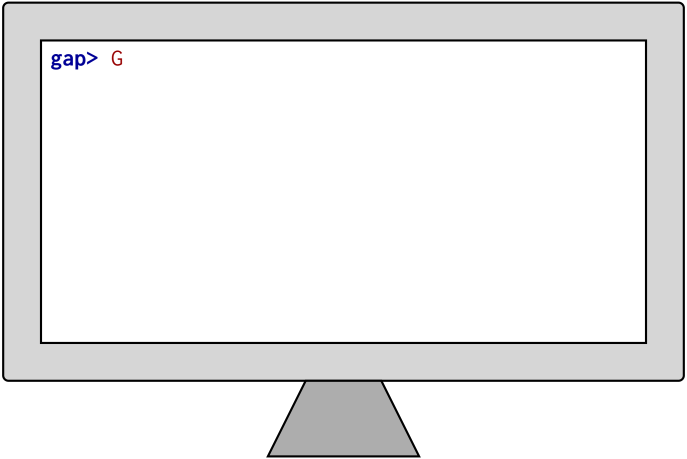
- **Group order:** Determine $|G|$.
- **Membership:** Given $x \in P$, is x an element of G?
- **Rewriting:** Given $g \in G$, write g as a word in X.

Solving these problems efficiently is a non-trivial task!

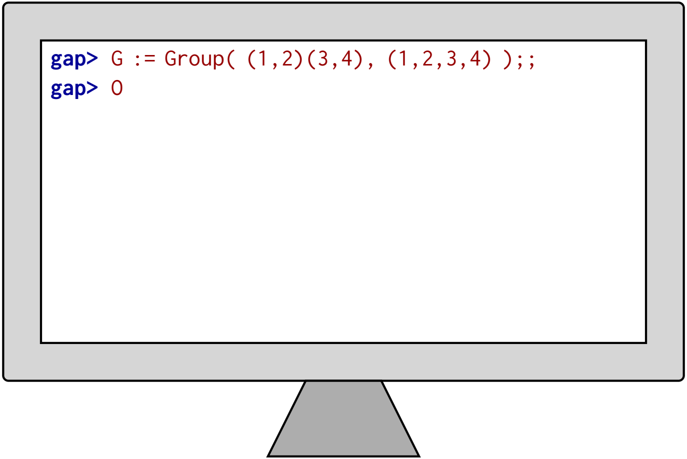
Talking to the computer



Talking to the computer



Talking to the computer



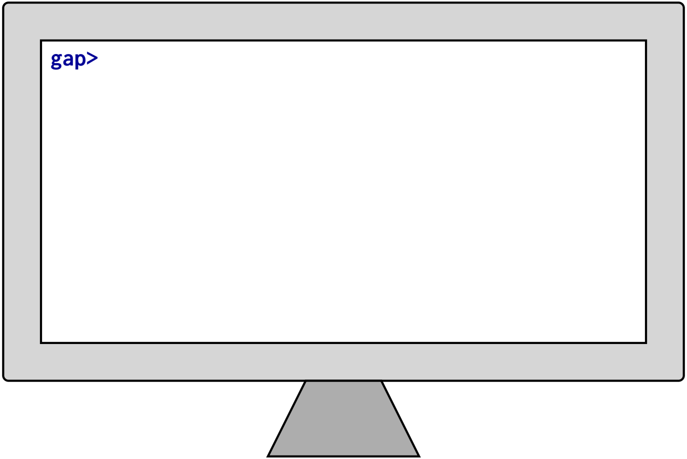
Talking to the computer

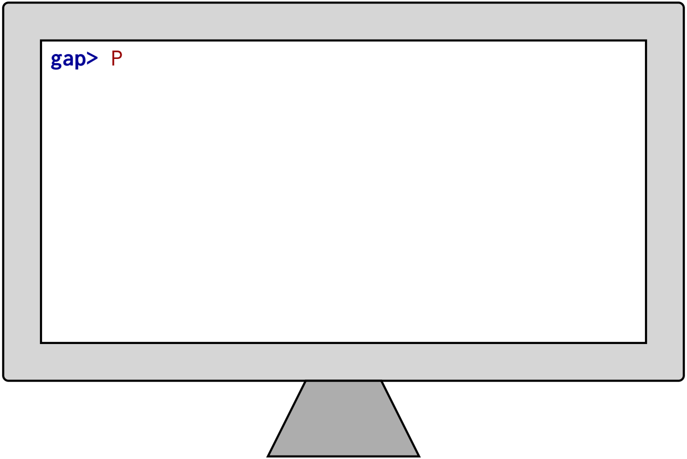
```
gap> G := Group( (1,2)(3,4), (1,2,3,4) );;
gap> Order(G);
8
gap> (
```

