Compresed word problem in wreath products

Markus Lohrey
Leipzig, Germany

May 30, 2013

The word problem for groups

In this talk: Only finitely generated groups

The word problem for groups

In this talk: Only finitely generated groups

Let G be a finitely generated group, and let Σ be a finite symmetric generating set for G.

The word problem for groups

In this talk: Only finitely generated groups

Let G be a finitely generated group, and let Σ be a finite symmetric generating set for G.

Word problem for G, $\mathrm{WP}(G)$ (Dehn 1910)

INPUT: Word $w \in \Sigma^*$

QUESTION: $w = 1$ in G ?

The word problem for groups

In this talk: Only finitely generated groups

Let G be a finitely generated group, and let Σ be a finite symmetric generating set for G.

Word problem for G, $\mathrm{WP}(G)$ (Dehn 1910)

INPUT: Word $w \in \Sigma^*$

QUESTION: $w = 1$ in G ?

Decidability/complexity of the word problem is independent of the generating set Σ.

The word problem for groups

In this talk: Only finitely generated groups

Let G be a finitely generated group, and let Σ be a finite symmetric generating set for G.

Word problem for G, $\mathrm{WP}(G)$ (Dehn 1910)

INPUT: Word $w \in \Sigma^*$

QUESTION: $w = 1$ in G ?

Decidability/complexity of the word problem is independent of the generating set Σ.

Novikov 1958, Boone 1959: There exists a finitely presented group with an undecidable word problem.

Some decidable word problems

Some classes of group with decidable word problems:

	computational complexity of word problem

Some decidable word problems

Some classes of group with decidable word problems:

	computational complexity of word problem
finitely generated linear groups	polynomial time (even logarithmic space)

Some decidable word problems

Some classes of group with decidable word problems:

	computational complexity of word problem
finitely generated linear groups	polynomial time (even logarithmic space)
hyperbolic groups	linear time

Some decidable word problems

Some classes of group with decidable word problems:

	computational complexity of word problem
finitely generated linear groups	polynomial time (even logarithmic space)
hyperbolic groups	linear time
automatic groups	quadratic time

Some decidable word problems

Some classes of group with decidable word problems:

	computational complexity of word problem
finitely generated linear groups	polynomial time (even logarithmic space)
hyperbolic groups	linear time
automatic groups	quadratic time
one-relator groups $\langle \Sigma, r \rangle$	primitive recursive

The word search problem

Let $G = \langle \Sigma \mid R \rangle$ be a finitely presented group.

The word search problem

Let $G = \langle \Sigma \mid R \rangle$ be a finitely presented group.

$R \subseteq \Sigma^*$ is the finite set of relators.

The word search problem

Let $G = \langle \Sigma \mid R \rangle$ be a finitely presented group.

$R \subseteq \Sigma^*$ is the finite set of relators.

Word search problem for G, $\mathrm{WSP}(G)$

INPUT: Word $w \in \Sigma^*$

The word search problem

Let $G = \langle \Sigma \mid R \rangle$ be a finitely presented group.

$R \subseteq \Sigma^*$ is the finite set of relators.

Word search problem for G, $\mathrm{WSP}(G)$

INPUT: Word $w \in \Sigma^*$

OUTPUT: If $w \neq 1$ in G then output "NO", otherwise output words $c_1, \ldots, c_n \in \Sigma^*$ and $r_1, \ldots, r_n \in R \cup R^{-1}$ with

$$w = \prod_{i=1}^{n} c_i r_i c_i^{-1} \text{ in } F(\Sigma).$$

The word search problem

Remarks:

- Complexity of the WSP is independent of the finite presentation.

The word search problem

Remarks:

- Complexity of the WSP is independent of the finite presentation.
- A group with a polynomial time WSP must have a polynomial Dehn function.

The word search problem

Remarks:

- Complexity of the WSP is independent of the finite presentation.
- A group with a polynomial time WSP must have a polynomial Dehn function.
- Groups with polynomial time WSP:

The word search problem

Remarks:

- Complexity of the WSP is independent of the finite presentation.
- A group with a polynomial time WSP must have a polynomial Dehn function.
- Groups with polynomial time WSP:
  - f.g. nilpotent groups

The word search problem

Remarks:

- Complexity of the WSP is independent of the finite presentation.
- A group with a polynomial time WSP must have a polynomial Dehn function.
- Groups with polynomial time WSP:
  - f.g. nilpotent groups
  - automatic groups

The compressed word problem

A straight-line program (SLP) over the alphabet Γ is a sequence of definitions $\mathbb{A} = (A_i := \alpha_i)_{1 \leq i \leq n}$, where either $\alpha_i \in \Gamma$ or $\alpha_i = A_j A_k$ for some $j, k < i$.

The compressed word problem

A straight-line program (SLP) over the alphabet Γ is a sequence of definitions $\mathbb{A} = (A_i := \alpha_i)_{1 \leq i \leq n}$, where either $\alpha_i \in \Gamma$ or $\alpha_i = A_j A_k$ for some $j, k < i$.

We write $\mathrm{val}(\mathbb{A})$ for the unique word generated by $\mathbb{A}$.

The compressed word problem

A straight-line program (SLP) over the alphabet Γ is a sequence of definitions $\mathbb{A} = (A_i := \alpha_i)_{1 \leq i \leq n}$, where either $\alpha_i \in \Gamma$ or $\alpha_i = A_j A_k$ for some $j, k < i$.

We write $\mathrm{val}(\mathbb{A})$ for the unique word generated by $\mathbb{A}$.

Example: $\mathbb{A} = (A_1 := b, \;\; A_2 := a, \;\; A_i := A_{i-1} A_{i-2}$ for $3 \leq i \leq 7)$

The compressed word problem

A straight-line program (SLP) over the alphabet Γ is a sequence of definitions $\mathbb{A} = (A_i := \alpha_i)_{1 \leq i \leq n}$, where either $\alpha_i \in \Gamma$ or $\alpha_i = A_j A_k$ for some $j, k < i$.

We write $\mathrm{val}(\mathbb{A})$ for the unique word generated by $\mathbb{A}$.

Example: $\mathbb{A} = (A_1 := b,\ \ A_2 := a,\ \ A_i := A_{i-1}A_{i-2}$ for $3 \leq i \leq 7)$

$$
\begin{aligned}
A_3 &= A_2A_1 = ab \\
A_4 &= A_3A_2 = aba \\
A_5 &= A_4A_3 = abaab \\
A_6 &= A_5A_4 = abaababa \\
A_7 &= A_6A_5 = abaababaabaab = \mathrm{val}(\mathbb{A})
\end{aligned}
$$

The compressed word problem

A straight-line program (SLP) over the alphabet Γ is a sequence of definitions $\mathbb{A} = (A_i := \alpha_i)_{1 \leq i \leq n}$, where either $\alpha_i \in \Gamma$ or $\alpha_i = A_j A_k$ for some $j, k < i$.

