Logspace and compressed-word computations in nilpotent groups

Andrey Nikolaev

Stevens Institute of Technology

New York
March 13, 2015

Joint work with J. Macdonald, A. Miasnikov, S. Vassileva

History

Approaches to algorithmic problems in nilpotent groups:

- Enumerative algorithms via residual finiteness.
  No reasonable complexity estimates (until recently).
- Embedding $G \hookrightarrow UT(n, \mathbb{Z})$.
  Only for torsion-free case. Does not work well for all problems.
- Normal forms via polycyclic/Mal'cev bases.
  Reasonable algorithms, no specific complexity estimates in most cases.

History

Approaches to algorithmic problems in nilpotent groups:

- Enumerative algorithms via residual finiteness.
  No reasonable complexity estimates (until recently).
- Embedding $G \hookrightarrow UT(n, \mathbb{Z})$.
  Only for torsion-free case. Does not work well for all problems.
- Normal forms via polycyclic/Mal'cev bases.
  Reasonable algorithms, no specific complexity estimates in most cases.

History

Approaches to algorithmic problems in nilpotent groups:

- Enumerative algorithms via residual finiteness.
  No reasonable complexity estimates (until recently).
- Embedding $G \hookrightarrow UT(n, \mathbb{Z})$.
  Only for torsion-free case. Does not work well for all problems.
- Normal forms via polycyclic/Mal'cev bases.
  Reasonable algorithms, no specific complexity estimates in most cases.

History

Approaches to algorithmic problems in nilpotent groups:

- Enumerative algorithms via residual finiteness.
  No reasonable complexity estimates (until recently).
- Embedding $G \hookrightarrow UT(n, \mathbb{Z})$.
  Only for torsion-free case. Does not work well for all problems.
- Normal forms via polycyclic/Mal'cev bases.
  Reasonable algorithms, no specific complexity estimates in most cases.

History

Approaches to algorithmic problems in nilpotent groups:

- Enumerative algorithms via residual finiteness.
  No reasonable complexity estimates (until recently).
- Embedding $G \hookrightarrow UT(n, \mathbb{Z})$.
  Only for torsion-free case. Does not work well for all problems.
- Normal forms via polycyclic/Mal'cev bases.
  Reasonable algorithms, no specific complexity estimates in most cases.

History

Approaches to algorithmic problems in nilpotent groups:

- Enumerative algorithms via residual finiteness.
  No reasonable complexity estimates (until recently).
- Embedding $G \hookrightarrow UT(n, \mathbb{Z})$.
  Only for torsion-free case. Does not work well for all problems.
- Normal forms via polycyclic/Mal'cev bases.
  Reasonable algorithms, no specific complexity estimates in most cases.

History

Approaches to algorithmic problems in nilpotent groups:

- Enumerative algorithms via residual finiteness.
  No reasonable complexity estimates (until recently).
- Embedding $G \hookrightarrow UT(n, \mathbb{Z})$.
  Only for torsion-free case. Does not work well for all problems.
- Normal forms via polycyclic/Mal'cev bases.
  Reasonable algorithms, no specific complexity estimates in most cases.

The problems

For G finitely generated nilpotent group.

(I) Compute Mal'cev normal form.
(II) Membership problem.
(III) Compute the kernel of a homomorphism.
(IV) Compute subgroup presentations.
(V) Compute the centralizer of an element.
(VI) Conjugacy (search) problem.

The results

1. Problems (I)-(VI) are decidable
   - in space $O(\log L)$, and simultaneously
   - in time $O(L \log^3 L)$.
2. We give polynomial bounds on the length of outputs.
3. Compressed-word versions of problems (I)-(VI) are decidable in polynomial time.

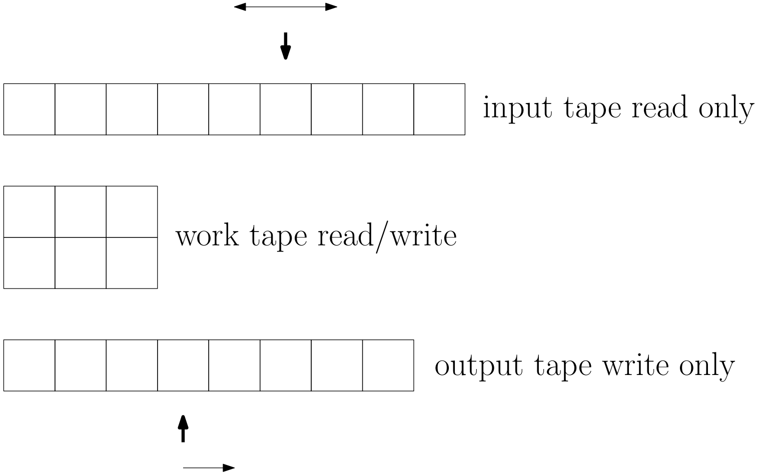
The results

1. Problems (I)-(VI) are decidable
   - in space $O(\log L)$, and simultaneously
   - in time $O(L \log^3 L)$.
2. We give polynomial bounds on the length of outputs.
3. Compressed-word versions of problems (I)-(VI) are decidable in polynomial time.

The results

1. Problems (I)-(VI) are decidable
   - in space $O(\log L)$, and simultaneously
   - in time $O(L \log^3 L)$.
2. We give polynomial bounds on the length of outputs.
3. Compressed-word versions of problems (I)-(VI) are decidable in polynomial time.

Log-space transducers



Logspace $\Rightarrow$ P-time.

- Input length = n.
- Number of cells on work tape $\leq k \log n$.
- Configurations cannot be repeated.
- Total number of configurations $\sim 2^{k \log n} \sim n^k$
- Therefore, $O(n^k)$ time.
- P-time $\stackrel{?}{\Rightarrow}$ logspace: open problem.