Talking to the computer

```
gap> G := Group( (1,2)(3,4), (1,2,3,4) );;
gap> Order(G);
8
gap> (1,2,3) in G;
false
gap> (
```

Talking to the computer




gap> P

```
gap> P := SymmetricGroup(100);;
gap> G
```

Talking to the computer

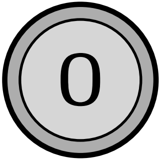
```
gap> P := SymmetricGroup(100);;
gap> G := Group(Random(P), Random(P));
<permutation group with 2 generators>
gap> O
```

Talking to the computer

```
gap> P := SymmetricGroup(100);;
gap> G := Group(Random(P), Random(P));
<permutation group with 2 generators>
gap> Order(G);
933262154439441526816992388562667004907159682643\
816214685929638952175999932299156089414639761565\
182862536979208272237582511852109168640000000000\
00000000000000
gap> t
```

Randomisation is key

Random Elements



Random Elements

Random Elements

Bits:

Input: $G = \langle X \rangle$ and $\ell \in \mathbb{N}$

Output: ℓ independent (nearly-)uniformly distributed elements from G

Randomisation is key

Random Elements

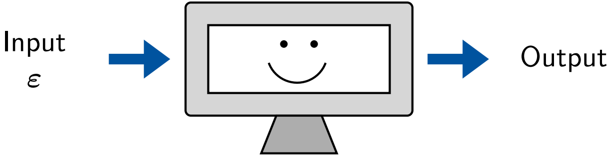
Bits: 1 0 0 1 0 1 1 1 ...

Input: $G = \langle X \rangle$ and $\ell \in \mathbb{N}$

Output: ℓ independent (nearly-)uniformly distributed elements from G

Monte-Carlo algorithms

Let $\varepsilon \in (0,1)$ be an error probability



$\mathrm{Prob}(\text{Output is incorrect}) \leq \varepsilon$

Runtime is $\mathcal{O}\big(f(\text{Input}) \cdot \log(1/\varepsilon)\big)$

Black box model

A computational model that covers all finite group representations

$$G = \langle X \rangle$$

$$\underbrace{\begin{matrix} 001000010 \\ 110001101 \\ 000100010 \\ \vdots \end{matrix}}_{\text{elements of } G}$$

A computational model that covers all finite group representations

allowed

x^{-1}

$x * y$

$x = y$

$G = \langle X \rangle$

$$\underbrace{\begin{matrix} 001000010 \\ 110001101 \\ 000100010 \\ \vdots \end{matrix}}_{\text{elements of } G}$$

A computational model that covers all finite group representations

allowed

x^{-1}

$x * y$

$x = y$

$G = \langle X \rangle$

not allowed

action on point

entry of a matrix

anything else...

$$\underbrace{\begin{matrix}001000010\\110001101\\000100010\\\vdots\end{matrix}}_{\text{elements of } G}$$

Input: $G = \langle X \rangle \leq \mathrm{Sym}(n)$

Idea: compression $\rightsquigarrow$ base $B \subseteq \{1, \ldots, n\}$
$\rightsquigarrow$ stabiliser chain & strong generators

Input: $G = \langle X \rangle \leq \mathrm{Sym}(n)$

Idea: compression $\leadsto$ base $B \subseteq \{1, \ldots, n\}$

$\leadsto$ stabiliser chain & strong generators

$$G = G^{[1]} \geqslant G^{[2]} \geqslant \cdots \geqslant G^{[\beta]} \geqslant G^{[\beta+1]} = \{1_G\}\,,$$

Input: $G = \langle X \rangle \leq \mathrm{Sym}(n)$

Idea: compression $\rightsquigarrow$ base $B \subseteq \{1, \ldots, n\}$
$\rightsquigarrow$ stabiliser chain & strong generators

$$G = G^{[1]} \geqslant G^{[2]} \geqslant \cdots \geqslant G^{[\beta]} \geqslant G^{[\beta+1]} = \{1_G\},$$

If G is **small-base**, then many algorithms are nearly-linear:

$$\mathcal{O}\left(n \log(n)^c \right)$$

If G is **large-base**, then many algorithms are polynomial:

$$\mathcal{O}\left(n^c \right)$$

Input: $G = \langle X \rangle \leq \mathrm{Sym}(n)$

Idea: compression $\rightsquigarrow$ base $B \subseteq \{1, \ldots$
$\rightsquigarrow$ stabiliser chain & strong ge

