Dehn's problems in Computational Group Theory

Bettina Eick

TU Braunschweig – Germany

Oldenburg, July 2026

Introduction

Introduction

Groups and Symmetries

Groups and Symmetries

- Groups are a mathematical model for studing symmetries.

Groups and Symmetries

Groups and Symmetries

- Groups are a mathematical model for studing symmetries.
- Example: Permutation groups (Rubik's cube)

Groups and Symmetries

Groups and Symmetries

- Groups are a mathematical model for studing symmetries.
- Example: Permutation groups (Rubik's cube)
- Example: Crystallographic groups (Wall papers and crystals)

Groups and Symmetries

Groups and Symmetries

- Groups are a mathematical model for studing symmetries.
- Example: Permutation groups (Rubik's cube)
- Example: Crystallographic groups (Wall papers and crystals)
- Example: Galois groups (Solving polynomial equations)

Abstract Groups

Cayley 1854

- Introduced the abstract definition for groups:

Abstract Groups

Cayley 1854

- Introduced the abstract definition for groups:
- Groups are sets with an associative multiplication.

Abstract Groups

Cayley 1854

- Introduced the abstract definition for groups:
- Groups are sets with an associative multiplication.
- Defined isomorphism between groups.

Computational group theory

How to represent a group on computer?

- (by multiplication table) (finite groups)

Computational group theory

How to represent a group on computer?

- (by multiplication table) (finite groups)
- as permutation group (finite groups)

Computational group theory

How to represent a group on computer?

- (by multiplication table) (finite groups)
- as permutation group (finite groups)
- as matrix group (over $\mathbb{Z}, \mathbb{Q}, \mathbb{C}, \mathbb{F}_p, \ldots$)

Computational group theory

How to represent a group on computer?

- (by multiplication table) (finite groups)
- as permutation group (finite groups)
- as matrix group (over $\mathbb{Z}, \mathbb{Q}, \mathbb{C}, \mathbb{F}_p, \ldots$)
- as finitely presented group

Computational group theory

How to represent a group on computer?

- (by multiplication table) (finite groups)
- as permutation group (finite groups)
- as matrix group (over $\mathbb{Z}, \mathbb{Q}, \mathbb{C}, \mathbb{F}_p, \ldots$)
- as finitely presented group
- as polycyclic groups (polycyclic groups)

CGT Systems

Which system are available for groups?

- GAP (open-source)

CGT Systems

Which system are available for groups?

- GAP (open-source)
- GAP is part of OSCAR and SageMath (open-source)

CGT Systems

Which system are available for groups?

- GAP (open-source)
- GAP is part of OSCAR and SageMath (open-source)
- Magma (commercial)

Dehn's problems

Dehn (1911) asked:

Given a finitely presented group $G = \langle X \mid R \rangle$:

- Is the word problem (WP) solvable?
  Is $w_1 \equiv w_2$

Dehn's problems

Dehn (1911) asked:

Given a finitely presented group $G = \langle X \mid R \rangle$:

- Is the word problem (WP) solvable?
  Is $w_1 \equiv w_2$
- Is the conjugacy problem (CP) solvable?
  Exists $g \in G$ with $w_1^g \equiv w_2$

Dehn's problems

Dehn (1911) asked:

Given a finitely presented group $G = \langle X \mid R \rangle$:

- Is the word problem (WP) solvable?
  Is $w_1 \equiv w_2$
- Is the conjugacy problem (CP) solvable?
  Exists $g \in G$ with $w_1^g \equiv w_2$
- Is the isomorpism problem (IP) solvable?
  Is $G \cong H$ for some given H?

History

History

- Dehn's problems are all undecidable for fp-groups in general.

History

History

- Dehn's problems are all undecidable for fp-groups in general.
- Novikov (1955), Adian (1957), Rabin (1958).

History

History

- Dehn's problems are all undecidable for fp-groups in general.
- Novikov (1955), Adian (1957), Rabin (1958).
- Special cases might still work.