Compressed words

- Σ is a set of symbols, called *terminal symbols* with $\epsilon \in \Sigma$.
- A *straight-line program* or *compressed word* $\mathbb{A}$ over Σ consists of
  - ($\mathcal{A}$,<) – ordered finite set, called the set of *non-terminal symbols*,
  - exactly one *production rule* for each $A \in \mathcal{A}$ of the form
    - $A \rightarrow BC$ where $B, C \in \mathcal{A}$ and $B, C < A$ or
    - $A \rightarrow x$ where $x \in \Sigma$.
- The *root* is the greatest non-terminal.
- $\mathrm{eval}(\mathbb{A})$ is the word in Σ^* obtained by starting with the root non-terminal and successively replacing every non-terminal symbol with the right-hand side of its production rule.
- The *size*, $|\mathbb{A}|$, of $\mathbb{A}$ is the number of non-terminal symbols.

Example of compression

Consider the program $\mathbb{B}$ over $\{x\}$ with production rules

$B_n \to B_{n-1}B_{n-1}$,
$B_{n-1} \to B_{n-2}B_{n-2}$,
...
$B_1 \to B_0B_0$,
$B_0 \to x$.

Unravel, $\mathrm{eval}(B_2) = x^4$ and $\mathrm{eval}(\mathbb{B}) = x^{2^n}$.

Size of $\mathbb{B}$ is $n + 1$, size of $\mathrm{eval}(\mathbb{B})$ is 2^n.

Nilpotent group

A group G is called *nilpotent* if it has a central series, i.e. a normal series

$$G = G_1 \rhd G_2 \rhd \ldots \rhd G_c \rhd G_{c+1} = 1 \tag{1}$$

such that $[G, G_i] \leq G_{i+1}$ for all $i = 1, \ldots, c$.

Mal'cev basis

- G_i/G_{i+1} is abelian.
  Pick $a_{i1}, \ldots, a_{im_i}$ a good basis for G_i/G_{i+1}.
- $A = \{a_{11}, a_{12}, \ldots, a_{cm_c}\}$ is a polycyclic generating set for G.
- Relabel A as $\{a_1, \ldots, a_m\}$ for convenience.
- A is a *Mal'cev basis associated to the central series* (1).

Mal’cev basis

- G_i/G_{i+1} is abelian.
  Pick $a_{i1}, \ldots, a_{im_i}$ a good basis for G_i/G_{i+1}.
- $A = \{a_{11}, a_{12}, \ldots, a_{cm_c}\}$ is a polycyclic generating set for G.
- Relabel A as $\{a_1, \ldots, a_m\}$ for convenience.
- A is a *Mal’cev basis associated to the central series* (1).

Normal forms and word problem

Mal'cev normal forms

Let $A = \{a_1, \ldots, a_m\}$ be a Mal'cev basis for G.
"Collect to the left" relations ($i < j$) and "Torsion" relations

$$a_j a_i = a_i a_j \cdot a_{j+1}^{\beta_{j+1}} \cdots a_m^{\beta_m} \qquad a_i^{\tau_i} = a_{i+1}^{\beta_{i+1}} \cdots a_m^{\beta_m}$$

allow to write every element $g \in G$ uniquely as

$$g = a_1^{\alpha_1} \ldots a_m^{\alpha_m},$$

with appropriate $\alpha_i \in \mathbb{Z}$.

$\mathrm{Coord}(g) = (\alpha_1, \ldots, \alpha_m)$ is the *coordinate tuple* of g.

$a_1^{\alpha_1} \ldots a_m^{\alpha_m}$ is the *(Mal'cev) normal form* of g.
Denote $\alpha_i = \mathrm{Coord}_i(g)$.

Mal'cev normal forms

Let $A = \{a_1, \ldots, a_m\}$ be a Mal'cev basis for G.
"Collect to the left" relations ($i < j$) and "Torsion" relations

$$a_j a_i = a_i a_j \cdot a_{j+1}^{\beta_{j+1}} \cdots a_m^{\beta_m} \qquad a_i^{\tau_i} = a_{i+1}^{\beta_{i+1}} \cdots a_m^{\beta_m}$$

allow to write every element $g \in G$ uniquely as

$$g = a_1^{\alpha_1} \ldots a_m^{\alpha_m},$$

with appropriate $\alpha_i \in \mathbb{Z}$.

$\mathrm{Coord}(g) = (\alpha_1, \ldots, \alpha_m)$ is the *coordinate tuple* of g.

$a_1^{\alpha_1} \ldots a_m^{\alpha_m}$ is the *(Mal'cev) normal form* of g.
Denote $\alpha_i = \mathrm{Coord}_i(g)$.

Mal'cev normal forms

Let $A = \{a_1, \ldots, a_m\}$ be a Mal'cev basis for G.
"Collect to the left" relations ($i < j$) and "Torsion" relations

$$a_j a_i = a_i a_j \cdot a_{j+1}^{\beta_{j+1}} \cdots a_m^{\beta_m} \qquad a_i^{\tau_i} = a_{i+1}^{\beta_{i+1}} \cdots a_m^{\beta_m}$$

allow to write every element $g \in G$ uniquely as

$$g = a_1^{\alpha_1} \ldots a_m^{\alpha_m},$$

with appropriate $\alpha_i \in \mathbb{Z}$.

$\mathrm{Coord}(g) = (\alpha_1, \ldots, \alpha_m)$ is the *coordinate tuple* of g.

$a_1^{\alpha_1} \ldots a_m^{\alpha_m}$ is the *(Mal'cev) normal form* of g.
Denote $\alpha_i = \mathrm{Coord}_i(g)$.