We write $\mathrm{val}(\mathbb{A})$ for the unique word generated by $\mathbb{A}$.

Example: $\mathbb{A} = (A_1 := b,\ \ A_2 := a,\ \ A_i := A_{i-1}A_{i-2}$ for $3 \leq i \leq 7)$

$$\begin{aligned} A_3 &= A_2A_1 = ab \\ A_4 &= A_3A_2 = aba \\ A_5 &= A_4A_3 = abaab \\ A_6 &= A_5A_4 = abaababa \\ A_7 &= A_6A_5 = abaababaabaab = \mathrm{val}(\mathbb{A}) \end{aligned}$$

If $|\mathbb{A}|$ is the number of definitions in $\mathbb{A}$, then $|\mathrm{val}(\mathbb{A})| \leq 2^{|\mathbb{A}|}$.

The compressed word problem

Plandowski 1994: The following problem can be solved in polynomial time:

INPUT: SLPs $\mathbb{A}, \mathbb{B}$
QUESTION: $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$?

The compressed word problem

Plandowski 1994: The following problem can be solved in polynomial time:

INPUT: SLPs $\mathbb{A}, \mathbb{B}$
QUESTION: $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$?

The best known algorithm is almost quadratic (Alstrup, Brodal Rauhe 2000).

The compressed word problem

Plandowski 1994: The following problem can be solved in polynomial time:

INPUT: SLPs $\mathbb{A}, \mathbb{B}$
QUESTION: $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$?

The best known algorithm is almost quadratic (Alstrup, Brodal Rauhe 2000).

Let the group G be finitely generated by Σ (symmetric).

The compressed word problem

Plandowski 1994: The following problem can be solved in polynomial time:

INPUT: SLPs $\mathbb{A}, \mathbb{B}$
QUESTION: $\mathrm{val}(\mathbb{A}) = \mathrm{val}(\mathbb{B})$?

The best known algorithm is almost quadratic (Alstrup, Brodal Rauhe 2000).

Let the group G be finitely generated by Σ (symmetric).

Compressed word problem for G, $\mathrm{CWP}(G)$

INPUT: SLP $\mathbb{A}$ over Σ

QUESTION: $\mathrm{val}(\mathbb{A}) = 1$ in G?

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:
  - f.g. nilpotent groups

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:
  - f.g. nilpotent groups (here, CWP even belongs to $\mathbf{NC}^2$)

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:
  - f.g. nilpotent groups (here, CWP even belongs to $\mathbf{NC}^2$)
  - right-angled Artin groups (RAAGs)

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:
  - f.g. nilpotent groups (here, CWP even belongs to $\mathbf{NC}^2$)
  - right-angled Artin groups (RAAGs)
  - finite extensions of subgroups of RAAGs
    (hence: virtually special groups)

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:
  - f.g. nilpotent groups (here, CWP even belongs to $\mathbf{NC}^2$)
  - right-angled Artin groups (RAAGs)
  - finite extensions of subgroups of RAAGs
    (hence: virtually special groups)
  - Coxeter groups

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:
  - f.g. nilpotent groups (here, CWP even belongs to $\mathbf{NC}^2$)
  - right-angled Artin groups (RAAGs)
  - finite extensions of subgroups of RAAGs (hence: virtually special groups)
  - Coxeter groups
  - fully residually free groups (independently shown by Macdonald 2010)

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:
  - f.g. nilpotent groups (here, CWP even belongs to $\mathbf{NC}^2$)
  - right-angled Artin groups (RAAGs)
  - finite extensions of subgroups of RAAGs (hence: virtually special groups)
  - Coxeter groups
  - fully residually free groups (independently shown by Macdonald 2010)
  - fundamental groups of hyperbolic 3-manifolds

The compressed word problem

Remarks:

- Complexity of the CWP is independent of the generating set.
- Groups with polynomial time CWP:
  - f.g. nilpotent groups (here, CWP even belongs to $\mathbf{NC}^2$)
  - right-angled Artin groups (RAAGs)
  - finite extensions of subgroups of RAAGs (hence: virtually special groups)
  - Coxeter groups
  - fully residually free groups (independently shown by Macdonald 2010)
  - fundamental groups of hyperbolic 3-manifolds
  - word hyperbolic groups (Saul Schleimer's talk)

What's interesting about the compressed word problem?

Let H be a finitely generated subgroup of $\mathrm{Aut}(G)$.

$\mathrm{CWP}(G) \in \mathbf{P} \Rightarrow \mathrm{WP}(H) \in \mathbf{P}$

What's interesting about the compressed word problem?

Let H be a finitely generated subgroup of $\mathrm{Aut}(G)$.

$\mathrm{CWP}(G) \in \mathbf{P} \Rightarrow \mathrm{WP}(H) \in \mathbf{P}$

Let $G = K \rtimes Q$ be a semi-direct product.

$\mathrm{WP}(Q) \in \mathbf{P}$, $\mathrm{CWP}(K) \in \mathbf{P} \Rightarrow \mathrm{WP}(G) \in \mathbf{P}$

What's interesting about the compressed word problem?

Let H be a finitely generated subgroup of $\mathrm{Aut}(G)$.

$\mathrm{CWP}(G) \in \mathbf{P} \Rightarrow \mathrm{WP}(H) \in \mathbf{P}$

Let $G = K \rtimes Q$ be a semi-direct product.

$\mathrm{WP}(Q) \in \mathbf{P}$, $\mathrm{CWP}(K) \in \mathbf{P} \Rightarrow \mathrm{WP}(G) \in \mathbf{P}$

Let $1 \to K \to G \to Q \to 1$ be a short exact sequence of f.g. groups such that the quotient Q is finitely presented.

$\mathrm{WSP}(Q) \in \mathbf{P}$, $\mathrm{CWP}(K) \in \mathbf{P} \Rightarrow \mathrm{WP}(G) \in \mathbf{P}$

Wreath products

Let A and B be groups and let

$$K = \bigoplus_{b \in B} A$$

be the direct sum of copies of A.

Wreath products

Let A and B be groups and let

$$K = \bigoplus_{b \in B} A$$

be the direct sum of copies of A.

Elements of K can be thought as mappings $k : B \to A$ with finite support (i.e., $k^{-1}(A \setminus 1)$ is finite).