History

History

- Dehn's problems are all undecidable for fp-groups in general.
- Novikov (1955), Adian (1957), Rabin (1958).
- Special cases might still work.
- The problems became important in all areas of CGT.

Aims of the talk

This talk ...

Aims of the talk

This talk ...

- ... discusses Dehn's problems in different areas of CGT.

Aims of the talk

This talk ...

- ... discusses Dehn's problems in different areas of CGT.
- ... reviews the different form to represent groups.

Aims of the talk

This talk ...

- ... discusses Dehn's problems in different areas of CGT.
- ... reviews the different form to represent groups.
- ... consider the variations of the problems and their applications.

Permutation Groups

Permutation Groups

Permutation groups

Permutation groups

- Subgroups G of $Sym(n)$

Permutation groups

Permutation groups

- Subgroups G of $Sym(n)$
- Typically given by generators

Permutation groups

Permutation groups

- Subgroups G of $Sym(n)$
- Typically given by generators
- Stabilizer chains are a main tool for computation

Permutation groups

Permutation groups

- Subgroups G of $Sym(n)$
- Typically given by generators
- Stabilizer chains are a main tool for computation
- $B = (b_1, \ldots, b_l)$ with $b_i \in \{1, \ldots, n\}$

Permutation groups

Permutation groups

- Subgroups G of $Sym(n)$
- Typically given by generators
- Stabilizer chains are a main tool for computation
- $B = (b_1, \ldots, b_l)$ with $b_i \in \{1, \ldots, n\}$
- $S_1 = G$ and $S_{i+1} = Stab_{S_i}(b_i)$

Permutation groups

Permutation groups

- Subgroups G of $Sym(n)$
- Typically given by generators
- Stabilizer chains are a main tool for computation
- $B = (b_1, \ldots, b_l)$ with $b_i \in \{1, \ldots, n\}$
- $S_1 = G$ and $S_{i+1} = Stab_{S_i}(b_i)$
- B is a *base* if $S_{l+1} = 1$; then obtain stabilizer chain

$$G = S_1 \geq S_2 \geq \ldots \geq S_{l+1} = 1$$

CP

Conjugacy Problem

- Use stabilizer chain to build up search tree for G

CP

Conjugacy Problem

- Use stabilizer chain to build up search tree for G
- Vertices at level i correspond to cosets of S_i in G

CP

Conjugacy Problem

- Use stabilizer chain to build up search tree for G
- Vertices at level i correspond to cosets of S_i in G
- Use backtrack to search conjugating element in tree

CP

Conjugacy Problem

- Use stabilizer chain to build up search tree for G
- Vertices at level i correspond to cosets of S_i in G
- Use backtrack to search conjugating element in tree
- Fast in practise

CP

Conjugacy Problem

- Use stabilizer chain to build up search tree for G
- Vertices at level i correspond to cosets of S_i in G
- Use backtrack to search conjugating element in tree
- Fast in practise
- Seress (2003): 'non-polynomial-time' method

CP

Conjugacy Problem

- Use stabilizer chain to build up search tree for G
- Vertices at level i correspond to cosets of S_i in G
- Use backtrack to search conjugating element in tree
- Fast in practise
- Seress (2003): 'non-polynomial-time' method
- Jefferson, Waldecker, Wilson (2023): fast canonical form

IP

Isomorphism Problem

- Use structure of groups G and H

IP

Isomorphism Problem

- Use structure of groups G and H
- For example: if $\iota : G \to H$ exists, then $\iota(Fit(G)) = Fit(H)$ and $G/Fit(G) \cong H/Fit(H)$

IP

Isomorphism Problem

- Use structure of groups G and H
- For example: if $\iota : G \to H$ exists, then $\iota(Fit(G)) = Fit(H)$ and $G/Fit(G) \cong H/Fit(H)$
- Use reductions of this type to break problem to *small* cases

IP

Isomorphism Problem

- Use structure of groups G and H
- For example: if $\iota : G \to H$ exists, then $\iota(Fit(G)) = Fit(H)$ and $G/Fit(G) \cong H/Fit(H)$
- Use reductions of this type to break problem to *small* cases
- Cannon, Holt (2003)

