Normalisers in Quasipolynomial Time and the Category of Permutation Groups

Sergio Siccha

May 9, 2019

Lehr- und Forschungsgebiet Algebra, RWTH Aachen

Introduction

Goal

Theorem

Let $G = \langle X \rangle \leq \operatorname{Sym}\Omega$ be a primitive group of PA type. The normaliser $N_{\operatorname{Sym}\Omega}(G)$ can be computed in quasipolynomial time $O(n^3 \cdot 2^{2\log n \log\log n} \cdot |X|)$.

Goal

Theorem

Let $G = \langle X \rangle \leq \operatorname{Sym}\Omega$ be a primitive group of PA type. The normaliser $N_{\operatorname{Sym}\Omega}(G)$ can be computed in quasipolynomial time $O(n^3 \cdot 2^{2\log n \log\log n} \cdot |X|)$.

Joint work with Prof. Colva Roney-Dougal.

Recursion for Normalisers

	Intransitive
Mun See Chang	$\updownarrow$
	Transitive
	$\updownarrow$
	Primitive
Me	$\updownarrow$
	Simple

Conventions

- $\log := \log_2$.

Conventions

- $\log := \log_2$.
- All groups and sets are finite!

Conventions

- $\log := \log_2$.
- All groups and sets are finite!
- Ω, Δ denote sets, G, H, T denote groups.

Conventions

- $\log := \log_2$.
- All groups and sets are finite!
- Ω, Δ denote sets, G, H, T denote groups.
    - T *always* denotes a finite non-abelian simple group.

Conventions

- $\log := \log_2$.
- All groups and sets are finite!
- Ω, Δ denote sets, G, H, T denote groups.
  - T *always* denotes a finite non-abelian simple group.
- Functions act from the left $f(x)$ but groups from the right: $\alpha^g = g(\alpha)$.

Complexity and Computational Group Theory

Complexity Classes

We use big O notation.

Complexity Classes

We use big O notation.

Polynomial Time: $f \in O(n^c)$

Complexity Classes

We use big O notation.

Polynomial Time: $f \in O(n^c)$

Quasipolynomial Time: $f \in 2^{O((\log n)^c)}$

Complexity Classes

We use big O notation.

Polynomial Time:	$f \in O(n^c)$
Quasipolynomial Time:	$f \in 2^{O((\log n)^c)}$
Simply Exponential Time:	$f \in 2^{O(n)}$

Complexity Classes

We use big O notation.

Polynomial Time:	$f \in O(n^c)$
Quasipolynomial Time:	$f \in 2^{O((\log n)^c)}$
Simply Exponential Time:	$f \in 2^{O(n)}$
Exponential Time:	$f \in 2^{O(n^c)}$

Normaliser and Subproblems

Simply Exponential

Normalisers of arbitrary groups

Normaliser and Subproblems

Simply Exponential

Normalisers of arbitrary groups

Polynomial

Normalisers of groups with restricted composition factors

Normaliser and Subproblems

Simply Exponential

Normalisers of arbitrary groups

Polynomial

Normalisers of groups with
restricted composition factors

Normalisers of simple groups

Normaliser and Subproblems

Simply Exponential

Normalisers of arbitrary groups

Polynomial

Normalisers of groups with restricted composition factors

Normalisers of simple groups

Quasipolynomial

Normalisers of primitive groups

Normaliser in the Symmetric Group

Why restrict to $N_{\operatorname{Sym}\Omega}(G)$?

PA Type Groups and How To Normalise Them

Definition

Let $G \leq \mathrm{Sym}\,\Omega$ be transitive. G is *primitive* if there exists no non-trivial G-invariant partition of Ω.

Fundamentals

Definition

Let $G \leq \operatorname{Sym}\Omega$ be transitive. G is *primitive* if
there exists no non-trivial G-invariant partition of Ω.

Definition

Let $G \leq \operatorname{Sym}\Omega$ and $H \leq \operatorname{Sym}\Delta$. We call (f, φ) with $f : \Omega \xrightarrow{\sim} \Delta$
and $\varphi\colon G \xrightarrow{\sim} H$ a *permutation isomorphism* if

Fundamentals

Definition

Let $G \leq \operatorname{Sym}\Omega$ be transitive. G is *primitive* if there exists no non-trivial G-invariant partition of Ω.