Wreath products

Let A and B be groups and let

$$K = \bigoplus_{b \in B} A$$

be the direct sum of copies of A.

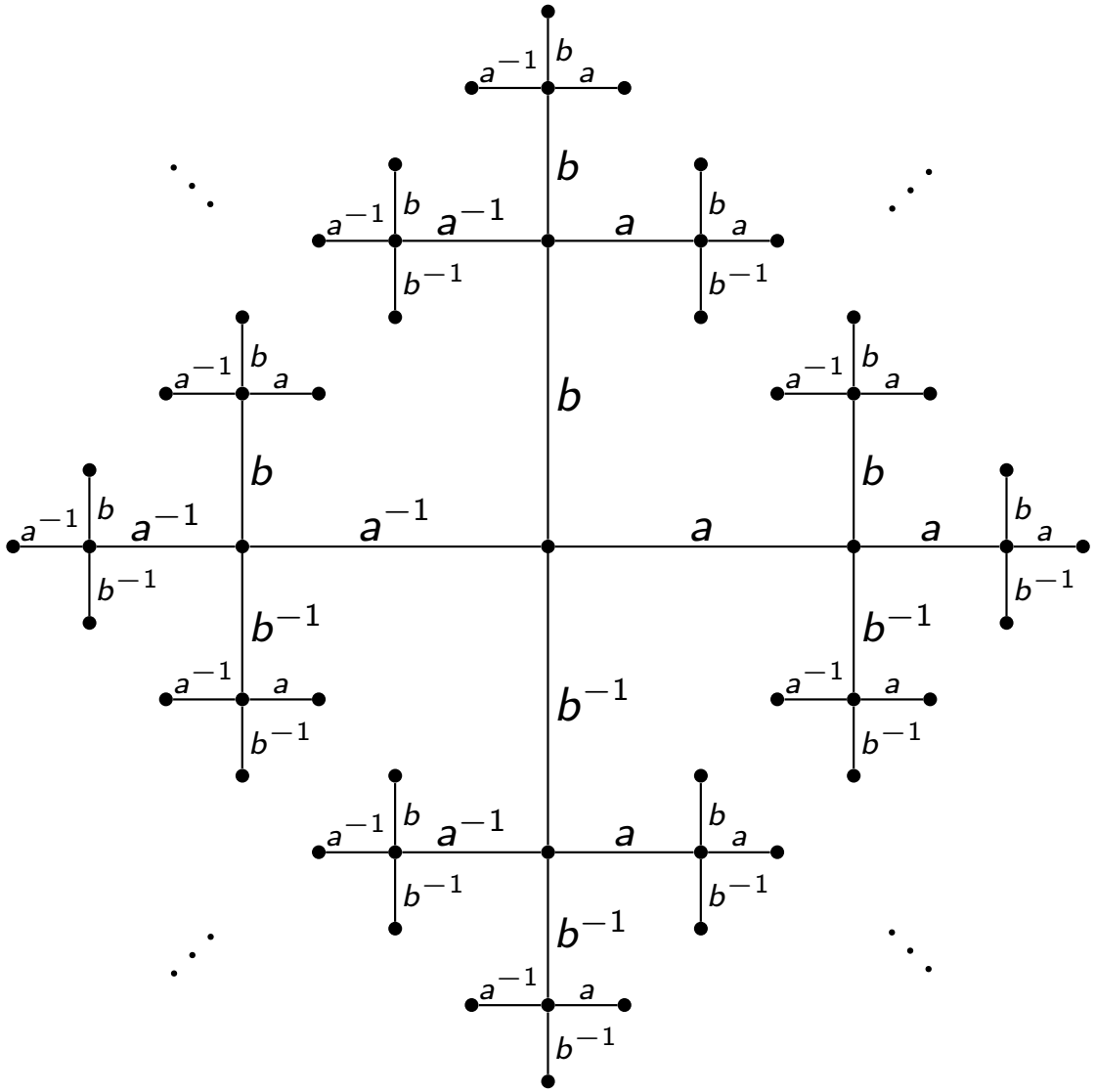
Elements of K can be thought as mappings $k : B \to A$ with finite support (i.e., $k^{-1}(A \setminus 1)$ is finite).

The wreath product $A \wr B$ is the set of all pairs $K \times B$ with the following multiplication, where $(k_1, b_1), (k_2, b_2) \in K \times B$:

$$(k_1, b_1)(k_2, b_2) = (k, b_1 b_2) \text{ with } \forall b \in B : k(b) = k_1(b) k_2(b_1^{-1} b).$$

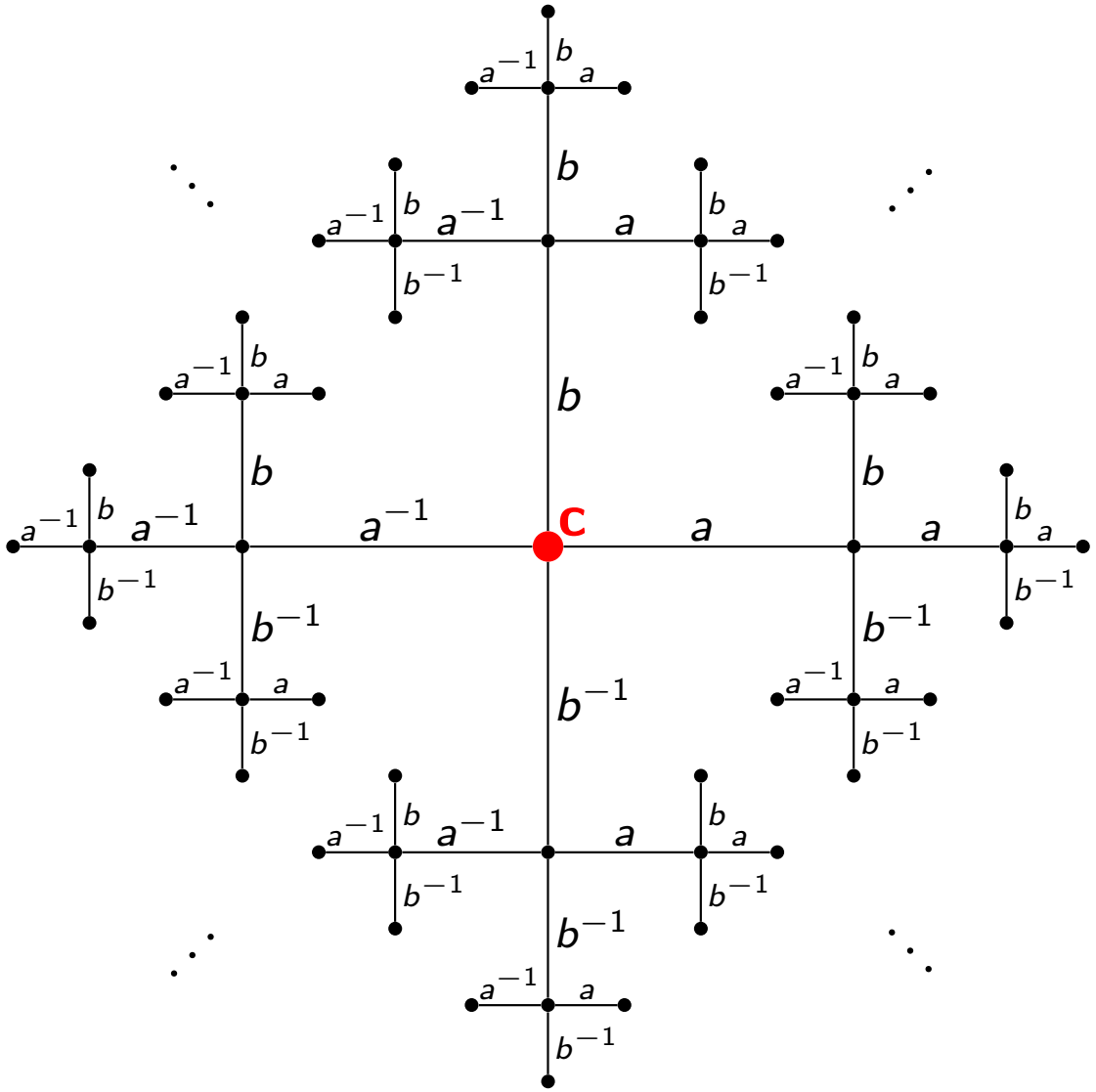
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



