Normalisers in Quasipolynomial Time and the Category of Permutation Groups

Sergio Siccha

May 9, 2019

Lehr- und Forschungsgebiet Algebra, RWTH Aachen

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.
  - If $T \leq \operatorname{Sym} \Delta$ it acts transitively and non-regularly on Δ.

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.
  - If $T \leq \operatorname{Sym} \Delta$ it acts transitively and non-regularly on Δ.
- Functions act from the left $f(x)$ but groups from the right: $\alpha^g = g(\alpha)$.

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

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

Complexity Classes

We use big O notation.

$$\begin{array}{ll} \text{Polynomial Time:} & f \in O(n^c) \\ \text{Quasipolynomial Time:} & f \in 2^{O((\log n)^c)} \\ \text{Simply Exponential Time:} & f \in 2^{O(n)} \\ \text{Exponential Time:} & f \in 2^{O(n^c)} \end{array}$$

We say a problem A is *polynomial time reducible* to a problem B if there exists a polynomial time algorithm that transforms

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

We say a problem A is *polynomial time reducible* to a problem B if there exists a polynomial time algorithm that transforms

- instances of A into instances of B, and

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

We say a problem A is *polynomial time reducible* to a problem B if there exists a polynomial time algorithm that transforms

- instances of A into instances of B, and
- solutions of B into solutions of A.

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

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

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

PA Type Groups and How To Normalise Them

Definition

Let $G \leq \operatorname{Sym} \Omega$ be transitive. G is called *imprimitive* if there exists a non-trivial G-invariant partition of Ω. Otherwise it is called *primitive*.

Definition

Let $G \leq \operatorname{Sym} \Omega$ be transitive. G is called *imprimitive* if there exists a non-trivial G-invariant partition of Ω. Otherwise it is called *primitive*.

Definition

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

Fundamentals

Definition

Let $G \leq \operatorname{Sym}\Omega$ be transitive. G is called *imprimitive* if there exists a non-trivial G-invariant partition of Ω. Otherwise it is called *primitive*.

Definition

Let $G \leq \operatorname{Sym}\Omega$ and $H \leq \operatorname{Sym}\Delta$ be permutation groups. We call a pair (f, φ) with $f \colon \Omega \to \Delta$ and $\varphi \colon G \to H$ a *permutation isomorphism* if for all $g \in G$ and $\alpha \in \Omega$ holds $f(\alpha^g) = f(\alpha)^{\varphi(g)}$.

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.

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.

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. All possible permutational isomorphism types of $\operatorname{soc} G$ and $N_{\operatorname{Sym} \Omega}(\operatorname{soc} G)$ are known.

Wreath Products (1)

Definition

Let $H \leq \operatorname{Sym} \Delta$ and $K \leq S_\ell$. K acts on the components of H^ℓ.

Wreath Products (1)

Definition

Let $H \leq \operatorname{Sym} \Delta$ and $K \leq S_\ell$. K acts on the components of H^ℓ. The semidirect product $H \wr K = H^\ell \rtimes K$ is called the *wreath product of H with K*.

Wreath Products (1)

Definition

Let $H \leq \operatorname{Sym}\Delta$ and $K \leq S_\ell$. K acts on the components of H^ℓ. The semidirect product $H \wr K = H^\ell \rtimes K$ is called the *wreath product of* H *with* K. H^ℓ is called the *base group*.

Wreath Products (1)

Definition

Let $H \leq \operatorname{Sym}\Delta$ and $K \leq S_\ell$. K acts on the components of H^ℓ. The semidirect product $H \wr K = H^\ell \rtimes K$ is called the *wreath product of H with K*. H^ℓ is called the *base group*. K is called the *top group*.

Wreath Products (1)

Definition

Let $H \leq \operatorname{Sym}\Delta$ and $K \leq S_\ell$. K acts on the components of H^ℓ. The semidirect product $H \wr K = H^\ell \rtimes K$ is called the *wreath product of H with K*. H^ℓ is called the *base group*. K is called the *top group*.

Theorem

$\operatorname{Aut}(T^\ell) \cong \operatorname{Aut}(T) \wr S_\ell$

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 the components of Δ^ℓ.