Definition

Let $G \leq \operatorname{Sym}\Omega$ and $H \leq \operatorname{Sym}\Delta$. We call (f, φ) with $f: \Omega \xrightarrow{\sim} \Delta$ and $\varphi\colon G \xrightarrow{\sim} H$ a *permutation isomorphism* if $\forall g \in G\ \forall \alpha \in \Omega:$

$$f(\alpha^g) = f(\alpha)^{\varphi(g)}.$$

Wreath Products (1)

Definition

Let H be a group and let $K \leq S_\ell$. K acts on H^ℓ by permuting components.

Wreath Products (1)

Definition

Let H be a group and let $K \leq S_\ell$. K acts on H^ℓ by permuting components. The group $H \wr K := H^\ell \rtimes K$ is the *wreath product of H with K*.

Wreath Products (2)

Definition

Let $H \leq \operatorname{Sym} \Delta$ and $K \leq S_\ell$. The base group H^ℓ acts component-wise on Δ^ℓ. The top group K acts on Δ^ℓ by permuting the components.

Wreath Products (2)

Definition

Let $H \leq \operatorname{Sym} \Delta$ and $K \leq S_\ell$. The base group H^ℓ acts component-wise on Δ^ℓ. The top group K acts on Δ^ℓ by permuting the components. This yields the *product action of $H \wr K$ on Δ^ℓ*.

Socles

Definition

Let G be a group. The *socle* of G, denoted $\operatorname{soc} G$, is the group generated by all minimal normal subgroups of G.

Socles

Definition

Let G be a group. The *socle* of G, denoted $\operatorname{soc} G$, is the group generated by all minimal normal subgroups of G.

Theorem

The socle of a primitive group is characteristically simple.

Definition

Let G be a group. The *socle* of G, denoted $\operatorname{soc} G$, is the group generated by all minimal normal subgroups of G.

Theorem

The socle of a primitive group is characteristically simple.

Theorem (O'Nan-Scott)

Let $G \leq \operatorname{Sym}\Omega$ be primitive. We know all possible permutational isomorphism types of

Socles

Definition

Let G be a group. The *socle* of G, denoted $\operatorname{soc} G$, is the group generated by all minimal normal subgroups of G.

Theorem

The socle of a primitive group is characteristically simple.

Theorem (O'Nan-Scott)

Let $G \leq \operatorname{Sym} \Omega$ be primitive. We know all possible permutational isomorphism types of

- $\operatorname{soc} G$,

Socles

Definition

Let G be a group. The *socle* of G, denoted $\operatorname{soc} G$, is the group generated by all minimal normal subgroups of G.

Theorem

The socle of a primitive group is characteristically simple.

Theorem (O'Nan-Scott)

Let $G \leq \operatorname{Sym}\Omega$ be primitive. We know all possible permutational isomorphism types of

- $\operatorname{soc} G$,
- $N_{\operatorname{Sym}\Omega}(\operatorname{soc} G)$.

The AS Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group.

G is a group of *AS type* if

Definition

Let $G \leq \mathrm{Sym}\,\Omega$ be a primitive group.
G is a group of *AS type* if

- G is almost simple,

The AS Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group.

G is a group of *AS type* if

- G is almost simple,
- $\operatorname{soc} G$ is non-abelian simple and non-regular.

Definition

Let $G \leq \mathrm{Sym}\,\Omega$ be a primitive group. G is a group of *PA type* if

The PA Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group. G is a group of *PA type* if

$$G \circlearrowright \Omega \xrightarrow{\sim} \widehat{G} \circlearrowright \Delta^{\ell}$$

with:

The PA Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group. G is a group of *PA type* if

$$G \circlearrowright \Omega \xrightarrow{\sim} \widehat{G} \circlearrowright \Delta^{\ell}$$

with:

- $T \circlearrowright \Delta$,

The PA Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group. G is a group of *PA type* if

$$G \circlearrowright \Omega \xrightarrow{\sim} \widehat{G} \circlearrowright \Delta^{\ell}$$

with:

- $T \circlearrowright \Delta$,
- $\operatorname{soc}\widehat{G} = T^{\ell}$ in component-wise action,

The PA Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group. G is a group of *PA type* if

$$G \circlearrowright \Omega \xrightarrow{\sim} \widehat{G} \circlearrowright \Delta^{\ell}$$

with:

- $T \circlearrowright \Delta$,
- $\operatorname{soc}\widehat{G} = T^{\ell}$ in component-wise action,
- $\widehat{G} \leq N_{\operatorname{Sym}\Delta}(T) \wr S_{\ell}$ in product action.