Working with Mal'cev coordinates

Let $\{a_1, \ldots, a_m\}$ be a Mal'cev basis for G. Then there are quasi-polynomials (compositions of polynomials and division with a remainder functions)

$$p_1, \ldots, p_m, q_1, \ldots, q_m$$

such that for $\mathrm{Coord}(g) = (\gamma_1, \ldots, \gamma_m)$ and $\mathrm{Coord}(h) = (\delta_1, \ldots, \delta_m)$,

(i) $\mathrm{Coord}_i(gh) = p_i(\gamma_1, \ldots, \gamma_m, \delta_1, \ldots, \delta_m)$,

(ii) $\mathrm{Coord}_i(g^l) = q_i(\gamma_1, \ldots, \gamma_m, l)$, and

(iii) if $\mathrm{Coord}(g) = (0, \ldots, 0, \gamma_k, \ldots, \gamma_m)$, then
   (a) $\forall i < k$, $\mathrm{Coord}_i(gh) = \delta_i$ and $\mathrm{Coord}_k(gh) = \gamma_k + \delta_k$
   (b) $\forall i < k$, $\mathrm{Coord}_i(g^l) = 0$ and $\mathrm{Coord}_k(g^l) = l\gamma_k$.
   Example. $(a_1 a_2 a_3 a_4 a_5) \cdot (a_3^2 a_4 a_5) = a_1 a_2 a_3^3 a_4^? a_5^?$.

Length bound for Mal'cev normal forms

Theorem

Let G be nilpotent group of class c with a Mal'cev basis A. Then, for any word w over A,

$$|\mathrm{Coord}_i(w)| \leq \kappa |w|^c$$

where κ is a constant that depends only on the presentation of G.

- $|\mathrm{Coord}_i(w)|$ is the absolute value of the integer $\mathrm{Coord}_i(w)$;
- $|w|$ is the word length of w in terms of A.
- Number of bits of $\mathrm{Coord}(w)$ is $\sim \log |w|$ (so can store $\mathrm{Coord}(w)$ in memory).

Remark on nilpotent vs. polycyclic

Proposition

Let H be a polycyclic group with polycyclic generators $A = \{a_1, \ldots, a_m\}$. Suppose there is a polynomial $P(n)$ such that if w is a word over $A^{\pm 1}$ of length n then

$$|\mathrm{Coord}_i(w)| \leq P(n)$$

for all $i = 1, 2, \ldots, m$. Then H is virtually nilpotent.

Therefore, the results cannot be immediately extended to polycyclic groups.

Usual vs. Mal'cev encoding

Consider $\mathbb{Z} = \langle a \rangle$.

- Encode a word w as $w = aaaaaaaaa$, so $|w| = 9$.
- Encode a word w as $w = a^9$, or, $w = 9$. So $\|w\| = \lceil \log_2 9 \rceil = 4$.

Similar with nilpotent groups. Let G have Mal'cev basis $a_1, \ldots, a_m$.

- Encode a word w as $w = a_{i_1} a_{i_2} \ldots a_{i_n}$. So $|w| = n$.
- This can be rewritten as $w = a_1 \ldots a_1 a_2 \ldots a_2 \cdots a_m \ldots a_m$.
- Here $|w| \sim n^c$.
- So $w = a_1^{\alpha_1} \ldots a_m^{\alpha_m}$ with $\alpha_1, \ldots \alpha_m \in \mathbb{Z}$.
- Encode $w = (\alpha_1, \ldots, \alpha_m) \in \mathbb{Z}^m$.
- Here $\|w\| \sim O(\log_2 n)$.

What about compressed words?

Usual vs. Mal'cev encoding

Consider $\mathbb{Z} = \langle a \rangle$.

- Encode a word w as $w = aaaaaaaaa$, so $|w| = 9$.
- Encode a word w as $w = a^9$, or, $w = 9$. So $\|w\| = \lceil \log_2 9 \rceil = 4$.

Similar with nilpotent groups. Let G have Mal'cev basis $a_1, \ldots, a_m$.

- Encode a word w as $w = a_{i_1} a_{i_2} \ldots a_{i_n}$. So $|w| = n$.
- This can be rewritten as $w = a_1 \ldots a_1 a_2 \ldots a_2 \cdots a_m \ldots a_m$.
- Here $|w| \sim n^c$.
- So $w = a_1^{\alpha_1} \ldots a_m^{\alpha_m}$ with $\alpha_1, \ldots \alpha_m \in \mathbb{Z}$.
- Encode $w = (\alpha_1, \ldots, \alpha_m) \in \mathbb{Z}^m$.
- Here $\|w\| \sim O(\log_2 n)$.

What about compressed words?

Working with compressed words

The strategy to do the compressed word version of problems is as follows.

- Convert the input SLPs to Mal'cev coordinates.
- Apply algorithms which work with Mal'cev coordinates in binary.
- Convert the output coordinate vectors to SLPs.

What about the size?

- Let L be the length of the SLP $\mathbb{A}$.
- The length of $\mathrm{eval}(\mathbb{A})$ is $\sim 2^L$.
- Each Mal'cev coordinate of $\mathrm{eval}(\mathbb{A})$ is $\sim 2^{cL}$.
- In binary, coordinates are $O(L)$ bits long.

Coordinate tuple $\longleftrightarrow$ SLP.

Theorem

Let G be a f.g. nilpotent group with Mal'cev generating set A.

- There is an algorithm that, given a straight-line program $\mathbb{A}$ over $A^{\pm}$, computes the coordinate vector $\mathrm{Coord}(\mathrm{eval}(\mathbb{A}))$.
- The algorithm runs in time $O(L^3)$, where $L = |\mathbb{A}|$.
- Each coordinate of $\mathrm{eval}(\mathbb{A})$ is expressed as a $O(L)$-bit number.

Computation of normal forms

Theorem

For every finitely generated nilpotent group G, the Mal'cev normal form of a word of length L is computable in

- space $O(\log(L))$ and, simultaneously,
- time $O(L \cdot \log^2 L)$.

Proof

The algorithm – compute coordinates element by element.

