Normalisers in Quasipolynomial Time and the Category of Permutation Groups

Sergio Siccha

May 9, 2019

Lehrstuhl B für Mathematik, RWTH Aachen

Introduction

Conventions

- $\log = \log_2$.
- All groups and sets are finite!
- Capital greek letters denote sets, Capital latin letters denote groups. Lower case letters denote elements or functions.
- Functions from the left $f(x)$ but group actions from the right: $\alpha^g = g(\alpha)$.
    - G acts on functions $\Omega \to \Delta$ via $f^g = f \circ g^{-1}$.
- T *always* denotes a finite non-abelian simple group. If $T \leq \operatorname{Sym}\Delta$ it acts transitively and non-regularly on Δ.

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

Recursion

Intransitive

Mun See Chang $\updownarrow$

Transitive

$\updownarrow$

Primitive

Me $\updownarrow$

Simple

Some Problems in Computational Group Theory

Complexity Classes

Big O Notation

Complexity Classes

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

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

- 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

<u>Simply Exponential:</u>

Permutation-Iso, Normaliser,
Canonical Labeling

<u>Quasipolynomial:</u>

String-Iso, Intersection, Centraliser

Graph-Iso

<u>Polynomial:</u>

Base & SGS, Composition Series, Socle

Normaliser and Subproblems

Simply Exponential

Normalisers of arbitrary groups

Polynomial

Normalisers of groups with restricted composition factors

Quasipolynomial

Normalisers of primitive groups

PA Type Groups and How To Normalise Them

Fundamentals

Definition

FIXME primitive

Definition

FIXME perm iso

Remark

FIXME $f : \Omega \xrightarrow{\sim} \Delta$ induces unique group hom $\operatorname{Sym}\Omega \xrightarrow{\sim} \operatorname{Sym}\Delta$.

Socles

Definition

FIXME Socle

Theorem

The socle of a primitive group is characteristically simple.

Theorem (O'Nan-Scott)

Let $G \leq \mathrm{Sym}\,\Omega$ be primitive. All possible permutational isomorphism types of $\mathrm{soc}\,G$ and $N_{\mathrm{Sym}\,\Omega}(\mathrm{soc}\,G)$ are known.

Wreath Products (1)

Definition

FIXME Abstract Wreath Product

Theorem

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

Wreath Products (2)

Definition

FIXME Imprimitive and product action

Theorem

FIXME H non-regular, K finite. $H \wr K$ primitive iff. H primitive and K transitive.

The PA Type

Definition
FIXME AS

Definition
FIXME PA

Lemma
FIXME Properties of PA type

The Key Idea ...

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

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:

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

Then:

$$\begin{aligned} G \ &\hookrightarrow N_{\operatorname{Sym} \Delta^{\ell}}(T^{\ell}) \\ &= N_{\operatorname{Sym} \Delta}(T) \wr S_{\ell} \end{aligned}$$

The Category of Permutation Groups

Permutation Homomorphisms (1)

Definition

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

FIXME COMMUTINGDIAGRAM

Permutation Homomorphisms (2)

Lemma

Let $G \leq \operatorname{Sym}\Omega$ and $f\colon \Omega \to \Delta$. There exist a group $H \leq \operatorname{Sym}\Delta$ and a group hom. $\varphi\colon G \to H$ such that (f, φ) is a permutation hom. if and only if

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

is G-invariant.

Permutation Homomorphisms (3)

FIXME EXAMPLE

Remark

Let $G \leq \operatorname{Sym}\Omega$ *and* $H \leq \operatorname{Sym}\Delta$*.* $f\colon \Omega \twoheadrightarrow \Delta$ *uniquely determines, if it exists, a group hom.* $\varphi\colon G \to H$ *such that* (f, φ) *is a permutation hom.*

PermGrp

Definition

FIXME Define **PermGrp**.

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**.

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 $(f_i)_{i \in I}$ is a product in C.

Definition

FIXME hom cartesian decomposition. For all $i, j \in I$ have $f_i(X) \cong f_j(X)$

Definition

FIXME strongly hom cartesian decomposition For all $i, j \in I$ have $f_i(X) = f_j(X)$

$\Rightarrow$ Compute a strongly homogeneous cartesian decomposition of $\operatorname{soc} G$!

Combinatorial Cartesian Decompositions

FIXME LEAVE THIS FRAME OUT?

Definition

CCD

Lemma

unordered cd bijection CCD

Theorem (Praeger, Schneider)

G leaves CCD invariant if and only if G embeds into PA WP.

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

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

Algorithm

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

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

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

Algorithm

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

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

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

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G$ $(\cong T_1 \times \ldots \times T_\ell)$.
- *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* $T_i^{g_i} = T_1$.

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G$ $(\cong T_1 \times \ldots \times T_\ell)$.
- *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* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i}$

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

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

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

- $\operatorname{soc} G$ ($\cong T_1 \times \ldots \times T_\ell$).
- *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* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T_i \to T_1$.
- $P_i := R_i \circ Q_i \colon \Omega \to \Delta_1 \quad \Rightarrow \quad \varphi_i \colon G \to T_1$.

The Algorithm - Str. Hom. Cartesian Decomposition

Algorithm

- $\operatorname{soc} G$ $(\cong T_1 \times \ldots \times T_\ell)$.
- *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* $T_i^{g_i} = T_1$.
- $R_i \colon \Delta_i \to \Delta_1,\ \delta \mapsto \delta^{g_i} \quad \Rightarrow \quad \rho_i \colon T_i \to T_1$.
- $P_i := R_i \circ Q_i \colon \Omega \to \Delta_1 \quad \Rightarrow \quad \varphi_i \colon G \to T_1$.

This computes $((P_i, \varphi_i))_{i \leq \ell}$ in polynomial time.

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 $\operatorname{Sym} \Omega$.

Summary

What To Take Away

- Category Theory makes (some) algorithms nicer.

What To Take Away

- Category Theory makes (some) algorithms nicer.
- For primitive groups of PA type we can construct the normaliser of the socle in polynomial time.

What To Take Away

- Category Theory makes (some) algorithms nicer.
- For primitive groups of PA type we can construct the normaliser of the socle in polynomial time.
- For primitive groups of PA type we can compute the normaliser in quasipolynomial (maybe even polynomial?) time.