Matrix Groups

Matrix Groups

Matrix groups

Matrix groups

- Subgroups G of $GL(n, K)$

Matrix groups

Matrix groups

- Subgroups G of $GL(n, K)$
- Typically given by generators

Matrix groups

Matrix groups

- Subgroups G of $GL(n, K)$
- Typically given by generators
- If G is finite, then matrix group recognition applies (or permutation representation)

Matrix groups

Matrix groups

- Subgroups G of $GL(n, K)$
- Typically given by generators
- If G is finite, then matrix group recognition applies (or permutation representation)
- If G is infinite, then general computations are difficult (special cases work well)

Special cases

Special cases

- $G \leq GL(n, \mathbb{Z})$ solvable: use polycyclic presentation
  Assmann, Eick (2005, 2007)

Special cases

Special cases

- $G \leq GL(n, \mathbb{Z})$ solvable: use polycyclic presentation
  Assmann, Eick (2005, 2007)
- CP in full matrix group over field
  Gauss algorithm

Special cases

Special cases

- $G \leq GL(n, \mathbb{Z})$ solvable: use polycyclic presentation
  Assmann, Eick (2005, 2007)
- CP in full matrix group over field
  Gauss algorithm
- CP in full matrix group over $\mathbb{Z}$
  Grunewald (1980), Eick, Hofmann, O'Brien (2019)
  Bley, Hofmann, Johnston (2022)

Polycyclic Groups

Polycyclic Groups

Polycyclic groups

Definition

A group G is polycyclic if it has a series

$$G = G_1 > G_2 > \ldots > G_n > G_{n+1} = \{1\}$$

with $G_{i+1} \trianglelefteq G_i$ and G_i/G_{i+1} cyclic.

Examples

Polycyclic groups

Definition

A group G is polycyclic if it has a series

$$G = G_1 > G_2 > \ldots > G_n > G_{n+1} = \{1\}$$

with $G_{i+1} \trianglelefteq G_i$ and G_i/G_{i+1} cyclic.

Examples

- Finitely generated abelian and nilpotent groups.

Polycyclic groups

Definition

A group G is polycyclic if it has a series

$$G = G_1 > G_2 > \ldots > G_n > G_{n+1} = \{1\}$$

with $G_{i+1} \trianglelefteq G_i$ and G_i/G_{i+1} cyclic.

Examples

- Finitely generated abelian and nilpotent groups.
- Finite solvable groups (in particular p-groups).

Polycyclic groups

Definition

A group G is polycyclic if it has a series

$$G = G_1 > G_2 > \ldots > G_n > G_{n+1} = \{1\}$$

with $G_{i+1} \trianglelefteq G_i$ and G_i/G_{i+1} cyclic.

Examples

- Finitely generated abelian and nilpotent groups.
- Finite solvable groups (in particular p-groups).
- Solvable subgroups of $GL(n, \mathbb{Z})$.

Polycyclic groups

Definition

A group G is polycyclic if it has a series

$$G = G_1 > G_2 > \ldots > G_n > G_{n+1} = \{1\}$$

with $G_{i+1} \trianglelefteq G_i$ and G_i/G_{i+1} cyclic.

Examples

- Finitely generated abelian and nilpotent groups.
- Finite solvable groups (in particular p-groups).
- Solvable subgroups of $GL(n, \mathbb{Z})$.
- Groups from algebraic number fields $O_K \rtimes U_K$.

Polycyclic groups

Definition

A group G is polycyclic if it has a series

$$G = G_1 > G_2 > \ldots > G_n > G_{n+1} = \{1\}$$

with $G_{i+1} \trianglelefteq G_i$ and G_i/G_{i+1} cyclic.

Examples

- Finitely generated abelian and nilpotent groups.
- Finite solvable groups (in particular p-groups).
- Solvable subgroups of $GL(n, \mathbb{Z})$.
- Groups from algebraic number fields $O_K \rtimes U_K$.
- Many crystallographic or topological groups.

Properties and Notation

The class of polycyclic groups

is closed under taking subgroups, factor groups and extensions.