- Denote $w = x_1 \cdots x_L$.
- Keep an array $\gamma = (\gamma_1, \ldots, \gamma_m)$ of coordinates in memory.
- At the end of step j, γ holds the coordinates of $x_1 \ldots x_j$.
- For $0 \leq j < L$, compute $\mathrm{Coord}(x_1 \cdots x_j x_{j+1})$ using the p_i with
  - $\mathrm{Coord}(x_1 \cdots x_j) = (\gamma_1, \ldots, \gamma_m)$ and
  - $\mathrm{Coord}(x_{j+1}) = (0, \ldots, 0, \pm 1, 0, \ldots, 0)$.

Complexity

- $|x_1 \cdots x_j| \leq L$, so $\gamma \leq \kappa L^c$ can be stored in logspace.
- $m(L-1)$ total evaluations of the polynomials p_i.
- Each evaluation of p_i requires arithmetic with $O(\log L)$-bit numbers, so can be performed in required space and time.

Proof

The algorithm – compute coordinates element by element.

- Denote $w = x_1 \cdots x_L$.
- Keep an array $\gamma = (\gamma_1, \ldots, \gamma_m)$ of coordinates in memory.
- At the end of step j, γ holds the coordinates of $x_1 \ldots x_j$.
- For $0 \leq j < L$, compute $\mathrm{Coord}(x_1 \cdots x_j x_{j+1})$ using the p_i with
  - $\mathrm{Coord}(x_1 \cdots x_j) = (\gamma_1, \ldots, \gamma_m)$ and
  - $\mathrm{Coord}(x_{j+1}) = (0, \ldots, 0, \pm 1, 0, \ldots, 0)$.

Complexity

- $|x_1 \cdots x_j| \leq L$, so $\gamma \leq \kappa L^c$ can be stored in logspace.
- $m(L-1)$ total evaluations of the polynomials p_i.
- Each evaluation of p_i requires arithmetic with $O(\log L)$-bit numbers, so can be performed in required space and time.

Proof

The algorithm – compute coordinates element by element.

- Denote $w = x_1 \cdots x_L$.
- Keep an array $\gamma = (\gamma_1, \ldots, \gamma_m)$ of coordinates in memory.
- At the end of step j, γ holds the coordinates of $x_1 \ldots x_j$.
- For $0 \leq j < L$, compute $\mathrm{Coord}(x_1 \cdots x_j x_{j+1})$ using the p_i with
  - $\mathrm{Coord}(x_1 \cdots x_j) = (\gamma_1, \ldots, \gamma_m)$ and
  - $\mathrm{Coord}(x_{j+1}) = (0, \ldots, 0, \pm 1, 0, \ldots, 0)$.

Complexity

- $|x_1 \cdots x_j| \leq L$, so $\gamma \leq \kappa L^c$ can be stored in logspace.
- $m(L-1)$ total evaluations of the polynomials p_i.
- Each evaluation of p_i requires arithmetic with $O(\log L)$-bit numbers, so can be performed in required space and time.

Compressed word problem

Corollary

The compressed word problem in every finitely generated nilpotent group is decidable in (sub)cubic time.

Note. Haubold, Lohrey, Mathissen had already observed that the compressed word problem is decidable in polynomial time via embedding in $UT_n(\mathbb{Z})$.

Subgroup membership and matrix reduction

Matrix notation

Let G have Mal'cev basis $\{a_1, \ldots, a_m\}$,
subgroup $H \leq G$ be given as $H = \langle h_1, \ldots, h_n \rangle$.

$$\left\{ \begin{array}{ccc} h_1 = a_1^{\alpha_{11}} & \cdots & a_m^{\alpha_{1m}} \\ \vdots & & \vdots \\ h_n = a_1^{\alpha_{1n}} & \cdots & a_m^{\alpha_{nm}} \end{array} \right. \quad \leftrightsquigarrow \quad \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1m} \\ \vdots & \ddots & \vdots \\ \alpha_{1n} & \cdots & \alpha_{nm} \end{pmatrix} = A.$$

- π_i is the column of the first non-zero entry ('pivot') in row i.
- $(h_1, \ldots, h_n)$ is in *standard form* if the matrix of coordinates A is in row-echelon form and entries above pivots are reduced.
- $(h_1, \ldots, h_n)$ is *full* if for each $1 \leq i \leq m$, the subgroup $H \cap \langle a_i, a_{i+1}, \ldots, a_m \rangle$ is generated by $\{h_j \mid \pi_j \geq i\}$.

Matrix notation

Let G have Mal'cev basis $\{a_1, \ldots, a_m\}$,
subgroup $H \leq G$ be given as $H = \langle h_1, \ldots, h_n \rangle$.

$$\left\{ \begin{array}{ccc} h_1 = a_1^{\alpha_{11}} & \cdots & a_m^{\alpha_{1m}} \\ \vdots & & \vdots \\ h_n = a_1^{\alpha_{1n}} & \cdots & a_m^{\alpha_{nm}} \end{array} \right. \leftrightsquigarrow \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1m} \\ \vdots & \ddots & \vdots \\ \alpha_{1n} & \cdots & \alpha_{nm} \end{pmatrix} = A.$$

- π_i is the column of the first non-zero entry ('pivot') in row i.
- $(h_1, \ldots, h_n)$ is in *standard form* if the matrix of coordinates A is in row-echelon form and entries above pivots are reduced.
- $(h_1, \ldots, h_n)$ is *full* if for each $1 \leq i \leq m$, the subgroup $H \cap \langle a_i, a_{i+1}, \ldots, a_m \rangle$ is generated by $\{h_j \mid \pi_j \geq i\}$.

Uniqueness of standard form

Lemma [Sims]

Let $H \leq G$. There is a unique full sequence $U = (h_1, \ldots, h_s)$ that generates H. Further,

$$H = \{h_1^{\beta_1} \cdots h_s^{\beta_s} \mid \beta_i \in \mathbb{Z}\}$$

and $s \leq m$.