The PA Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group. G is a group of *PA type* if

$$G \circlearrowright \Omega \xrightarrow{\sim} \widehat{G} \circlearrowright \Delta^{\ell}$$

with:

- $T \circlearrowright \Delta$,
- $\operatorname{soc}\widehat{G} = T^{\ell}$ in component-wise action,
- $\widehat{G} \leq N_{\operatorname{Sym}\Delta}(T) \wr S_{\ell}$ in product action.

Lemma

Let $T^{\ell} \leq \operatorname{Sym}\Delta^{\ell}$ act component-wise, transitively, and non-regularly. Then

The PA Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group. G is a group of *PA type* if

$$G \circlearrowright \Omega \xrightarrow{\sim} \widehat{G} \circlearrowright \Delta^\ell$$

with:

- $T \circlearrowright \Delta$,
- $\operatorname{soc}\widehat{G} = T^\ell$ in component-wise action,
- $\widehat{G} \leq N_{\operatorname{Sym}\Delta}(T) \wr S_\ell$ in product action.

Lemma

Let $T^\ell \leq \operatorname{Sym}\Delta^\ell$ act component-wise, transitively, and non-regularly. Then

$$N_{\operatorname{Sym}\Delta^\ell}(T^\ell) = N_{\operatorname{Sym}\Delta}(T) \wr S_\ell.$$

The Key Idea ...

Construct $N_{\mathrm{Sym}\,\Omega}(\mathrm{soc}\,G)$!

… And Why It Works …

Lemma

Let $G \leq \operatorname{Sym}\Omega$ be primitive of type PA. Then

$$[N_{\operatorname{Sym}\Omega}(\operatorname{soc} G) : \operatorname{soc} G] \leq \sqrt{n} \cdot 2^{\log n \log\log n}.$$

… And Why It Works …

Lemma

Let $G \leq \operatorname{Sym}\Omega$ *be primitive of type PA. Then*

$$[N_{\operatorname{Sym}\Omega}(\operatorname{soc} G) : \operatorname{soc} G] \leq \sqrt{n} \cdot 2^{\log n \log\log n}.$$

Lemma

Let $G = \langle X \rangle \leq \operatorname{Sym}\Omega$ *be primitive of type PA. Let* $N_{\operatorname{Sym}\Omega}(\operatorname{soc} G) = \langle Y \rangle$ *be known.*

… And Why It Works …

Lemma

Let $G \leq \operatorname{Sym}\Omega$ be primitive of type PA. Then

$$[N_{\operatorname{Sym}\Omega}(\operatorname{soc} G) : \operatorname{soc} G] \leq \sqrt{n} \cdot 2^{\log n \log\log n}.$$

Lemma

Let $G = \langle X \rangle \leq \operatorname{Sym}\Omega$ be primitive of type PA. Let $N_{\operatorname{Sym}\Omega}(\operatorname{soc} G) = \langle Y \rangle$ be known.
Then $N_{\operatorname{Sym}\Omega}(G)$ can be computed in time

$$O(n^3 \cdot 2^{2\log n \log\log n} \cdot |X|).$$

… And How To Do It

Compute:

$$\operatorname{soc} G \circlearrowright \Omega \xrightarrow{\sim} T^{\ell} \circlearrowright \Delta^{\ell}$$

The Category of Permutation Groups

Permutation Homomorphisms (1)

Definition

Let $G \leq \operatorname{Sym}\Omega$ and $H \leq \operatorname{Sym}\Delta$.

Permutation Homomorphisms (1)

Definition

Let $G \leq \operatorname{Sym} \Omega$ and $H \leq \operatorname{Sym} \Delta$.

Let $f : \Omega \to \Delta$ be a map and $\varphi \colon G \to H$ be a group hom..

Permutation Homomorphisms (1)

Definition

Let $G \leq \operatorname{Sym} \Omega$ and $H \leq \operatorname{Sym} \Delta$.

Let $f : \Omega \to \Delta$ be a map and $\varphi \colon G \to H$ be a group hom..