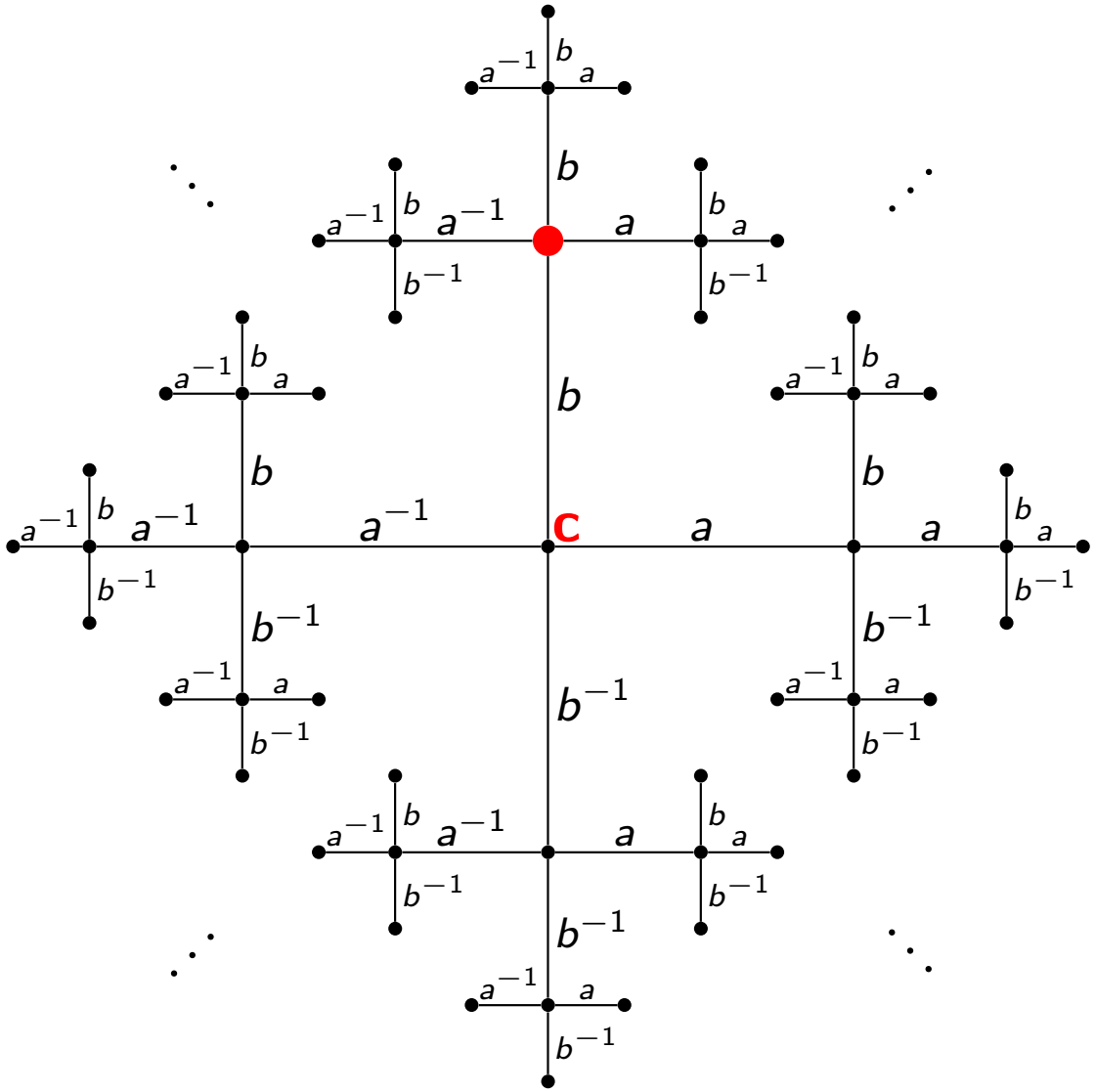
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