Wreath Products

$$G = G^{[1]} \geqslant G^{[2]} \geqslant \cdots \geqslant G^{[\beta]} \geqslant G^{[}$$

If G is **small-base**,
then many algorithms are
nearly-linear:

$$\mathcal{O}\left(\; n \log(n)^c \; \right)$$

If G is **large-base**,
then many algorithms are
polynomial:

$$\mathcal{O}\left(\; n^c \; \right)$$

Input: $G = \langle X \rangle \leq \mathrm{GL}(d, q)$

Idea: divide & conquer $\rightsquigarrow$ recognition tree

Input: $G = \langle X \rangle \leq \mathrm{GL}(d, q)$

Idea: divide & conquer $\rightsquigarrow$ recognition tree

$$
\begin{tikzcd}
 & \underset{\varphi_1}{G} \arrow[dr, two heads] & & \\
N_1 \arrow[ur, hook] & & \underset{\varphi_2}{H_1} \arrow[dr, two heads] & \\
 & N_2 \arrow[ur, hook] & & H_2
\end{tikzcd}
$$

Input: $G = \langle X \rangle \leq \mathrm{GL}(d, q)$

Idea: divide & conquer $\rightsquigarrow$ recognition tree

$$
\begin{tikzcd}
 & \underset{\varphi_1}{G} \arrow[dr, two heads] & & \\
N_1 \arrow[ur, hook] & & \underset{\varphi_2}{H_1} \arrow[dr, two heads] & \\
 & N_2 \arrow[ur, hook] & & H_2
\end{tikzcd}
$$

Leaf Nodes: (quasi-)simple groups

Computing in large matrix groups

Input: $G = \langle X \rangle \leq \mathrm{GL}(d, q)$

Idea: divide & conquer $\rightsquigarrow$ recognition tree

$$\begin{tikzcd}
 & \underset{\varphi_1}{G} \arrow[dr, two heads] & & \\
N_1 \arrow[ur, hook] & & \underset{\varphi_2}{H_1} \arrow[dr, two heads] & \\
 & N_2 \arrow[ur, hook] & & H_2
\end{tikzcd}$$

Leaf Nodes: (quasi-)simple groups $\rightsquigarrow$ **constructive recognition**

Theorem (Classification of finite simple groups (CFSG))

Let G be a finite simple group. Then G is one of the following:

- a cyclic group of prime order;
- a finite alternating group of degree at least 5;
- a finite group of Lie type;
- one of 26 sporadic groups.

Theorem (Classification of finite simple groups (CFSG))

Let G be a finite simple group. Then G is one of the following:

- a cyclic group of prime order;
- a finite alternating group of degree at least 5;
- a finite group of Lie type;
- one of 26 sporadic groups.

Theorem (Classification of finite simple groups (CFSG))

Let G be a finite simple group. Then G is one of the following:

- a cyclic group of prime order;
- a finite alternating group of degree at least 5;
- a finite group of Lie type;
- one of 26 sporadic groups.

Input: $G = \langle X \rangle$, a (quasi-)simple group

Idea: identify and translate to nice representation

Theorem (Classification of finite simple groups (CFSG))

Let G be a finite simple group. Then G is one of the following:

- a cyclic group of prime order;
- a finite alternating group of degree at least 5;
- a finite group of Lie type;
- one of 26 sporadic groups.

Input: $G = \langle X \rangle$, a (quasi-)simple group

Idea: identify and translate to nice representation

Output: $\varphi : G \to \overset{♛}{G}$ isomorphism to golden copy of G

	Matrix Groups	Permutation Groups
Input:	$G \leq \mathrm{GL}(d, q)$	$G \leq \mathrm{Sym}(n)$

How about large permutation groups?

	Matrix Groups	Permutation Groups
Input:	$G \leq \mathrm{GL}(d, q)$	$G \leq \mathrm{Sym}(n)$
Idea:	recognition tree	

$$
\begin{tikzcd}
 & G \arrow[dr, two heads, "\varphi_1"] & & \\
N_1 \arrow[ur, hook] & & H_1 \arrow[dr, two heads, "\varphi_2"] & \\
 & N_2 \arrow[ur, hook] & & H_2
\end{tikzcd}
$$

How about large permutation groups?