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 the components of Δ^ℓ. This yields an action of $H \wr K$ on Δ^ℓ which we call the *product action of $H \wr K$*.

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 the components of Δ^ℓ. This yields an action of $H \wr K$ on Δ^ℓ which we call the *product action of $H \wr K$*.

We call the permutation group on Δ^ℓ induced by $H \wr K$ the *product action wreath product of H wirh K* and also denote it by $H \wr K$.

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 the components of Δ^ℓ. This yields an action of $H \wr K$ on Δ^ℓ which we call the *product action of* $H \wr K$.

We call the permutation group on Δ^ℓ induced by $H \wr K$ the *product action wreath product of* H *wirh* K and also denote it by $H \wr K$.

Theorem

Let $H \leq \operatorname{Sym}\Delta$ *and* $K \leq S_\ell$. $H \wr K$ *in product action is primitive if and only if* H *is primitive and non-regular and* K *is transitive.*

The AS Type

Definition

Let $G \leq \operatorname{Sym}\Omega$ be a primitive group.
We say G is a group of *AS type* if $\operatorname{soc} G = T$ is non-abelian simple and G is almost simple.

The PA Type

Definition

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

We say G is a group of *PA type* if it is permutation isomorphic to a group $\widehat{G} \leq \mathrm{Sym}\,\Delta^\ell$ with:

The PA Type

Definition

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

We say G is a group of *PA type* if it is permutation isomorphic to a group $\widehat{G} \leq \operatorname{Sym}\Delta^{\ell}$ with:

- $\operatorname{soc}\widehat{G} = T^{\ell}$,

The PA Type

Definition

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

We say G is a group of *PA type* if it is permutation isomorphic to a group $\widehat{G} \leq \operatorname{Sym}\Delta^{\ell}$ with:

- $\operatorname{soc}\widehat{G} = T^{\ell}$,
- $\widehat{G} \leq N_{\operatorname{Sym}\Delta}(T) \wr S_{\ell}$.

The PA Type

Definition

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

We say G is a group of *PA type* if it is permutation isomorphic to a group $\widehat{G} \leq \operatorname{Sym}\Delta^\ell$ with:

- $\operatorname{soc}\widehat{G} = T^\ell$,
- $\widehat{G} \leq N_{\operatorname{Sym}\Delta}(T) \wr S_\ell$.

Lemma

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

The Key Idea ...

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

... 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. Furthermore let a generating set for* $N_{\operatorname{Sym}\Omega}(\operatorname{soc} G)$ *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. Furthermore let a generating set for* $N_{\operatorname{Sym}\Omega}(\operatorname{soc} G)$ *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:

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

… And How To Do It

Compute:

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

Then:

$$G \qquad \hookrightarrow \quad N_{\operatorname{Sym} \Delta^{\ell}}(T^{\ell})$$

… And How To Do It

Compute:

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

Then:

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

… And How To Do It

Compute:

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

Then:

$$\begin{array}{ccccc} G & \hookrightarrow & N_{\operatorname{Sym}\Delta^\ell}(T^\ell) & = & N_{\operatorname{Sym}\Delta}(T) \wr S_\ell \\ N_{\operatorname{Sym}\Omega}(\operatorname{soc} G) & \xleftarrow{\sim} & N_{\operatorname{Sym}\Delta^\ell}(T^\ell) & & \end{array}$$

The Category of Permutation Groups

Permutation Homomorphisms (1)

Definition

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

Permutation Homomorphisms (1)

Definition

Let $G \leq \operatorname{Sym}\Omega$ and $H \leq \operatorname{Sym}\Delta$ be permutation groups. Let $f\colon \Omega \to \Delta$ be a map and $\varphi\colon G \to H$ be a group hom..