The pair (f, φ) is a *permutation hom. from* (G, Ω) *to* (H, Δ) if

Permutation Homomorphisms (1)

Definition

Let $G \leq \operatorname{Sym}\Omega$ and $H \leq \operatorname{Sym}\Delta$.
Let $f\colon \Omega \to \Delta$ be a map and $\varphi\colon G \to H$ be a group hom..
The pair (f, φ) is a *permutation hom. from* (G, Ω) *to* (H, Δ) if $\forall g \in G$:

$$\begin{array}{ccc} \Omega & \xrightarrow{g} & \Omega \\ \big\downarrow{\scriptstyle f} & & \big\downarrow{\scriptstyle f} \\ \Delta & \xrightarrow{\varphi(g)} & \Delta \end{array}$$

Permutation Homomorphisms (2)

Lemma

Let $G \leq \operatorname{Sym}\Omega$ *and* $f : \Omega \to \Delta$. *There exists a permutation hom.* (f, φ) *if and only if*

Permutation Homomorphisms (2)

Lemma

Let $G \leq \operatorname{Sym}\Omega$ and $f : \Omega \to \Delta$. There exists a permutation hom. (f, φ) if and only if

$$\left\{ f^{-1}(\{x\}) \;\middle|\; x \in \operatorname{Im} f \right\}$$

is G-invariant.

Permutation Homomorphisms (3)

$$\begin{array}{ccc} \Omega & \xrightarrow{g} & \Omega \\ \Big\downarrow f & & \Big\downarrow f \\ \Delta & \xrightarrow{\varphi(g)} & \Delta \end{array}$$

Remark

Let $G \leq \operatorname{Sym}\Omega$ *and* $H \leq \operatorname{Sym}\Delta$. *A surjective map* $f : \Omega \twoheadrightarrow \Delta$

Permutation Homomorphisms (3)

$$\begin{array}{ccc} \Omega & \xrightarrow{g} & \Omega \\ \downarrow{\scriptstyle f} & & \downarrow{\scriptstyle f} \\ \Delta & \xrightarrow{\varphi(g)} & \Delta \end{array}$$

Remark

Let $G \leq \operatorname{Sym}\Omega$ and $H \leq \operatorname{Sym}\Delta$. A surjective map $f : \Omega \twoheadrightarrow \Delta$ uniquely determines, if it exists, a permutation homomorphism (f, φ).

Definition

The *category of permutation groups* **PermGrp** consists of

Definition

The *category of permutation groups* **PermGrp** consists of

- all pairs (G, Ω) with $G \leq \operatorname{Sym} \Omega$ as objects

Definition

The *category of permutation groups* **PermGrp** consists of

- all pairs (G, Ω) with $G \leq \operatorname{Sym}\Omega$ as objects
- permutation homomorphisms as morphisms.

Product in PermGrp

Lemma

Let $G \leq \operatorname{Sym}\Omega$ *and* $H \leq \operatorname{Sym}\Delta$ *be permutation groups. Then* $(G \times H, \Omega \times \Delta)$ *with*

Product in PermGrp

Lemma

Let $G \leq \operatorname{Sym}\Omega$ and $H \leq \operatorname{Sym}\Delta$ be permutation groups. Then $(G \times H, \Omega \times \Delta)$ with

$$G \xleftarrow{\pi_1} G \times H \xrightarrow{\pi_2} H$$

$$\Omega \xleftarrow{p_1} \Omega \times \Delta \xrightarrow{p_2} \Delta$$

is a product in **PermGrp**.

Definition

Let $\mathcal{C}$ be a category and X an object of $\mathcal{C}$. A family of morphisms $(f_i)_{i \in I}$ with $f_i \colon X \to X_i$ is called a *cartesian decomposition of X* if

Definition

Let $\mathcal{C}$ be a category and X an object of $\mathcal{C}$. A family of morphisms $(f_i)_{i \in I}$ with $f_i \colon X \to X_i$ is called a *cartesian decomposition of* X if

$$\prod_{i \in I} f_i \colon X \to \prod_{i \in I} X_i$$

is an isomorphism.

Cartesian Decompositions

Definition

Let $\mathcal{C}$ be a category and X an object of $\mathcal{C}$. A family of morphisms $(f_i)_{i\in I}$ with $f_i\colon X \to X_i$ is called a *cartesian decomposition of X* if

$$\prod_{i\in I} f_i\colon X \to \prod_{i\in I} X_i$$

is an isomorphism.