Let G be polycyclic

Properties and Notation

The class of polycyclic groups

is closed under taking subgroups, factor groups and extensions.

Let G be polycyclic

- Number of infinite cyclic factors of G = the *Hirsch length*.

Properties and Notation

The class of polycyclic groups

is closed under taking subgroups, factor groups and extensions.

Let G be polycyclic

- Number of infinite cyclic factors of G = the *Hirsch length*.
- $Fit(G)$ is f.g. nilpotent and $G/Fit(G)$ is abelian-by-finite.

Properties and Notation

The class of polycyclic groups

is closed under taking subgroups, factor groups and extensions.

Let G be polycyclic

- Number of infinite cyclic factors of G = the *Hirsch length*.
- $Fit(G)$ is f.g. nilpotent and $G/Fit(G)$ is abelian-by-finite.
- Each subgroup of G is finitely generated.

Representations

Polycyclic groups arise as (some examples):

- Subgroups of $GL(n, \mathbb{F}_p)$ or $GL(n, \mathbb{Q})$.

Representations

Polycyclic groups arise as (some examples):

- Subgroups of $GL(n, \mathbb{F}_p)$ or $GL(n, \mathbb{Q})$.
- Subgroups of $Sym(n)$.

Representations

Polycyclic groups arise as (some examples):

- Subgroups of $GL(n, \mathbb{F}_p)$ or $GL(n, \mathbb{Q})$.
- Subgroups of $Sym(n)$.
- Quotients of finitely presented groups.

Representations

Polycyclic groups arise as (some examples):

- Subgroups of $GL(n, \mathbb{F}_p)$ or $GL(n, \mathbb{Q})$.
- Subgroups of $Sym(n)$.
- Quotients of finitely presented groups.
- Groups with polycyclic presentation.

Polycyclic generating sequences

Let G be polycyclic

Polycyclic generating sequences

Let G be polycyclic

- Let $G = G_1 > \ldots > G_{n+1} = \{1\}$ be a polycyclic series.

Polycyclic generating sequences

Let G be polycyclic

- Let $G = G_1 > \ldots > G_{n+1} = \{1\}$ be a polycyclic series.
- Choose $g_i \in G$ so that $g_i G_{i+1}$ generates G_i/G_{i+1}.

Polycyclic generating sequences

Let G be polycyclic

- Let $G = G_1 > \ldots > G_{n+1} = \{1\}$ be a polycyclic series.
- Choose $g_i \in G$ so that $g_i G_{i+1}$ generates G_i/G_{i+1}.
- Define $r_i = [G_i : G_{i+1}]$ if this is finite and $r_i = 0$ otherwise.

Polycyclic generating sequences

Let G be polycyclic

- Let $G = G_1 > \ldots > G_{n+1} = \{1\}$ be a polycyclic series.
- Choose $g_i \in G$ so that $g_i G_{i+1}$ generates G_i/G_{i+1}.
- Define $r_i = [G_i : G_{i+1}]$ if this is finite and $r_i = 0$ otherwise.

Then

- $(g_1, \ldots, g_n)$ is a **polycyclic generating sequence (Pcgs)**
- $(r_1, \ldots, r_n)$ are its **relative orders**.

Normal forms

Let G polycyclic with Pcgs $(g_1, \ldots, g_n)$

Each $g \in G$ has a unique **normal form**

$$g = g_1^{e_1} \cdots g_n^{e_n}$$

with $e_i \in \mathbb{Z}$ and $e_i \in \{0, \ldots, r_i - 1\}$ if $r_i \in \mathbb{N}$.

Polycyclic presentations

Let G be polycyclic

Each Pcgs of G induces a finite presentation on this Pcgs

- relations are a rewriting system
- allows to compute normal forms of elements

Polycyclic presentations

Let G be polycyclic

Each Pcgs of G induces a finite presentation on this Pcgs

- relations are a rewriting system
- allows to compute normal forms of elements

More precisely: Given the Pcgs $(g_1, \ldots, g_n)$

the relations have the form

$$\begin{aligned} g_i^x g_j^y &= g_j^{t_{i,j,j}(x,y)} \cdots g_n^{t_{i,j,n}(x,y)} \text{ for } j < i, x, y = \pm 1 \\ g_i^{r_i} &= g_{i+1}^{s_{i,j}} \cdots g_n^{s_{i,n}} \text{ if } r_i \in \mathbb{N} \end{aligned}$$

The word problem

The word problem

Given a polycyclic presentations:

- The word problem can be solved.