	Matrix Groups	Permutation Groups
Input:	$G \leq \mathrm{GL}(d, q)$	$G \leq \mathrm{Sym}(n)$
Idea:	recognition tree	

$$
\begin{tikzcd}
 & G \arrow[dr, two heads, "\varphi_1"] & & \\
N_1 \arrow[ur, hook] & & H_1 \arrow[dr, two heads, "\varphi_2"] & \\
 & N_2 \arrow[ur, hook] & & H_2
\end{tikzcd}
$$

Leaves:	(quasi-)simple groups	**primitive groups**

How about large permutation groups?

	Matrix Groups	Permutation Groups
Input:	$G \leq \mathrm{GL}(d, q)$	$G \leq \mathrm{Sym}(n)$
Idea:	recognition tree	

$$\begin{tikzcd}
 & G \arrow[dr, two heads, "\varphi_1"] & & \\
N_1 \arrow[ur, hook] & & H_1 \arrow[dr, two heads, "\varphi_2"] & \\
 & N_2 \arrow[ur, hook] & & H_2
\end{tikzcd}$$

Leaves:	(quasi-)simple groups	**primitive groups**
Structure:	CFSG	O'Nan–Scott classification

How about large permutation groups?

	Matrix Groups	Permutation Groups
Input:	$G \leq \mathrm{GL}(d, q)$	$G \leq \mathrm{Sym}(n)$

Idea: recognition tree

$$\begin{tikzcd}
 & G \arrow[dr, two heads, "\varphi_1"] & & \\
N_1 \arrow[ur, hook] & & H_1 \arrow[dr, two heads, "\varphi_2"] & \\
 & N_2 \arrow[ur, hook] & & H_2
\end{tikzcd}$$

Wreath Products

	Matrix Groups	Permutation Groups
Leaves:	(quasi-)simple groups	**primitive groups**
Structure:	CFSG	O'Nan–Scott classification

Wreath Products

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

$$\text{Let } w := \big(\overbrace{\underset{w_1}{\boxed{1}},\ \underset{w_2}{\boxed{2}},\ \underset{w_3}{\boxed{3}},\ \ldots,\ \underset{w_m}{\boxed{m}}}^{\text{base}}\ ;\ \overset{\text{top}}{\pi} \big) \in W$$

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

$$\text{Let } w := \Big(\overbrace{\underset{\boxed{1}}{} w_1, \; \underset{\boxed{2}}{} w_2, \; \underset{\boxed{3}}{} w_3, \; \ldots, \underset{\boxed{m}}{} w_m}^{\text{base}} \; ; \; \overset{\text{top}}{\pi} \Big) \in W$$

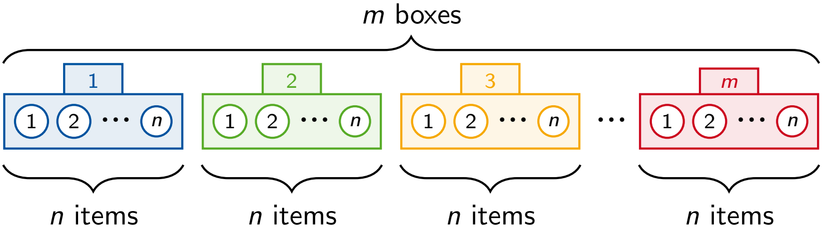
The **imprimitive action** is on $(n \cdot m)$ points:

Wreath product - imprimitive action

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

$$\text{Let } w := \Big(\overbrace{\overset{\boxed{1}}{w_1},\ \overset{\boxed{2}}{w_2},\ \overset{\boxed{3}}{w_3},\ \ldots,\ \overset{\boxed{m}}{w_m}}^{\text{base}}\ ;\ \overset{\text{top}}{\pi}\Big) \in W$$

The **imprimitive action** is on $(n \cdot m)$ points:

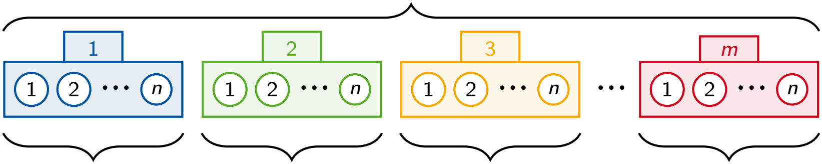


Wreath product - imprimitive action

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

$$\text{Let } w := \Big(\overbrace{\overset{\boxed{1}}{w_1},\ \overset{\boxed{2}}{w_2},\ \overset{\boxed{3}}{w_3},\ \dots,\ \overset{\boxed{m}}{w_m}}^{\text{base}}\ ;\ \overset{\text{top}}{\pi} \Big) \in W$$