Goal: convert $(h_1, \ldots, h_n)$ to a full sequence in standard form generating the same subgroup.

Uniqueness of standard form

Lemma [Sims]

Let $H \le G$. There is a unique full sequence $U = (h_1, \ldots, h_s)$ that generates H. Further,

$$H = \{h_1^{\beta_1} \cdots h_s^{\beta_s} \mid \beta_i \in \mathbb{Z}\}$$

and $s \le m$.

Goal: convert $(h_1, \ldots, h_n)$ to a full sequence in standard form generating the same subgroup.

Matrix operations

Define three operations on tuples $(h_1, \ldots, h_n)$ of elements of G by their corresponding operations on the associated matrix are:

(1) swap row i with row j;

(2) replace row i by $\mathrm{Coord}(h_i h_j^N)$;

(3) add or remove a trivial row.

All three of these operations preserve the subgroup $\langle h_1, \ldots, h_n \rangle$.

Row-reducing the matrix

Let A be an $n \times m$ matrix. Similar to row-reducing a matrix over $\mathbb{Z}$ (in fact, works same as over $\mathbb{Z}$ in the first column).

- Identify pivot.
- Use the gcd of the pivot column to clear out the column.
- Number of operations $\sim n$.
- Repeat for each column (m times).
- Total number of operations $\sim mn$.

Magnitude of entries may increase

There is an issue:

- When using the operation $h_i \to h_i h_j^N$, the magnitude of the largest entry may increase from M to M^d, $d =$ degree of multiplication polynomials.
- Greatest entry could be size $\sim M^{d^{mn}}$.

Length bound for reduced matrix

Lemma

Let $h_1, \ldots, h_n \in G$ and let R be the standard form of the associated matrix of coordinates. Then every entry, α_{ij}, of R is bounded by

$$|\alpha_{ij}| \leq CL^K,$$

where $L = |h_1| + \cdots + |h_n|$ is the total length of the given elements, and K and C are constants depending on G.

Computing standard form

Lemma

There is an algorithm that, given $h_1, \ldots, h_n \in G$, computes the standard form of the matrix of coordinates in space logarithmic in $L = \sum_{i=1}^{n} |h_i|$ and in time $O(L \log^3 L)$.

- Start with $m \times m$ matrix (constant size).
- Reduce to standard form.
- Add a row and reduce (still constant size).
- Repeat until all n rows accounted for.
- Size never goes beyond $\sim 2m \times m$. Entries are bounded.
- The size of the reduced matrix is $m \times m$.

Computing standard form

Lemma

There is an algorithm that, given $h_1, \ldots, h_n \in G$, computes the standard form of the matrix of coordinates in space logarithmic in $L = \sum_{i=1}^{n} |h_i|$ and in time $O(L \log^3 L)$.

- Start with $m \times m$ matrix (constant size).
- Reduce to standard form.
- Add a row and reduce (still constant size).
- Repeat until all n rows accounted for.
- Size never goes beyond $\sim 2m \times m$. Entries are bounded.
- The size of the reduced matrix is $m \times m$.

Computing standard form

Lemma

There is an algorithm that, given $h_1, \ldots, h_n \in G$, computes the standard form of the matrix of coordinates in space logarithmic in $L = \sum_{i=1}^{n} |h_i|$ and in time $O(L \log^3 L)$.

- Start with $m \times m$ matrix (constant size).
- Reduce to standard form.
- Add a row and reduce (still constant size).
- Repeat until all n rows accounted for.
- Size never goes beyond $\sim 2m \times m$. Entries are bounded.
- The size of the reduced matrix is $m \times m$.

Computing standard form

Lemma

There is an algorithm that, given $h_1, \ldots, h_n \in G$, computes the standard form of the matrix of coordinates in space logarithmic in $L = \sum_{i=1}^{n} |h_i|$ and in time $O(L \log^3 L)$.

- Start with $m \times m$ matrix (constant size).
- Reduce to standard form.
- Add a row and reduce (still constant size).
- Repeat until all n rows accounted for.
- Size never goes beyond $\sim 2m \times m$. Entries are bounded.
- The size of the reduced matrix is $m \times m$.

Computing standard form

Lemma

There is an algorithm that, given $h_1, \ldots, h_n \in G$, computes the standard form of the matrix of coordinates in space logarithmic in $L = \sum_{i=1}^{n} |h_i|$ and in time $O(L \log^3 L)$.

- Start with $m \times m$ matrix (constant size).
- Reduce to standard form.
- Add a row and reduce (still constant size).
- Repeat until all n rows accounted for.
- Size never goes beyond $\sim 2m \times m$. Entries are bounded.
- The size of the reduced matrix is $m \times m$.

Computing standard form

Lemma

There is an algorithm that, given $h_1, \ldots, h_n \in G$, computes the standard form of the matrix of coordinates in space logarithmic in $L = \sum_{i=1}^{n} |h_i|$ and in time $O(L \log^3 L)$.

- Start with $m \times m$ matrix (constant size).
- Reduce to standard form.
- Add a row and reduce (still constant size).
- Repeat until all n rows accounted for.
- Size never goes beyond $\sim 2m \times m$. Entries are bounded.
- The size of the reduced matrix is $m \times m$.

Computing standard form

Lemma

There is an algorithm that, given $h_1, \ldots, h_n \in G$, computes the standard form of the matrix of coordinates in space logarithmic in $L = \sum_{i=1}^{n} |h_i|$ and in time $O(L \log^3 L)$.

- Start with $m \times m$ matrix (constant size).
- Reduce to standard form.
- Add a row and reduce (still constant size).
- Repeat until all n rows accounted for.
- Size never goes beyond $\sim 2m \times m$. Entries are bounded.
- The size of the reduced matrix is $m \times m$.