Permutation Homomorphisms (1)

Definition

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

Permutation Homomorphisms (1)

Definition

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

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

Permutation Homomorphisms (2)

$$\begin{tikzcd} \Omega \arrow[r, "g"] \arrow[d, "f"] & \Omega \arrow[d, "f"] \\ \Delta \arrow[r, "\varphi(g)"] & \Delta \end{tikzcd}$$

Remark

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

Permutation Homomorphisms (3)

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 (3)

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.*

Definition

The *category of permutation groups*, denoted **PermGrp**, consists of all pairs (G, Ω) with $G \leq \operatorname{Sym} \Omega$ as objects with 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* (p_1, π_1) *and* (p_2, π_2) *is a product in* **PermGrp**.

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

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 C.

Definition

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

Homogeneous Cartesian Decompositions

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X. We call $(f_i)_{i \in I}$ a *homogeneous cartesian decomposition of X* if
for all $i, j \in I$ we have $f_i(X) \cong f_j(X)$.

Homogeneous Cartesian Decompositions

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X. We call $(f_i)_{i \in I}$ a *homogeneous cartesian decomposition of X* if
for all $i, j \in I$ we have $f_i(X) \cong f_j(X)$.

Definition

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

Homogeneous Cartesian Decompositions

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X. We call $(f_i)_{i \in I}$ a *homogeneous cartesian decomposition of* X if
for all $i, j \in I$ we have $f_i(X) \cong f_j(X)$.

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X. We call $(f_i)_{i \in I}$ a *strongly homogeneous cartesian decomposition of* X if
for all $i, j \in I$ we have $f_i(X) = f_j(X)$.

Homogeneous Cartesian Decompositions

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X. We call $(f_i)_{i \in I}$ a *homogeneous cartesian decomposition of* X if
for all $i, j \in I$ we have $f_i(X) \cong f_j(X)$.

Definition

Let $(f_i)_{i \in I}$ be a cartesian decomposition of X. We call $(f_i)_{i \in I}$ a *strongly homogeneous cartesian decomposition of* X if
for all $i, j \in I$ we have $f_i(X) = f_j(X)$.

$\rightsquigarrow$ Compute a strongly homogeneous cartesian decomposition of the permutation group $\operatorname{soc} G$!

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.

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

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

Algorithm

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

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,*

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 $C_i\}$.*

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* $\Delta_i = \{\textit{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i}$

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* $\Delta_i = \{\textit{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i, \ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon G \to T_i$.

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* $\Delta_i = \{\text{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i, \ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon G \to T_i$.
- $g_1, \ldots, g_\ell \in G$ *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* $\Delta_i = \{\text{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon G \to T_i$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i}$

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* $C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon G \to T_i$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T_i \to T_1$.

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* $C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon G \to T_i$.
- $g_1, \ldots, g_\ell \in G$ *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* $\Delta_i = \{\text{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon G \to T_i$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T_i \to T_1$.
- $P_i := R_i \circ Q_i \colon \Omega \to \Delta_1 \quad \Rightarrow \quad \varphi_i \colon G \to T_1$.

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* $\Delta_i = \{\textit{orbits of } C_i\}$.
- $Q_i \colon \Omega \to \Delta_i,\ \alpha \mapsto \alpha^{C_i} \quad \Rightarrow \quad \psi_i \colon G \to T_i$.
- $g_1, \ldots, g_\ell \in G$ *such that* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T_i \to T_1$.
- $P_i := R_i \circ Q_i \colon \Omega \to \Delta_1 \quad \Rightarrow \quad \varphi_i \colon G \to T_1$.

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

The Algorithm - Normaliser of Socle

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

The Algorithm - Normaliser of Socle

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

The Algorithm - Normaliser of Socle

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

The Algorithm - Normaliser of Socle

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

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

Outlook and Summary

Food for Thought

- $G \hookrightarrow H \wr K \leq N_{\mathsf{Sym}\,\Delta}(T) \wr S_\ell$.
  $\rightsquigarrow$ 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 $S_m \wr S_\ell$.
  $\rightsquigarrow$ Universal property?

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?

- Define a tree data structure via permutation homomorphisms to do many normaliser computations "at once".

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 $G \leq \operatorname{Sym}\Omega$ leaves a homogeneous combinatorial cartesian decomposition invariant if and only if G embeds into a product action wreath product $\operatorname{Sym}\Delta \wr \operatorname{Sym}\Gamma$.