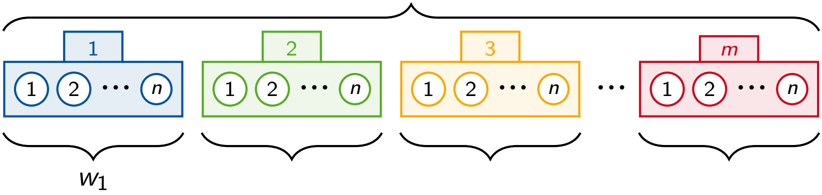
The **imprimitive action** is on $(n \cdot m)$ points:



Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

$$\text{Let } w := \Big(\overbrace{\overset{\boxed{1}}{w_1}, \ \overset{\boxed{2}}{w_2}, \ \overset{\boxed{3}}{w_3}, \ \ldots, \overset{\boxed{m}}{w_m}}^{\text{base}} ; \ \overset{\text{top}}{\pi} \Big) \in W$$

The **imprimitive action** is on $(n \cdot m)$ points:

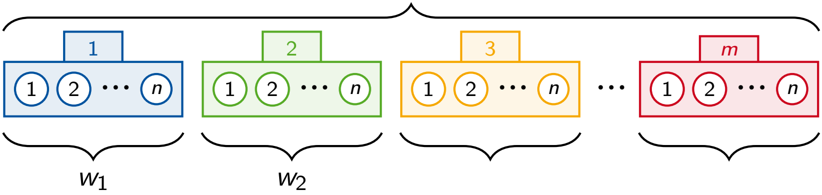


Wreath product - imprimitive action

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

$$\text{Let } w := \Big(\overbrace{\overset{\boxed{1}}{w_1},\ \overset{\boxed{2}}{w_2},\ \overset{\boxed{3}}{w_3},\ \ldots,\ \overset{\boxed{m}}{w_m}}^{\text{base}}\,;\ \overset{\text{top}}{\pi}\Big) \in W$$

The **imprimitive action** is on $(n \cdot m)$ points:

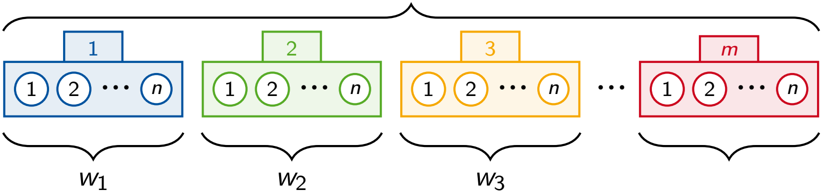


Wreath product - imprimitive action

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

$$\text{Let } w := \Big(\overbrace{\overset{\boxed{1}}{w_1},\ \overset{\boxed{2}}{w_2},\ \overset{\boxed{3}}{w_3},\ \dots,\ \overset{\boxed{m}}{w_m}}^{\text{base}};\ \overset{\text{top}}{\pi}\Big) \in W$$

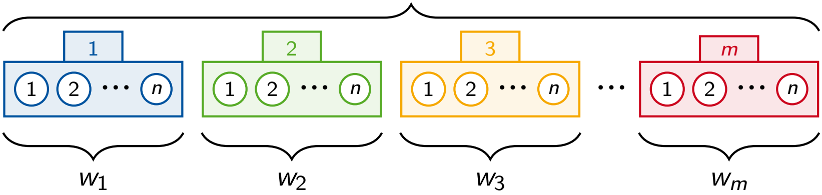
The **imprimitive action** is on $(n \cdot m)$ points:



Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

$$\text{Let } w := \Big(\overbrace{\overset{\boxed{1}}{w_1},\ \overset{\boxed{2}}{w_2},\ \overset{\boxed{3}}{w_3},\ \ldots,\ \overset{\boxed{m}}{w_m}}^{\text{base}};\ \overset{\text{top}}{\pi}\Big) \in W$$

The **imprimitive action** is on $(n \cdot m)$ points:

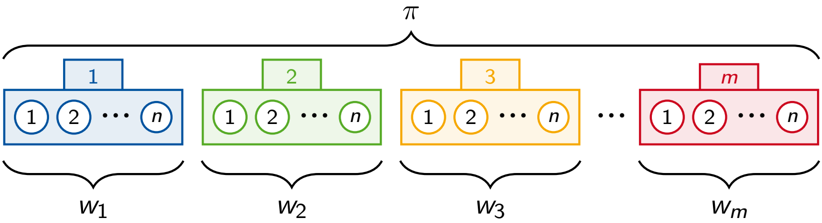


Wreath product - imprimitive action

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H := K^m \rtimes H$ be the wreath product.