Membership problem

Theorem

Let G be a finitely generated nilpotent group.
Let $h_1, \ldots, h_n \in G$ and $h \in G$.
Denote $L = |h| + |h_1| + \cdots + |h_n|$ and $H = \langle h_1, \ldots, h_n \rangle$.

- There is an algorithm that, decides whether or not $h \in H$.
- The algorithm runs in space $O(\log L)$ and time $O(L \log^3 L)$.
- If $h \in H$ the algorithm returns the unique expression $h = g_1^{\gamma_1} \cdots g_s^{\gamma_s}$, where $(g_1, \ldots, g_s)$ is the unique full standard-form sequence for H, and the length of h is bounded by a degree $2m(6c^3)^m$ polynomial function of L.

Proof

- $(h_1, \ldots, h_n) \rightsquigarrow (g_1, \ldots, g_s)$.
- Denote $\mathrm{Coord}(h) = (\beta_1, \ldots, \beta_m)$.
- If $\beta_l \neq 0$ for some $1 \leq l < \pi_1$, then $h \notin H$.
- If $\mathrm{Coord}_{\pi_1}(g_1) \nmid \beta_{\pi_1}$, then $h \notin H$.
- Else, let
$$\gamma_1 = \frac{\beta_{\pi_1}}{\mathrm{Coord}_{\pi_1}(g_1)} \quad h' = g_1^{-\gamma_1} h.$$
- Repeat, replacing h by h' and $(g_1, \ldots, g_s)$ by $(g_2, \ldots, g_s)$.

Proof

- $(h_1, \ldots, h_n) \rightsquigarrow (g_1, \ldots, g_s)$.
- Denote $\mathrm{Coord}(h) = (\beta_1, \ldots, \beta_m)$.

- If $\beta_l \neq 0$ for some $1 \leq l < \pi_1$, then $h \notin H$.
- If $\mathrm{Coord}_{\pi_1}(g_1) \nmid \beta_{\pi_1}$, then $h \notin H$.
- Else, let
$$\gamma_1 = \frac{\beta_{\pi_1}}{\mathrm{Coord}_{\pi_1}(g_1)} \quad h' = g_1^{-\gamma_1} h.$$
- Repeat, replacing h by h' and $(g_1, \ldots, g_s)$ by $(g_2, \ldots, g_s)$.

Compressed word membership problem

Theorem

There is an algorithm that, given compressed words $\mathbb{A}_1, \ldots, \mathbb{A}_n, \mathbb{B}$ over a fixed finitely generated nilpotent group G, decides in time polynomial in $|\mathbb{B}| + |\mathbb{A}_1| + \ldots + |\mathbb{A}_n|$ whether or not $\mathrm{eval}(\mathbb{B})$ belongs to the subgroup generated by $\mathrm{eval}(\mathbb{A}_1), \ldots, \mathrm{eval}(\mathbb{A}_n)$.

Kernels, centralizers, conjugacy problem

Computing the kernel and pre-image of a homomorphism

- Let G and H be disjoint finitely generated nilpotent groups.
- Let $K = \langle g_1, \ldots, g_n \rangle \leq G$
- We specify a homomorphism $\phi : K \to H$ by a list of elements $h_1, \ldots, h_n \in H$ such that $\phi(g_i) = h_i$ for $i = 1, \ldots, n$.
- Denote $L = |h| + \sum_{i=1}^{m}(|h_i| + |g_i|)$.

Theorem

There is an algorithm that, given an element $h \in H$ guaranteed to be in the image of ϕ,

(i) computes a generating set X for the kernel of ϕ, and

(ii) computes an element $g \in G$ such that $\phi(g) = h$.

The algorithm runs in space $O(\log L)$ and time $O(L \log^3 L)$.

Computing subgroup presentation

Theorem

Let G be a finitely presented nilpotent group. Let $g_1, \ldots, g_n$ be finite set of elements of G. Denote $L = \sum_{i=1}^{n} |g_i|$. There is an algorithm that computes a presentation for the subgroup $\langle g_1, \ldots, g_n \rangle$. The algorithm runs in space $O(\log L)$ and time $O(L \log^3 L)$.

- Let $N = \langle x_1, \ldots, x_n \rangle$ be the free nilpotent group of class c.
- Define $\phi : N \to G$ by $x_i \mapsto g_i$.
- Compute $\ker \phi$.
- $N / \ker \phi \simeq \mathrm{im}\phi \simeq \langle g_1, \ldots, g_n \rangle$.

Presentation for compressed-word subgroups

Theorem

- Let G be a finitely presented nilpotent group.
- Let $\mathbb{A}_1, \ldots, \mathbb{A}_n$ be a finite set of straight-line programs over G.
- Denote $L = \sum_{i=1}^{n} |\mathbb{A}_i|$.

There is an algorithm that

- computes a presentation for $\langle \mathrm{eval}(\mathbb{A}_1), \ldots, \mathrm{eval}(\mathbb{A}_n) \rangle$,
- runs in time polynomial in L, and
- the size of the presentation is bounded by a polynomial of L.

Note. Size of presentation = number of generators plus sum of the lengths of the relators.

An example on encoding presentations for SLPs

- When working with SLPs, we get the relators as SLPs.
- How do we write down a presentation involving these relators?

Example. Suppose the following SLP is a relator.

$$\mathbb{A} = \{A_1 \to A_2A_3; \quad A_2 \to A_3A_4; \quad A_3 \to A_4A_4; \quad A_4 \to x\}.$$

Then $\mathrm{eval}(\mathbb{A}) = x^5$ and $|\mathrm{eval}(\mathbb{A})| \sim 2^L$.
To write a presentation using this relator we might do the following.

(1) $\langle x | xxxxx \rangle$ (but the length here is $\sim 2^L$), so bad. Or,

(2) $\langle x | \mathbb{A} \rangle$ (but this mixes encodings), so bad.