The word problem

The word problem

Given a polycyclic presentations:

- The word problem can be solved.
- This algorithm (called **Collection**)

The word problem

The word problem

Given a polycyclic presentations:

- The word problem can be solved.
- This algorithm (called **Collection**)
- Fast in practise

The word problem

The word problem

Given a polycyclic presentations:

- The word problem can be solved.
- This algorithm (called **Collection**)
- Fast in practise
- Non-polynomial complexity

The word problem

The word problem

Given a polycyclic presentations:

- The word problem can be solved.
- This algorithm (called **Collection**)
- Fast in practise
- Non-polynomial complexity
- Leedham-Green, Soicher (1990): Collection from the left.

The word problem

The word problem

Given a polycyclic presentations:

- The word problem can be solved.
- This algorithm (called **Collection**)
- Fast in practise
- Non-polynomial complexity
- Leedham-Green, Soicher (1990): Collection from the left.
- Assmann, Linton (2007): Malcev-Collection.

Collection in nilpotent groups

Multiplication with Pcgs $(g_1, \ldots, g_n)$

Collection in nilpotent groups

Multiplication with Pcgs $(g_1, \ldots, g_n)$

- $(g_1^{x_1} \cdots g_n^{x_n})(g_1^{y_1} \cdots g_n^{y_n}) = g_1^{f_1(x,y)} \cdots g_n^{f_n(x,y)}$

Collection in nilpotent groups

Multiplication with Pcgs $(g_1, \ldots, g_n)$

- $(g_1^{x_1} \cdots g_n^{x_n})(g_1^{y_1} \cdots g_n^{y_n}) = g_1^{f_1(x,y)} \cdots g_n^{f_n(x,y)}$
- What type of functions are $f_1, \ldots, f_n$?

Collection in nilpotent groups

Multiplication with Pcgs $(g_1, \ldots, g_n)$

- $(g_1^{x_1} \cdots g_n^{x_n})(g_1^{y_1} \cdots g_n^{y_n}) = g_1^{f_1(x,y)} \cdots g_n^{f_n(x,y)}$
- What type of functions are $f_1, \ldots, f_n$?

Hall (1969)

If G is torsion-free nilpotent and the Pcgs refines a central series with torsion-free quotients, then $f_1, \ldots, f_n$ can be represented by polynomials in $\mathbb{Q}[x, y]$.

Collection in nilpotent groups

Multiplication with Pcgs $(g_1, \ldots, g_n)$

- $(g_1^{x_1} \cdots g_n^{x_n})(g_1^{y_1} \cdots g_n^{y_n}) = g_1^{f_1(x,y)} \cdots g_n^{f_n(x,y)}$
- What type of functions are $f_1, \ldots, f_n$?

Hall (1969)

If G is torsion-free nilpotent and the Pcgs refines a central series with torsion-free quotients, then $f_1, \ldots, f_n$ can be represented by polynomials in $\mathbb{Q}[x, y]$.

Algorithms

- Leedham-Green & Soicher (1998): Deep Thought
- Cant & Eick (2018), extending Sims (1994)

Translation methods

Translation to polycyclic presentation

- Subgroups of $GL(n, \mathbb{Q})$: Assmann & Eick (2005, 2007)

Translation methods

Translation to polycyclic presentation

- Subgroups of $GL(n, \mathbb{Q})$: Assmann & Eick (2005, 2007)
- Subgroups of $Sym(n)$: Sims (1990)