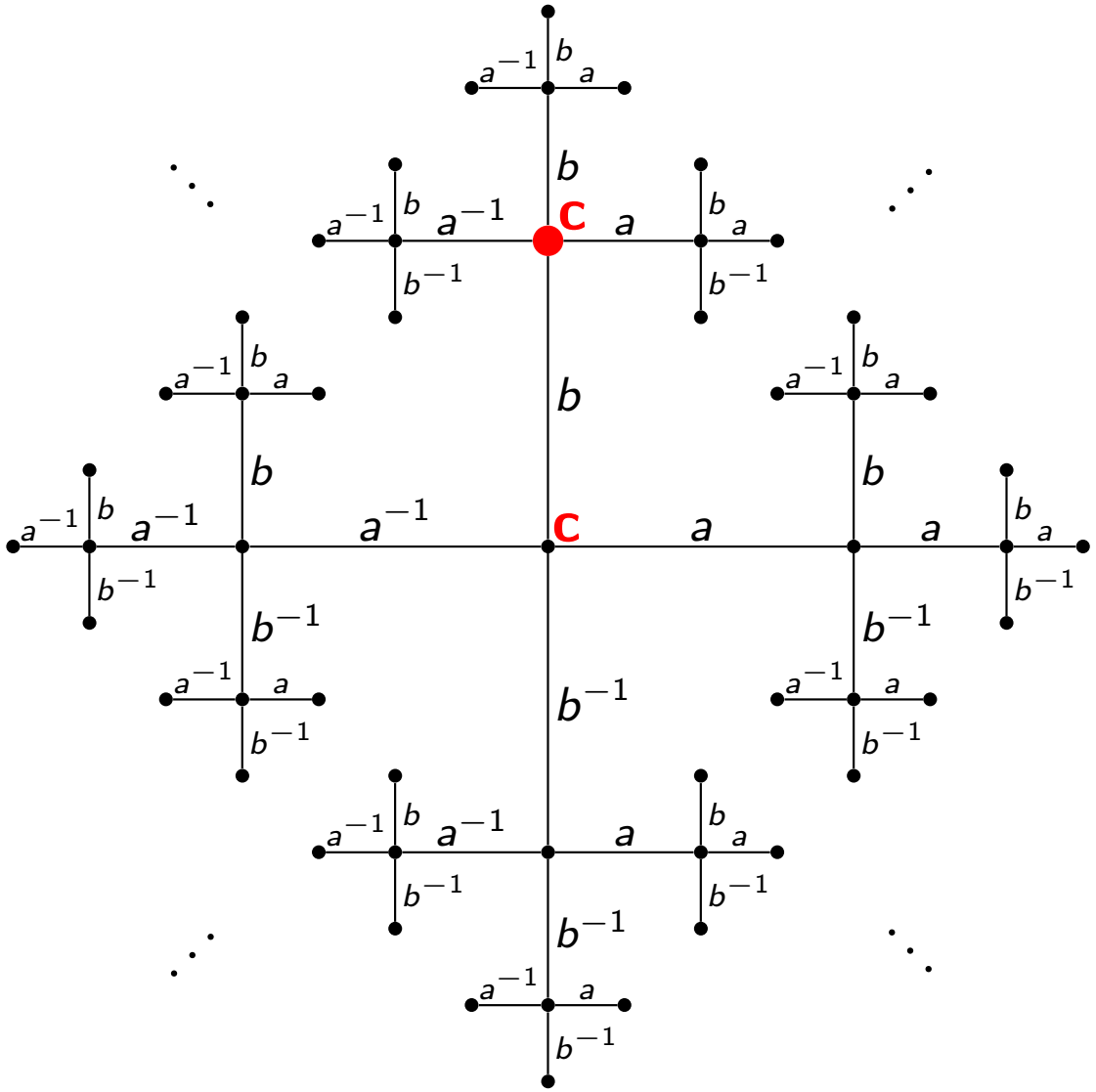
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



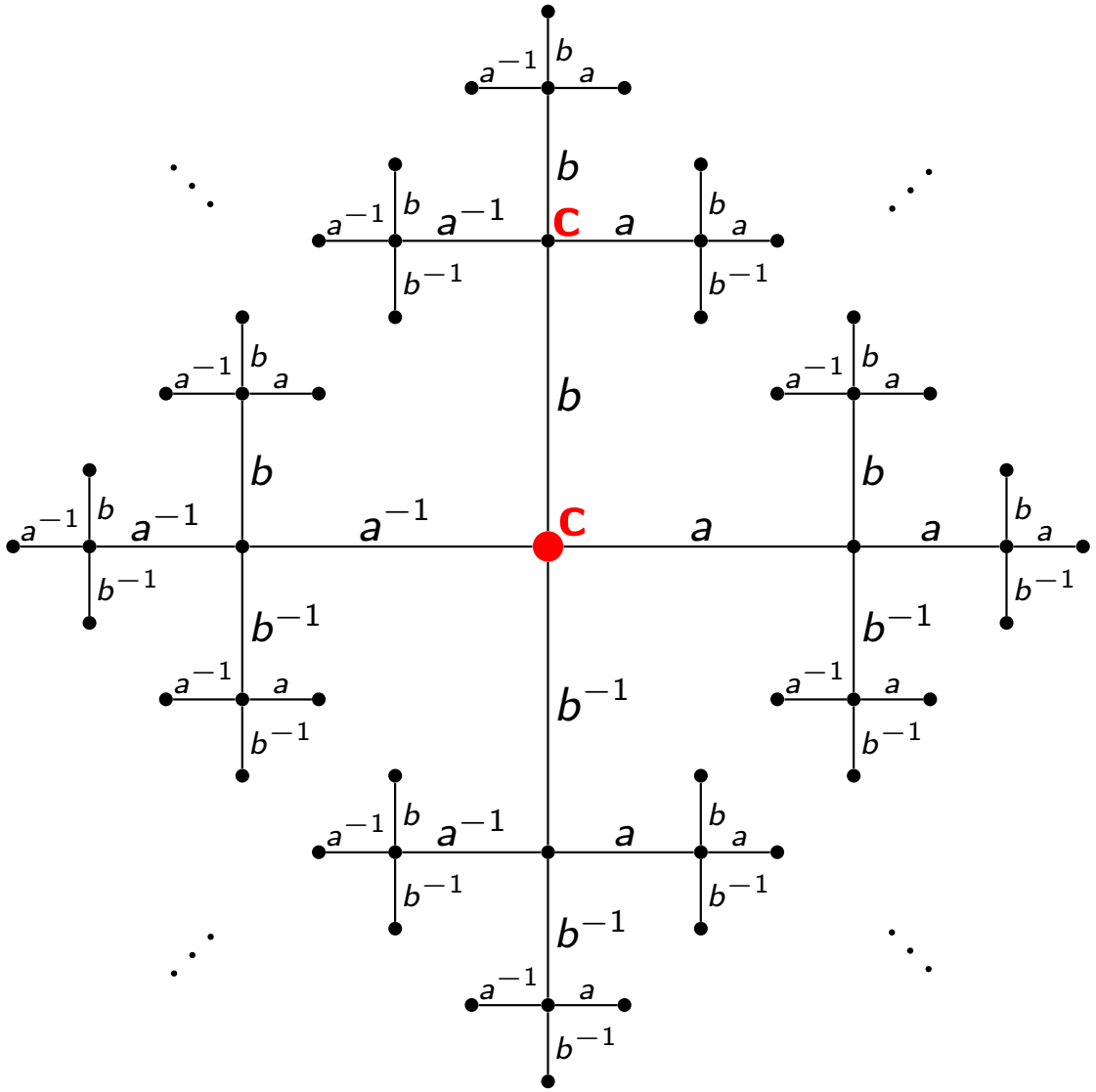
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