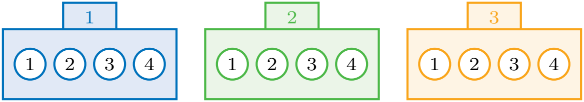
$$\text{Let } w := \Big(\overbrace{\overset{\boxed{1}}{w_1},\ \overset{\boxed{2}}{w_2},\ \overset{\boxed{3}}{w_3},\ \ldots,\ \overset{\boxed{m}}{w_m}}^{\text{base}};\ \overset{\text{top}}{\pi}\Big) \in W$$

The **imprimitive action** is on $(n \cdot m)$ points:



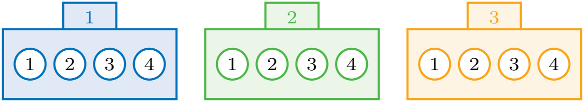
Wreath product - imprimitive action

$$w := \Big(\overset{\boxed{1}}{(1,4)},\ \overset{\boxed{2}}{(1,2)(3,4)},\ \overset{\boxed{3}}{(1,2,3)}\ ;\ \overset{\text{top}}{(2,3)} \Big) \in \mathrm{Sym}(4) \wr \mathrm{Sym}(3)$$

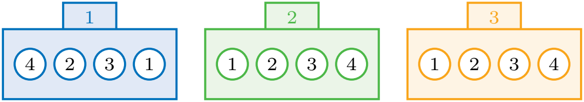


Wreath product - imprimitive action

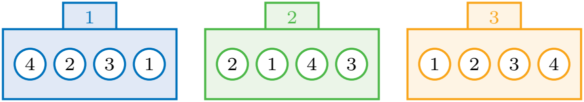
$$w := \Big(\overset{\boxed{1}}{\underline{(1,4)}},\ \ \overset{\boxed{2}}{(1,2)(3,4)},\ \ \overset{\boxed{3}}{(1,2,3)}\ ;\ \ \overset{\text{top}}{(2,3)} \Big) \in \mathrm{Sym}(4) \wr \mathrm{Sym}(3)$$



$$w := \left(\overset{\boxed{1}}{(1,4)},\ \underline{\overset{\boxed{2}}{(1,2)(3,4)}},\ \overset{\boxed{3}}{(1,2,3)}\ ;\ \overset{\text{top}}{(2,3)} \right) \in \mathrm{Sym}(4) \wr \mathrm{Sym}(3)$$

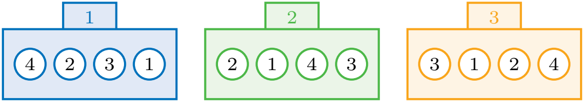


$$w := \Big(\, \overset{\boxed{1}}{(1,4)},\ \overset{\boxed{2}}{(1,2)(3,4)},\ \overset{\boxed{3}}{\underline{(1,2,3)}}\,;\ \overset{\text{top}}{(2,3)} \,\Big) \in \mathrm{Sym}(4) \wr \mathrm{Sym}(3)$$



Wreath product - imprimitive action

$$w := \Big(\, \overset{\boxed{1}}{(1,4)},\ \ \overset{\boxed{2}}{(1,2)(3,4)},\ \ \overset{\boxed{3}}{(1,2,3)}\,;\ \ \overset{\text{top}}{\underline{(2,3)}}\, \Big) \in \mathrm{Sym}(4) \wr \mathrm{Sym}(3)$$



Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H$ be the wreath product. The **product action (PA)** is on (n^m) points.

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H$ be the wreath product. The **product action (PA)** is on (n^m) points.

Theorem (Liebeck, 84)

If $G \leq \mathrm{Sym}(N)$ be large-base primitive, then

$$\exists\, \pi \in \mathrm{Sym}(N) : \mathrm{Alt}(\ell)^m \leq G^\pi \leq \mathrm{Sym}(\ell) \wr \mathrm{Sym}(m) =: W,$$

where W acts in (PA) on $N = n^m$ points and $n = \binom{\ell}{k}$.

Let $K \leq \mathrm{Sym}(n)$ and $H \leq \mathrm{Sym}(m)$. Let $W := K \wr H$ be the wreath product. The **product action (PA)** is on (n^m) points.