Lemma

A family $(f_i)_{i\in I}$ is a cartesian decomposition of X if and only if X with $(f_i)_{i\in I}$ forms a product in $\mathcal{C}$.

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X with $f_i \colon X \to X_i$.
We call $(f_i)_{i \in I}$ *homogeneous* if

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X with $f_i \colon X \to X_i$.
We call $(f_i)_{i \in I}$ *homogeneous* if

$$X_i \cong X_j \quad \forall i, j \in I.$$

Definition

Let $(f_i)_{i\in I}$ be a cartesian decomposition of X with $f_i\colon X \to X_i$. We call $(f_i)_{i\in I}$ *homogeneous* if

$$X_i \cong X_j \quad \forall i,j \in I.$$

Definition

Let $(f_i)_{i\in I}$ be a cartesian decomposition of X with $f_i\colon X \to X_i$. We call $(f_i)_{i\in I}$ *strictly homogeneous* if

Homogeneous Cartesian Decompositions

Definition

Let $(f_i)_{i\in I}$ be a cartesian decomposition of X with $f_i\colon X \to X_i$.
We call $(f_i)_{i\in I}$ *homogeneous* if

$$X_i \cong X_j \quad \forall i, j \in I.$$

Definition

Let $(f_i)_{i\in I}$ be a cartesian decomposition of X with $f_i\colon X \to X_i$.
We call $(f_i)_{i\in I}$ *strictly homogeneous* if

$$X_i = X_j \quad \forall i, j \in I.$$

Homogeneous Cartesian Decompositions

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X with $f_i \colon X \to X_i$.
We call $(f_i)_{i \in I}$ *homogeneous* if

$$X_i \cong X_j \quad \forall i, j \in I.$$

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X with $f_i \colon X \to X_i$.
We call $(f_i)_{i \in I}$ *strictly homogeneous* if

$$X_i = X_j \quad \forall i, j \in I.$$

$\rightsquigarrow$ Compute a strictly homogeneous cartesian decomposition of $\operatorname{soc} G$!

Constructing the Normaliser of the Socle

The Algorithm - Input

Let $G = \langle\, X \,\rangle \leq \operatorname{Sym}\Omega$ be a primitive group of PA type.

The Algorithm - Input

Let $G = \langle\, X \,\rangle \leq \operatorname{Sym}\Omega$ be a primitive group of PA type.

Note that T^ℓ has *exactly* ℓ minimal normal subgroups.

Algorithm

- $\operatorname{soc} G$

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G \ (= T_1 \times \ldots \times T_\ell)$.

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups* T_i *of* $\operatorname{soc} G$.

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups T_i of* $\operatorname{soc} G$.
- *complements C_i of the T_i,*

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups T_i of* $\operatorname{soc} G$.
- *complements C_i of the T_i, partitions $\Delta_i = \{$orbits of $C_i\}$.*

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\mathrm{soc}\, G \; (= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups* T_i *of* $\mathrm{soc}\, G$.
- *complements* C_i *of the* T_i, *partitions* $\Delta_i = \{$*orbits of* $C_i\}$.
- $Q_i \colon \Omega \to \Delta_i, \; \alpha \mapsto \alpha^{C_i}$

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups* T_i *of* $\operatorname{soc} G$.
- *complements* C_i *of the* T_i, *partitions* $\Delta_i = \{\textit{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon \operatorname{soc} G \to T^{(i)}$.

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups T_i of* $\operatorname{soc} G$.
- *complements C_i of the T_i, partitions $\Delta_i = \{$orbits of $C_i\}$.*
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon \operatorname{soc} G \to T^{(i)}$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups* T_i *of* $\operatorname{soc} G$.
- *complements* C_i *of the* T_i, *partitions* $\Delta_i = \{$*orbits of* $C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon \operatorname{soc} G \to T^{(i)}$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i}$

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups* T_i *of* $\operatorname{soc} G$.
- *complements* C_i *of the* T_i, *partitions* $\Delta_i = \{$*orbits of* $C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon \operatorname{soc} G \to T^{(i)}$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T^{(i)} \to T^{(1)}$.