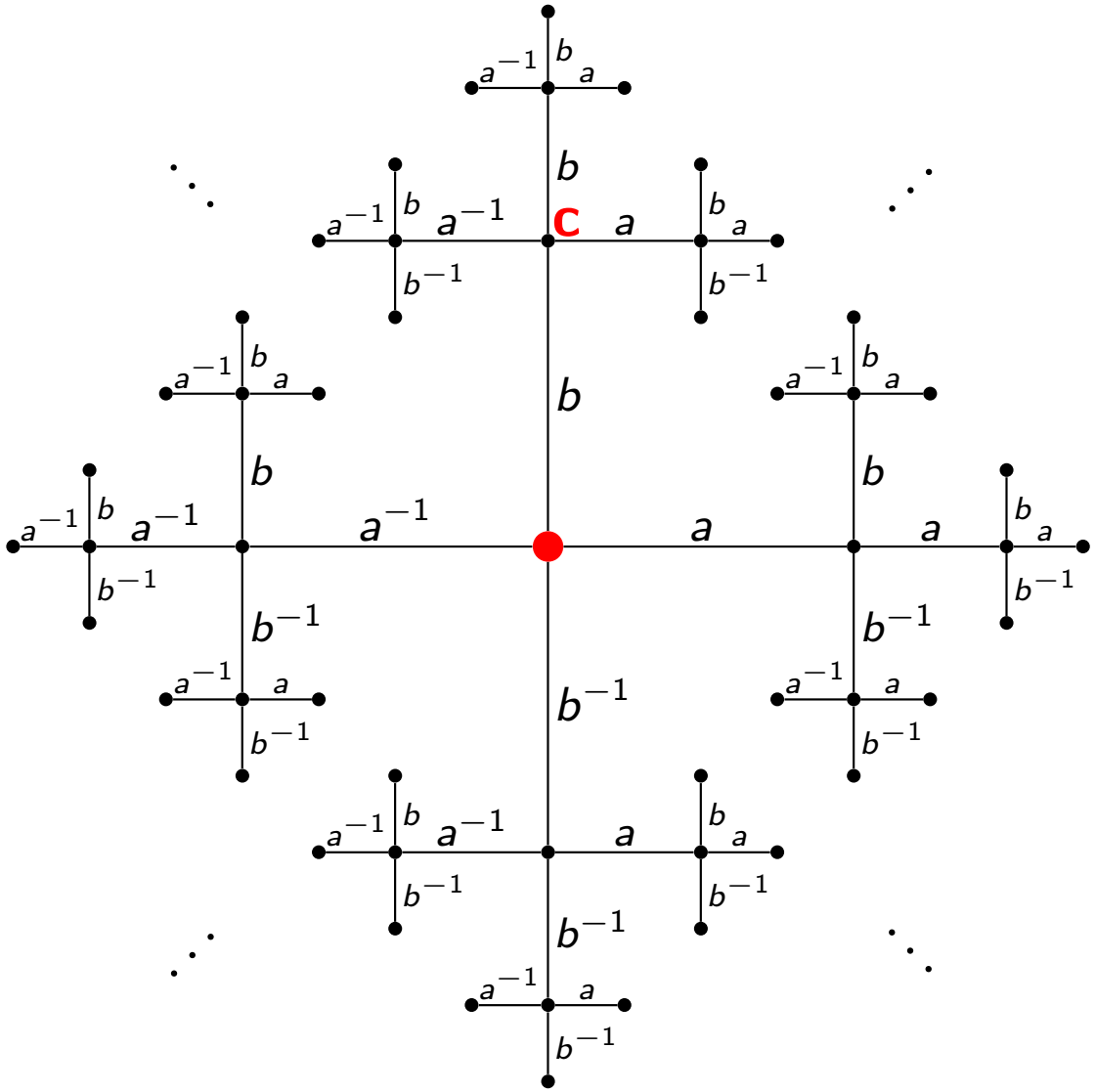
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



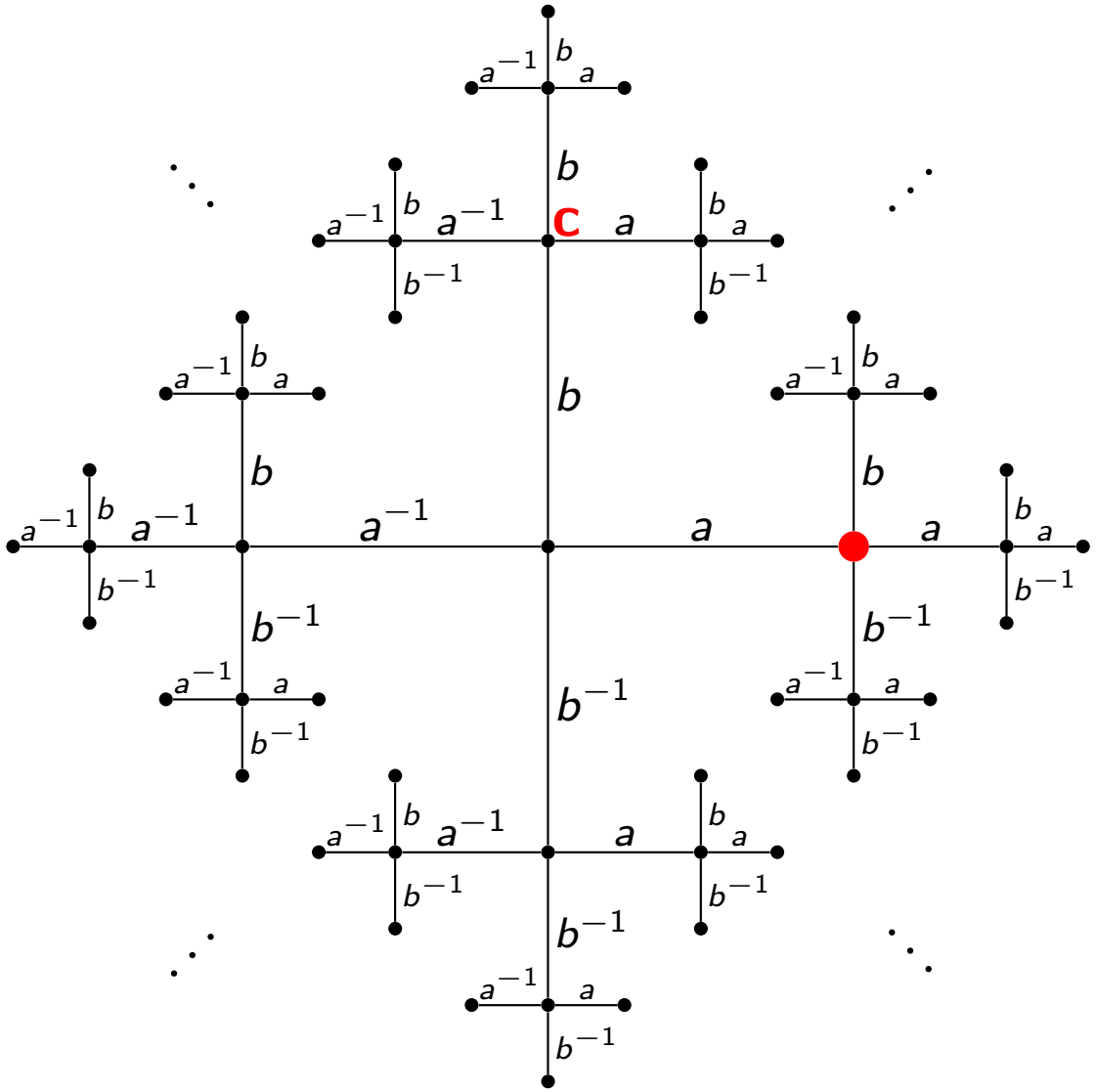
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