Translation methods

Translation to polycyclic presentation

- Subgroups of $GL(n, \mathbb{Q})$: Assmann & Eick (2005, 2007)
- Subgroups of $Sym(n)$: Sims (1990)
- Subgroups of $GL(n, \mathbb{F}_p)$: similar to subgroups of $Sym(n)$

Translation methods

Translation to polycyclic presentation

- Subgroups of $GL(n, \mathbb{Q})$: Assmann & Eick (2005, 2007)
- Subgroups of $Sym(n)$: Sims (1990)
- Subgroups of $GL(n, \mathbb{F}_p)$: similar to subgroups of $Sym(n)$
- Quotients of finitely presented groups: Lo (1995), others

Conjugacy problem

Conjugacy Problem

- Induction along normal series

Conjugacy problem

Conjugacy Problem

- Induction along normal series
- $N \trianglelefteq G$ with $N \cong \mathbb{F}_p^d$ or $N \cong \mathbb{Z}^d$

Conjugacy problem

Conjugacy Problem

- Induction along normal series
- $N \trianglelefteq G$ with $N \cong \mathbb{F}_p^d$ or $N \cong \mathbb{Z}^d$
- Assume by induction $g^x \equiv h \bmod N$ known

Conjugacy problem

Conjugacy Problem

- Induction along normal series
- $N \trianglelefteq G$ with $N \cong \mathbb{F}_p^d$ or $N \cong \mathbb{Z}^d$
- Assume by induction $g^x \equiv h \bmod N$ known
- Find $(g^x)^y = h$ in G

Conjugacy problem

Conjugacy Problem

- Induction along normal series
- $N \trianglelefteq G$ with $N \cong \mathbb{F}_p^d$ or $N \cong \mathbb{Z}^d$
- Assume by induction $g^x \equiv h \bmod N$ known
- Find $(g^x)^y = h$ in G
- Translates to orbit problem $\overline{G} \leq GL(d, R)$ acts on R^d.

Conjugacy problem

Conjugacy Problem

- Induction along normal series
- $N \trianglelefteq G$ with $N \cong \mathbb{F}_p^d$ or $N \cong \mathbb{Z}^d$
- Assume by induction $g^x \equiv h \bmod N$ known
- Find $(g^x)^y = h$ in G
- Translates to orbit problem $\overline{G} \leq GL(d, R)$ acts on R^d.
- If $R = \mathbb{F}_p$, then orbit is finite

Conjugacy problem

Conjugacy Problem

- Induction along normal series
- $N \trianglelefteq G$ with $N \cong \mathbb{F}_p^d$ or $N \cong \mathbb{Z}^d$
- Assume by induction $g^x \equiv h \bmod N$ known
- Find $(g^x)^y = h$ in G
- Translates to orbit problem $\overline{G} \leq GL(d, R)$ acts on R^d.
- If $R = \mathbb{F}_p$, then orbit is finite
- If $R = \mathbb{Z}$, then use number theory
  Eick, Ostheimer (2003)

Conjugacy by normal form

Conjugacy by normal form

- G finite: normal form by brute force

Conjugacy by normal form

Conjugacy by normal form

- G finite: normal form by brute force
- G nilpotent: normal form by induction
  Schwingel (1994), Eick, Fernandez-Ayala (2025)

Isomorphism problem

Isomorphism problem

Splits in cases:

- G finite p-group: normal form method
  O'Brien (1994)

Isomorphism problem

Isomorphism problem

Splits in cases:

- G finite p-group: normal form method
  O'Brien (1994)
- G finite solvable: normal form method
  Barrera-Acevedo, Dietrich, Horn (2026)

Isomorphism problem

Isomorphism problem

Splits in cases:

- G finite p-group: normal form method
  O'Brien (1994)
- G finite solvable: normal form method
  Barrera-Acevedo, Dietrich, Horn (2026)
- G torsion-free nilpotent, small HL: normal form method
  Eick, Engel (2017)

Isomorphism problem

Isomorphism problem

Splits in cases:

- G finite p-group: normal form method
  O'Brien (1994)
- G finite solvable: normal form method
  Barrera-Acevedo, Dietrich, Horn (2026)
- G torsion-free nilpotent, small HL: normal form method
  Eick, Engel (2017)
- General case: IP is decidable
  Grunewald, Segal (1980)