(3) $\left\langle x, a_1, a_2, a_3, a_4 | a_1 = 1, \begin{array}{ll} a_1 = a_2a_3, & a_2 = a_3a_4, \\ a_3 = a_4a_4, & a_4 = x \end{array} \right\rangle$. Size $O(L)$.

An example on encoding presentations for SLPs

- When working with SLPs, we get the relators as SLPs.
- How do we write down a presentation involving these relators?

Example. Suppose the following SLP is a relator.

$$\mathbb{A} = \{A_1 \to A_2A_3; \quad A_2 \to A_3A_4; \quad A_3 \to A_4A_4; \quad A_4 \to x\}.$$

Then $\mathrm{eval}(\mathbb{A}) = x^5$ and $|\mathrm{eval}(\mathbb{A})| \sim 2^L$.
To write a presentation using this relator we might do the following.

(1) $\langle x | xxxxx \rangle$ (but the length here is $\sim 2^L$), so bad. Or,

(2) $\langle x | \mathbb{A} \rangle$ (but this mixes encodings), so bad.

(3) $\left\langle x, a_1, a_2, a_3, a_4 | a_1 = 1, \begin{array}{ll} a_1 = a_2a_3, & a_2 = a_3a_4, \\ a_3 = a_4a_4, & a_4 = x \end{array} \right\rangle$. Size $O(L)$.

Conjugacy problem

- A group is *conjugately separable* if whenever two elements are not conjugate, there is a finite quotient in which they are not conjugate.
- Gives rise to an enumerative algorithm to decide CP.
- F.g. nilpotent groups are conjugately separable (Remeslennikov '69, Formanek '76).
- Sims '94 gave an algorithm based on matrix reductions and homomorphisms.
- Complexity not analysed.

Computing centralizers

Theorem

- Let G be a f.p. nilpotent group with Mal'cev basis of length m.
- Let $g \in G$.
- Denote $L = |g|$.

There is an algorithm that

- computes a generating set X for the centralizer of g in G,
- runs in space $O(\log L)$ and time $O(L \log^2 L)$.
- X contains at most m elements, and
- there is a degree $(6mc^2)^{m^2}$ polynomial function of L that bounds the length of each element of X.

The conjugacy problem is logspace decidable

Theorem

- Let G be a finitely presented nilpotent group.
- Let $g, h \in G$ be given as words.
- Denote $L = |g| + |h|$.

There is an algorithm that

- (i) produces $u \in G$ such that $g = u^{-1}hu$, or
  (ii) determines that no such element u exists,
- runs in space $O(\log L)$ and time $O(L \log^2 L)$, and
- the word length of u is bounded by a degree $2^m(6mc^2)^{m^2}$ polynomial function of L.

Compressed-word CP is polynomial-time decidable

Theorem

Let G be a finitely presented nilpotent group. There is an algorithm that, given two straight-line programs $\mathbb{A}$ and $\mathbb{B}$ over G, determines in time polynomial in $n = |\mathbb{A}| + |\mathbb{B}|$ whether or not $\mathrm{eval}(\mathbb{A})$ and $\mathrm{eval}(\mathbb{B})$ are conjugate in G. If so, a straight-line program over G of size polynomial in n producing a conjugating element is returned.

Presentation-uniform algorithms

Presentation-uniform algorithms

Main issue: given an arbitrary presentation of a nilpotent group, to use the above algorithms, one needs to find a "good" presentation first.

Theorem

Let c, r be fixed. There is a polynomial time algorithm that, given a group presentation $\langle X \mid R \rangle$ for G in the class of r-generated class $\leq c$ nilpotent groups, produces a "good" presentation for G.

Presentation-uniform algorithms

Corollary

Let Π denote any of the problems (I)–(VI). For all $c, r \in \mathbb{N}$, there is a polynomial time algorithm that, given a finite presentation $\langle X|R\rangle$ of a group in $\mathcal{N}_{c,r}$ and input of Π as words in X, solves Π in $\langle X|R\rangle$ on that input.

(I) Compute Mal'cev normal form.
(II) Membership problem.
(III) Compute the kernel of a homomorphism.
(IV) Compute subgroup presentations.
(V) Compute the centralizer of an element.
(VI) Conjugacy (search) problem.

Free stuff

Finite separability growth

Theorem [K.Bou-Rabbee, D.Studenmund 2014]

Let G be a finitely generated nilpotent group. There is a polynomial $P(n)$ such that the ball B_n in the Cayley graph of G is discriminated in a finite group of order $\leq P(n)$.

Add our poly bounds, obtain

Theorem

Let G be a finitely generated nilpotent group. There is a polynomial R such that $g, h \in G$ are conjugate in G if and only if their images are conjugate in some finite quotient $\bar{G}$ of G with $|\bar{G}| \leq R(|g| + |h|)$.

Theorem

Let G be a finitely generated nilpotent group. There is a polynomial S such that $h \in G$ belongs to a subgroup generated by $h_1, \ldots, h_k \in G$ if and only if the same is true for their images in some finite quotient $\bar{G}$ of G with $|\bar{G}| \leq S(|g| + |h|)$.

Add our poly bounds, obtain

Theorem

Let G be a finitely generated nilpotent group. There is a polynomial R such that $g, h \in G$ are conjugate in G if and only if their images are conjugate in some finite quotient $\bar{G}$ of G with $|\bar{G}| \leq R(|g| + |h|)$.

Theorem

Let G be a finitely generated nilpotent group. There is a polynomial S such that $h \in G$ belongs to a subgroup generated by $h_1, \ldots, h_k \in G$ if and only if the same is true for their images in some finite quotient $\bar{G}$ of G with $|\bar{G}| \leq S(|g| + |h|)$.

Add our poly bounds, obtain

Theorem

Let G be a finitely generated nilpotent group. There is a polynomial R such that $g, h \in G$ are conjugate in G if and only if their images are conjugate in some finite quotient $\bar{G}$ of G with $|\bar{G}| \leq R(|g| + |h|)$.