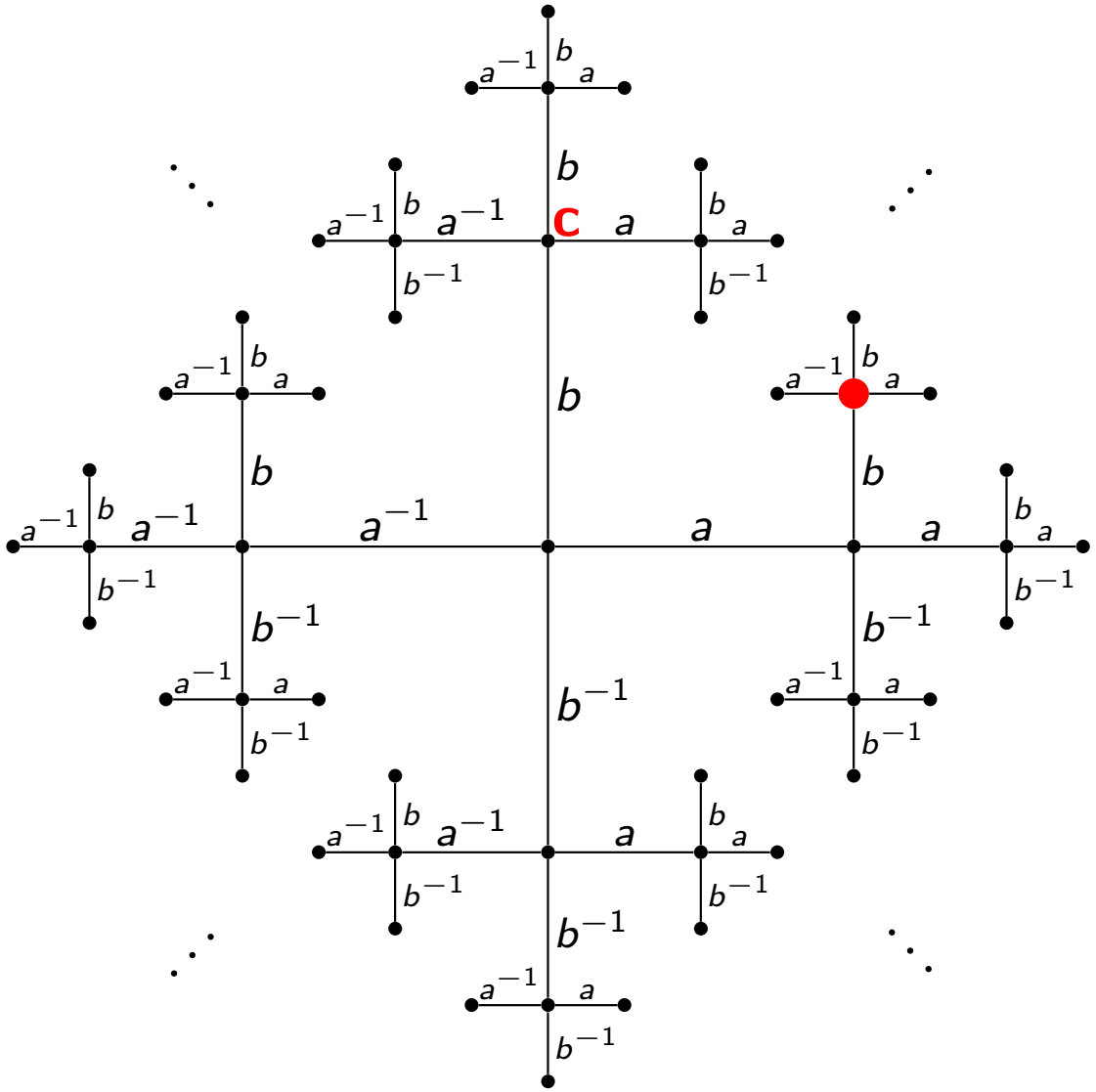
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