Practical algorithms

Practical algorithms for polycyclic presented G:

Practical algorithms

Practical algorithms for polycyclic presented G:

- Derived series and other commutator subgroups,

Practical algorithms

Practical algorithms for polycyclic presented G:

- Derived series and other commutator subgroups,
- Torsion subgroup and finite subgroups up to conjugacy,

Practical algorithms

Practical algorithms for polycyclic presented G:

- Derived series and other commutator subgroups,
- Torsion subgroup and finite subgroups up to conjugacy,
- Maximal subgroups and subgroups of finite index,

Practical algorithms

Practical algorithms for polycyclic presented G:

- Derived series and other commutator subgroups,
- Torsion subgroup and finite subgroups up to conjugacy,
- Maximal subgroups and subgroups of finite index,
- Solving the conjugacy problem for elements and subgroups,

Practical algorithms

Practical algorithms for polycyclic presented G:

- Derived series and other commutator subgroups,
- Torsion subgroup and finite subgroups up to conjugacy,
- Maximal subgroups and subgroups of finite index,
- Solving the conjugacy problem for elements and subgroups,
- Fitting subgroup $Fit(G)$,

Practical algorithms

Practical algorithms for polycyclic presented G:

- Derived series and other commutator subgroups,
- Torsion subgroup and finite subgroups up to conjugacy,
- Maximal subgroups and subgroups of finite index,
- Solving the conjugacy problem for elements and subgroups,
- Fitting subgroup $Fit(G)$,
- Center $Z(G)$,

Practical algorithms

Practical algorithms for polycyclic presented G:

- Derived series and other commutator subgroups,
- Torsion subgroup and finite subgroups up to conjugacy,
- Maximal subgroups and subgroups of finite index,
- Solving the conjugacy problem for elements and subgroups,
- Fitting subgroup $Fit(G)$,
- Center $Z(G)$,
- Frattini subgroup $\Phi(G)$,

Practical algorithms

Practical algorithms for polycyclic presented G:

- Derived series and other commutator subgroups,
- Torsion subgroup and finite subgroups up to conjugacy,
- Maximal subgroups and subgroups of finite index,
- Solving the conjugacy problem for elements and subgroups,
- Fitting subgroup $Fit(G)$,
- Center $Z(G)$,
- Frattini subgroup $\Phi(G)$,
- and many more ... see GAP / Polycyclic package

Applications

Applications

AAG Cryptosystem

Anshel, Anshel, Goldfeld (1999)

- Cryptosystem that needs a non-abelian group given by generators

AAG Cryptosystem

Anshel, Anshel, Goldfeld (1999)

- Cryptosystem that needs a non-abelian group given by generators
- with fast multiplication and word problem

AAG Cryptosystem

Anshel, Anshel, Goldfeld (1999)

- Cryptosystem that needs a non-abelian group given by generators
- with fast multiplication and word problem
- slow or impossible conjugacy problem

Group identification

Group identification

- SmallGroups Library: groups of order n for $n \leq 2000$ except 1024.

Group identification

Group identification

- SmallGroups Library: groups of order n for $n \leq 2000$ except 1024.
- For each order a complete, irredundant list of groups.

Group identification

Group identification

- SmallGroups Library: groups of order n for $n \leq 2000$ except 1024.
- For each order a complete, irredundant list of groups.
- Solvable groups are described by pc presentation
  Others are given by generating permutations

Group identification

Group identification

- SmallGroups Library: groups of order n for $n \leq 2000$ except 1024.
- For each order a complete, irredundant list of groups.
- Solvable groups are described by pc presentation
  Others are given by generating permutations
- Group identification determines the number of a given group in the library

Group identification

Group identification

- SmallGroups Library: groups of order n for $n \leq 2000$ except 1024.
- For each order a complete, irredundant list of groups.
- Solvable groups are described by pc presentation
  Others are given by generating permutations
- Group identification determines the number of a given group in the library
- Uses search tree and invariants of a given group