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups* T_i *of* $\operatorname{soc} G$.
- *complements* C_i *of the* T_i, *partitions* $\Delta_i = \{\text{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon \operatorname{soc} G \to T^{(i)}$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T^{(i)} \to T^{(1)}$.
- $P_i := R_i \circ Q_i \colon \Omega \to \Delta_1$

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G$ $(= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups T_i of* $\operatorname{soc} G$.
- *complements C_i of the T_i, partitions $\Delta_i = \{$orbits of $C_i\}$.*
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon \operatorname{soc} G \to T^{(i)}$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T^{(i)} \to T^{(1)}$.
- $P_i := R_i \circ Q_i \colon \Omega \to \Delta_1 \quad \Rightarrow \quad \varphi_i \colon \operatorname{soc} G \to T^{(1)}$.

Algorithm

- $\operatorname{soc} G \ (= T_1 \times \ldots \times T_\ell)$.
- *minimal normal subgroups* T_i *of* $\operatorname{soc} G$.
- *complements* C_i *of the* T_i, *partitions* $\Delta_i = \{\text{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i, \ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon \operatorname{soc} G \to T^{(i)}$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1, \ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T^{(i)} \to T^{(1)}$.
- $P_i := R_i \circ Q_i \colon \Omega \to \Delta_1 \quad \Rightarrow \quad \varphi_i \colon \operatorname{soc} G \to T^{(1)}$.

$((P_i, \varphi_i))_{i \leq \ell}$ is strictly homogeneous cartesian decomposition.

The Algorithm - Normaliser of Socle

- $((P_i, \varphi_i))_{i \leq \ell}$ is a strictly homogeneous cartesian decomposition of $\operatorname{soc} G$.

The Algorithm - Normaliser of Socle

- $((P_i, \varphi_i))_{i \leq \ell}$ is a strictly homogeneous cartesian decomposition of $\operatorname{soc} G$.

- This yields $\operatorname{soc} G \circlearrowright \Omega \xrightarrow{\sim} T^\ell \circlearrowright \Delta^\ell$.

The Algorithm - Normaliser of Socle

- $((P_i, \varphi_i))_{i \leq \ell}$ is a strictly homogeneous cartesian decomposition of $\operatorname{soc} G$.
- This yields $\operatorname{soc} G \circlearrowright \Omega \xrightarrow{\sim} T^\ell \circlearrowright \Delta^\ell$.
- Construct $N_{\operatorname{Sym}\Delta}(T) \wr S_\ell \leq \operatorname{Sym}\Delta^\ell$.

The Algorithm - Normaliser of Socle

- $((P_i, \varphi_i))_{i \leq \ell}$ is a strictly homogeneous cartesian decomposition of $\operatorname{soc} G$.
- This yields $\operatorname{soc} G \circlearrowright \Omega \xrightarrow{\sim} T^\ell \circlearrowright \Delta^\ell$.
- Construct $N_{\operatorname{Sym} \Delta}(T) \wr S_\ell \leq \operatorname{Sym} \Delta^\ell$.

$\rightsquigarrow N_{\operatorname{Sym} \Omega}(\operatorname{soc} G)$.

Outlook and Summary

Food for Thought

- $G \hookrightarrow H \wr K \leq N_{\mathrm{Sym}\,\Delta}(T) \wr S_\ell$.

Food for Thought

- $G \hookrightarrow H \wr K \leq N_{\mathrm{Sym}\,\Delta}(T) \wr S_\ell$.
  $\rightsquigarrow$ Normalisers in polynomial time?

Food for Thought

- $G \hookrightarrow H \wr K \leq N_{\mathrm{Sym}\,\Delta}(T) \wr S_\ell$.
  $\rightsquigarrow$ Normalisers in polynomial time?

- G leaves a combinatorial cartesian decomposition invariant if and only if it can be embedded into a product action wreath product $S_m \wr S_\ell$.

Food for Thought

- $G \hookrightarrow H \wr K \leq N_{\mathrm{Sym}\,\Delta}(T) \wr S_\ell$.
  $\rightsquigarrow$ Normalisers in polynomial time?

- G leaves a combinatorial cartesian decomposition invariant if and only if it can be embedded into a product action wreath product $S_m \wr S_\ell$.
  $\rightsquigarrow$ Universal property?

What To Take Away

- Category Theory makes (some) algorithms nicer.