Theorem (Liebeck, 84)

If $G \leq \mathrm{Sym}(N)$ be large-base primitive, then

$$\exists\, \pi \in \mathrm{Sym}(N) : \mathrm{Alt}(\ell)^m \leq G^\pi \leq \mathrm{Sym}(\ell) \wr \mathrm{Sym}(m) =: W,$$

where W acts in (PA) on $N = n^m$ points and $n = \binom{\ell}{k}$.

$\rightsquigarrow$ **Idea:** find π and translate from product action on (n^m) points into imprimitive action on $(n \cdot m)$ points.

Input: $G = \langle X \rangle$, a black box group such that
$\mathrm{Soc}(K)^m = \mathrm{Soc}(W) \lesssim G \lesssim W := K \wr \mathrm{Sym}(m)$,
K is **almost simple**, and
G acts **transitively** on the set of socle factors.

Input: $G = \langle X \rangle$, a black box group such that
$\mathrm{Soc}(K)^m = \mathrm{Soc}(W) \lesssim G \lesssim W := K \wr \mathrm{Sym}(m)$,
K is **almost simple**, and
G acts **transitively** on the set of socle factors.

Output: $\varphi : G \hookrightarrow W$ such that an image of $g \in G$
under φ is of the form $(w_1, \ldots, w_m; \pi) \in W$.

Wreath product decomposition

Input: $G = \langle X \rangle$, a black box group such that
$\mathrm{Soc}(K)^m = \mathrm{Soc}(W) \lesssim G \lesssim W := K \wr \mathrm{Sym}(m)$,
K is **almost simple**, and
G acts **transitively** on the set of socle factors.

Output: $\varphi : G \hookrightarrow W$ such that an image of $g \in G$
under φ is of the form $(w_1, \ldots, w_m; \pi) \in W$.

Theorem (R., 2025)

If $K = \mathrm{Sym}(n)$ or $K = \mathrm{Alt}(n)$ or $K = \mathrm{PSX}(d, q)$ with q odd, then there exists a polynomial-time* Monte-Carlo algorithm to compute a **wreath product decomposition**.

* under the assumption of a random element oracle and element order oracle

Wreath product decomposition

Input: $G = \langle X \rangle$, a black box group such that
$\mathrm{Soc}(K)^m = \mathrm{Soc}(W) \lesssim G \lesssim W := K \wr \mathrm{Sym}(m)$,
K is **almost simple**, and
G acts **transitively** on the set of socle factors.

Output: $\varphi : G \hookrightarrow W$ such that an image of $g \in G$ under φ is of the form $(w_1, \ldots, w_m; \pi) \in W$.

Theorem (R., 2025)

If $K = \mathrm{Sym}(n)$ or $K = \mathrm{Alt}(n)$ or $K = \mathrm{PSX}(d, q)$ with q odd, then there exists a polynomial-time* Monte-Carlo algorithm to compute a **wreath product decomposition**.

- compression: (n^m) points $\to$ $(n \cdot m)$ points
- seeing components is computationally very useful!

* under the assumption of a random element oracle and element order oracle

Strategy

Problem (simplified)

Input: $G = \langle X \rangle$

Problem (simplified)

Input: $G = \langle X \rangle$

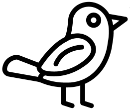
$\exists\ T$ non-abelian simple

$\exists\ m \in \mathbb{N}$:

$T^m \lesssim G \lesssim T \wr \mathrm{Sym}(m)$

& G acts transitively

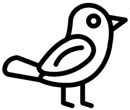
on base components



Problem (simplified)

Input: $G = \langle X \rangle$

$\exists\ T$ non-abelian simple
$\exists\ m \in \mathbb{N}$:
$T^m \lesssim G \lesssim T \wr \mathrm{Sym}(m)$
& G acts transitively
on base components

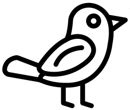


But what is T or m?

Problem (simplified)

Input: $G = \langle X \rangle$

$\exists\ T$ non-abelian simple
$\exists\ m \in \mathbb{N}$:
$T^m \lesssim G \lesssim T \wr \mathrm{Sym}(m)$
& G acts transitively
on base components



But what is T or m?

$$W := T \wr \mathrm{Sym}(m) = T_1 \times \cdots \times T_m \rtimes \mathrm{Sym}(m)$$

Steps of algorithm

Let $\varphi_0 : G \to W = T \wr \mathrm{Sym}(m)$ be an unknown embedding.

Step 1: Single-Component Group

Compute $S \leq G$ with $S^{\varphi_0} = T_1 \cong T$.

Step 2: Constructive Recognition

Compute isomorphism $\lambda : S \to T$

Step 3: Top Group Action

Compute $t_1, \ldots, t_m \in G$ with $(S^{t_i})^{\varphi_0} = T_i$.