Theorem

Let G be a finitely generated nilpotent group. There is a polynomial S such that $h \in G$ belongs to a subgroup generated by $h_1, \ldots, h_k \in G$ if and only if the same is true for their images in some finite quotient $\bar{G}$ of G with $|\bar{G}| \leq S(|g| + |h|)$.

Theorem [I.Bumagin 2014]

Conjugacy (search) problem is polynomial time solvable in relatively hyperbolic groups if it is solvable in polynomial time in each parabolic subgroup.

Add our results, get

Theorem

Conjugacy (search) problem is polynomial time solvable in groups hyperbolic relative to nilpotent subgroups.

Conjugacy in relatively hyperbolic groups

Theorem [I.Bumagin 2014]

Conjugacy (search) problem is polynomial time solvable in relatively hyperbolic groups if it is solvable in polynomial time in each parabolic subgroup.

Add our results, get

Theorem

Conjugacy (search) problem is polynomial time solvable in groups hyperbolic relative to nilpotent subgroups.

Not so free stuff

Further developments

Work in progress (J.M, A.M, A.N., S.V., K.Blaney, A.Garreta, F.Gul, D.Ovchinnikov, M.Sohrabi):

- Finite separability questions.
- Algorithms to compute torsion subgroup and isolators.
- Algorithms to compute intersections of subgroups and cosets.
- Algorithms to solve subgroup conjugacy, simultaneous conjugacy, to compute normalizers.
- Distortion of embeddings into $UT(n, \mathbb{Z})$.
- (Un)solvability of systems of quadratic equations over nilpotent groups.
- Fast practical algorithms in generalized Heisenberg groups.

Further developments

Work in progress (J.M, A.M, A.N., S.V., K.Blaney, A.Garreta, F.Gul, D.Ovchinnikov, M.Sohrabi):

- Finite separability questions.
- Algorithms to compute torsion subgroup and isolators.
- Algorithms to compute intersections of subgroups and cosets.
- Algorithms to solve subgroup conjugacy, simultaneous conjugacy, to compute normalizers.
- Distortion of embeddings into $UT(n, \mathbb{Z})$.
- (Un)solvability of systems of quadratic equations over nilpotent groups.
- Fast practical algorithms in generalized Heisenberg groups.

Further developments

Work in progress (J.M, A.M, A.N., S.V., K.Blaney, A.Garreta, F.Gul, D.Ovchinnikov, M.Sohrabi):

- Finite separability questions.
- Algorithms to compute torsion subgroup and isolators.
- Algorithms to compute intersections of subgroups and cosets.
- Algorithms to solve subgroup conjugacy, simultaneous conjugacy, to compute normalizers.
- Distortion of embeddings into $UT(n, \mathbb{Z})$.
- (Un)solvability of systems of quadratic equations over nilpotent groups.
- Fast practical algorithms in generalized Heisenberg groups.

Further developments

Work in progress (J.M, A.M, A.N., S.V., K.Blaney, A.Garreta, F.Gul, D.Ovchinnikov, M.Sohrabi):

- Finite separability questions.
- Algorithms to compute torsion subgroup and isolators.
- Algorithms to compute intersections of subgroups and cosets.
- Algorithms to solve subgroup conjugacy, simultaneous conjugacy, to compute normalizers.
- Distortion of embeddings into $UT(n, \mathbb{Z})$.
- (Un)solvability of systems of quadratic equations over nilpotent groups.
- Fast practical algorithms in generalized Heisenberg groups.

Further developments

Work in progress (J.M, A.M, A.N., S.V., K.Blaney, A.Garreta, F.Gul, D.Ovchinnikov, M.Sohrabi):

- Finite separability questions.
- Algorithms to compute torsion subgroup and isolators.
- Algorithms to compute intersections of subgroups and cosets.
- Algorithms to solve subgroup conjugacy, simultaneous conjugacy, to compute normalizers.
- Distortion of embeddings into $UT(n, \mathbb{Z})$.
- (Un)solvability of systems of quadratic equations over nilpotent groups.
- Fast practical algorithms in generalized Heisenberg groups.

Further developments

Work in progress (J.M, A.M, A.N., S.V., K.Blaney, A.Garreta, F.Gul, D.Ovchinnikov, M.Sohrabi):

- Finite separability questions.
- Algorithms to compute torsion subgroup and isolators.
- Algorithms to compute intersections of subgroups and cosets.
- Algorithms to solve subgroup conjugacy, simultaneous conjugacy, to compute normalizers.
- Distortion of embeddings into $UT(n, \mathbb{Z})$.
- (Un)solvability of systems of quadratic equations over nilpotent groups.
- Fast practical algorithms in generalized Heisenberg groups.

Further developments

Work in progress (J.M, A.M, A.N., S.V., K.Blaney, A.Garreta, F.Gul, D.Ovchinnikov, M.Sohrabi):

- Finite separability questions.
- Algorithms to compute torsion subgroup and isolators.
- Algorithms to compute intersections of subgroups and cosets.
- Algorithms to solve subgroup conjugacy, simultaneous conjugacy, to compute normalizers.
- Distortion of embeddings into $UT(n, \mathbb{Z})$.
- (Un)solvability of systems of quadratic equations over nilpotent groups.
- Fast practical algorithms in generalized Heisenberg groups.

Further developments

Work in progress (J.M, A.M, A.N., S.V., K.Blaney, A.Garreta, F.Gul, D.Ovchinnikov, M.Sohrabi):

- Finite separability questions.
- Algorithms to compute torsion subgroup and isolators.
- Algorithms to compute intersections of subgroups and cosets.
- Algorithms to solve subgroup conjugacy, simultaneous conjugacy, to compute normalizers.
- Distortion of embeddings into $UT(n, \mathbb{Z})$.
- (Un)solvability of systems of quadratic equations over nilpotent groups.
- Fast practical algorithms in generalized Heisenberg groups.