What To Take Away

- Category Theory makes (some) algorithms nicer.
- Let G be a primitive group of PA type. We can

What To Take Away

- Category Theory makes (some) algorithms nicer.
- Let G be a primitive group of PA type. We can
  - construct the normaliser of the socle in polynomial time,

What To Take Away

- Category Theory makes (some) algorithms nicer.
- Let G be a primitive group of PA type. We can
    - construct the normaliser of the socle in polynomial time,
    - compute the normaliser in quasipolynomial time.

What To Take Away

- Category Theory makes (some) algorithms nicer.
- Let G be a primitive group of PA type. We can
  - construct the normaliser of the socle in polynomial time,
  - compute the normaliser in quasipolynomial time.
    (maybe even in polynomial time?)

Thank you!

Universal Property of Wreath Products

Let $H \leq \operatorname{Sym}\Delta$, $K \leq \operatorname{Sym}\Gamma$.

$$H^{\Gamma} \longrightarrow G \longleftarrow K$$

$$\Delta^{\Gamma} \longrightarrow \Delta^{\Gamma} \longleftarrow \Gamma$$

Universal Property of Wreath Products

Let $H \leq \operatorname{Sym}\Delta$, $K \leq \operatorname{Sym}\Gamma$.

$$H^\Gamma \longrightarrow G \longleftarrow K$$

$$\Delta^\Gamma \longrightarrow \Delta^\Gamma \longleftarrow \Gamma$$

$$\begin{tikzcd} H^\Gamma \arrow[r] & G \arrow[r] & K \arrow[l, bend left] \end{tikzcd}$$

$$\begin{tikzcd} \Delta \times \Gamma \arrow[r] & \Delta \times \Gamma \arrow[r] & \Gamma \arrow[l, bend left] \end{tikzcd}$$

Combinatorial Cartesian Decompositions (1)

Definition

Let Ω be a set. For each $\gamma \in \Gamma$ let Δ_γ be a partition of Ω with $|\Delta_\gamma| \geq 2$. We say that $\{\Delta_\gamma\}_{\gamma\in\Gamma}$ is a *(combinatorial) cartesian decomposition of* Ω if

Combinatorial Cartesian Decompositions (1)

Definition

Let Ω be a set. For each $\gamma \in \Gamma$ let Δ_γ be a partition of Ω with $|\Delta_\gamma| \geq 2$. We say that $\{\Delta_\gamma\}_{\gamma\in\Gamma}$ is a *(combinatorial) cartesian decomposition of* Ω if for any choice of $\delta_\gamma \in \Delta_\gamma$ we have that

$$\bigcap_{\gamma\in\Gamma} \delta_\gamma$$

is a singleton set.

Combinatorial Cartesian Decompositions (1)

Definition

Let Ω be a set. For each $\gamma \in \Gamma$ let Δ_γ be a partition of Ω with $|\Delta_\gamma| \geq 2$. We say that $\{\Delta_\gamma\}_{\gamma\in\Gamma}$ is a *(combinatorial) cartesian decomposition of* Ω if for any choice of $\delta_\gamma \in \Delta_\gamma$ we have that

$$\bigcap_{\gamma\in\Gamma} \delta_\gamma$$

is a singleton set.

Lemma

There is a one-to-one correspondence between unordered cartesian decompositions and combinatorial cartesian decompositions.

Combinatorial Cartesian Decompositions (2)

Theorem (Praeger, Schneider)
A group $G \leq \mathrm{Sym}\,\Omega$ leaves a homogeneous combinatorial cartesian decomposition invariant if and only if G embeds into a product action wreath product $\mathrm{Sym}\,\Delta \wr \mathrm{Sym}\,\Gamma$.

Complexity Overview

Polynomial:

Base & SGS, Composition Series, Socle

Complexity Overview

Quasipolynomial:

Graph-Iso

Polynomial:

Base & SGS, Composition Series, Socle

Complexity Overview

Quasipolynomial:

String-Iso, Intersection, Centraliser

Graph-Iso

Polynomial:

Base & SGS, Composition Series, Socle

Complexity Overview

Simply Exponential:

Normaliser

Quasipolynomial:

String-Iso, Intersection, Centraliser

Graph-Iso

Polynomial:

Base & SGS, Composition Series, Socle