Step 4: Embedding

Compute embedding $\varphi : G \to W$.

After a long reduction, the success probability depends on

$$P_2(K) := \frac{\left|\left\{(x,y) \in K^2 : |x|_2 \neq |y|_2\right\}\right|}{|K^2|}$$

After a long reduction, the success probability depends on

$$P_2(K) := \frac{\left|\left\{(x, y) \in K^2 : |x|_2 \neq |y|_2\right\}\right|}{|K^2|}$$

If $K = \mathrm{Sym}(n)$ or $K = \mathrm{Alt}(n)$ or $K = \mathrm{PSX}(d, q)$ with q odd, ...

After a long reduction, the success probability depends on

$$P_2(K) := \frac{\left|\left\{(x,y) \in K^2 : |x|_2 \neq |y|_2\right\}\right|}{|K^2|}$$

If $K = \mathrm{Sym}(n)$ or $K = \mathrm{Alt}(n)$ or $K = \mathrm{PSX}(d,q)$ with q odd, ...

Theorem (R., 2025)

$$\begin{aligned} \text{Let } \mathscr{K} = \quad & \{\mathrm{Alt}(n) : n \geq 5\} \\ \cup\, & \{\mathrm{Sym}(n) : n \geq 5\} \\ \cup\, & \{\mathrm{SX}(d,q) : d > 1, q \text{ odd}\} \\ \cup\, & \{\mathrm{PSX}(d,q) : d > 1, q \text{ odd}\}. \end{aligned}$$

There exists an explicit constant $c > 0$ such that

$$\forall\, K \in \mathscr{K} : P_2(K) \geq c$$

Applications

Input: $G = \langle X \rangle$, a black box group such that
$\mathrm{Soc}(K)^m \cong \mathrm{Soc}(G) \leq G \cong K \wr H$,
K is almost simple, and
$H \leq \mathrm{Sym}(m)$ acts transitively.

Constructive recognition

Input: $G = \langle X \rangle$, a black box group such that
$\mathrm{Soc}(K)^m \cong \mathrm{Soc}(G) \leq G \cong K \wr H$,
K is almost simple, and
$H \leq \mathrm{Sym}(m)$ acts transitively.

Output: $\varphi : G \to G$ isomorphism to natural wreath product
and we can solve fundamental problems for G.

Input: $G = \langle X \rangle$, a black box group such that
$\mathrm{Soc}(K)^m \cong \mathrm{Soc}(G) \le G \cong K \wr H$,
K is almost simple, and
$H \le \mathrm{Sym}(m)$ acts transitively.

Output: $\varphi : G \to \overset{👑}{G}$ isomorphism to natural wreath product
and we can solve fundamental problems for $\overset{👑}{G}$.

Theorem (R., 2025)

If $K = \mathrm{Sym}(n)$ or $K = \mathrm{Alt}(n)$ or $K = \mathrm{PSX}(d, q)$ with q odd, and if $H \le \mathrm{Sym}(m)$ transitive, then constructive recognition of $G = K \wr H$ can be reduced in polynomial-time* to constructive recognition of K and H.

* under the assumption of a random element oracle and element order oracle

Conjugacy-related problems

Results from [Bernhardt, Niemeyer, R., Wollenhaupt, '22]

Let $W := K \wr H$ and $H \leq \mathrm{Sym}(m)$. We describe algorithms ...

- to solve the **conjugacy problem** for two elements in W;
- to compute the **centraliser** of an element in W;
- to compute all **conjugacy classes** of elements in W.

Conjugacy-related problems

Results from [Bernhardt, Niemeyer, R., Wollenhaupt, '22]

Let $W := K \wr H$ and $H \leq \mathrm{Sym}(m)$. We describe algorithms ...

- to solve the **conjugacy problem** for two elements in W;
- to compute the **centraliser** of an element in W;
- to compute all **conjugacy classes** of elements in W.

Main Idea

Break down problems ...

- from wreath product elements onto **wreath cycles**;
- from W onto K and H.

Results from [Bernhardt, Niemeyer, ...

Shameless Advertisement

My algorithms in GAP:

For example, we can enumerate representatives of all **514 976** conjugacy classes in $\mathrm{Sym}(8) \wr \mathrm{Alt}(6)$ in around **5 seconds**,

or compute conjugating elements in $\mathrm{Sym}(25) \wr \mathrm{Alt}(100)$ in a few milliseconds

Thanks for your attention!