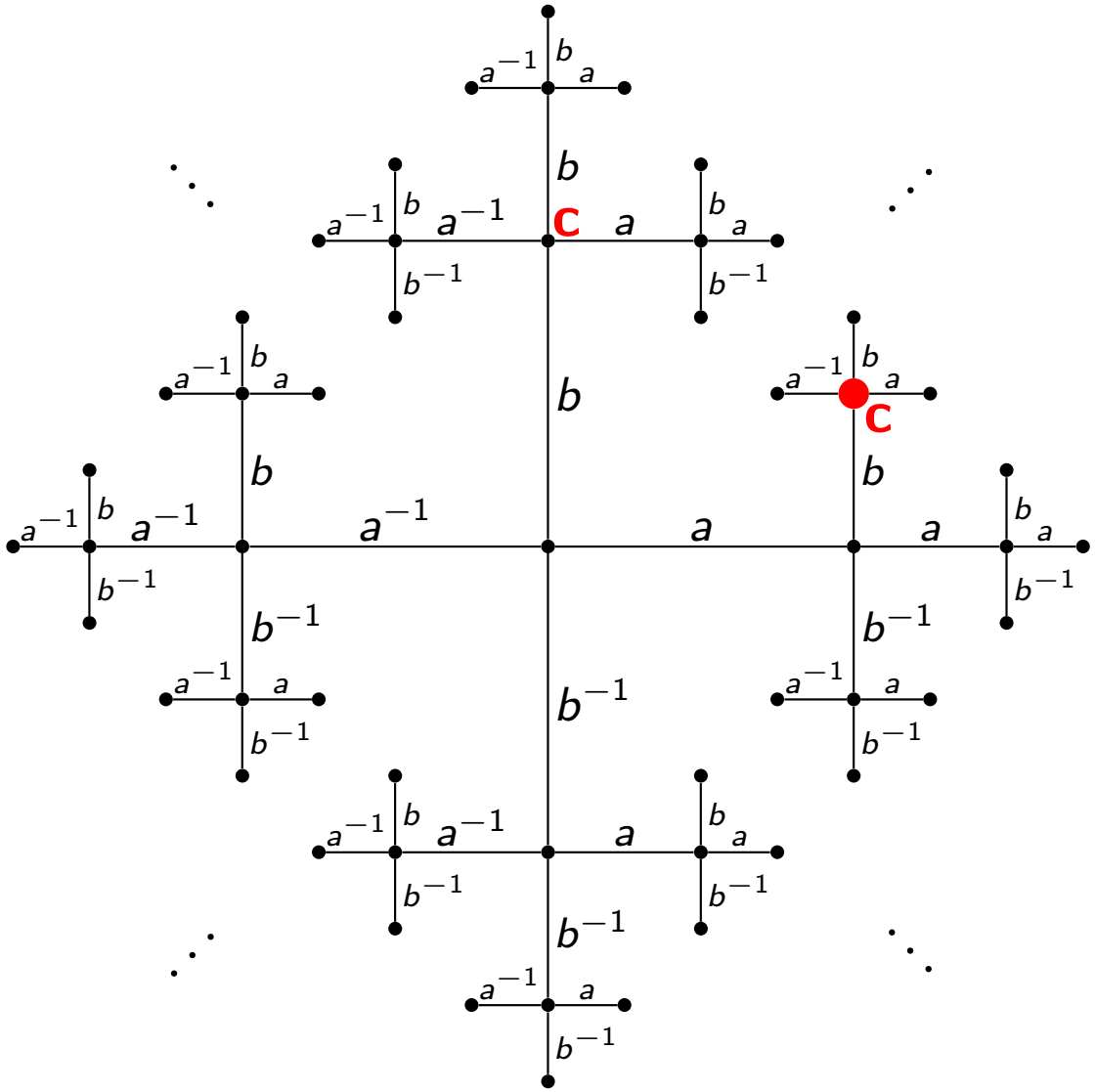
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



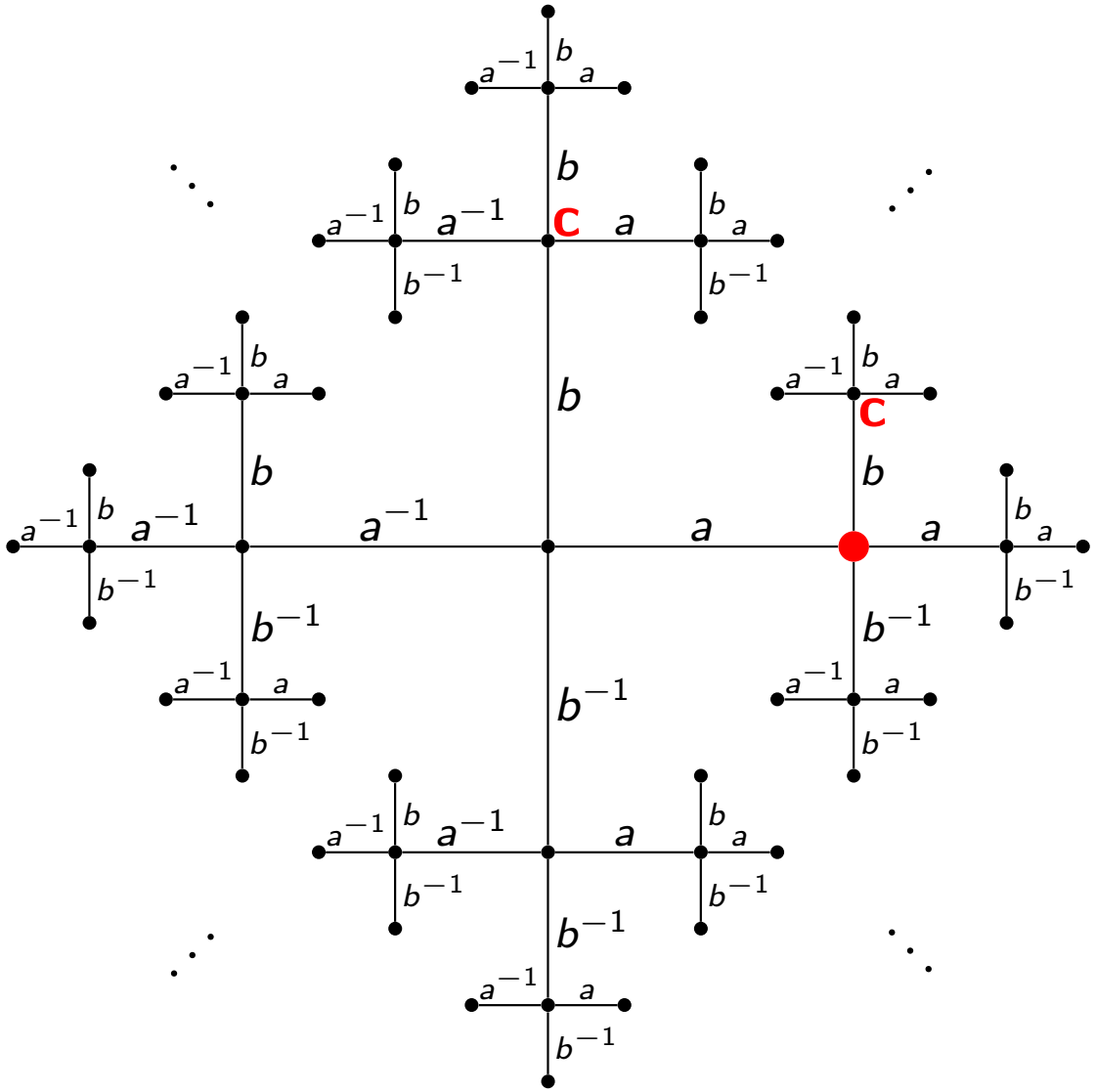
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



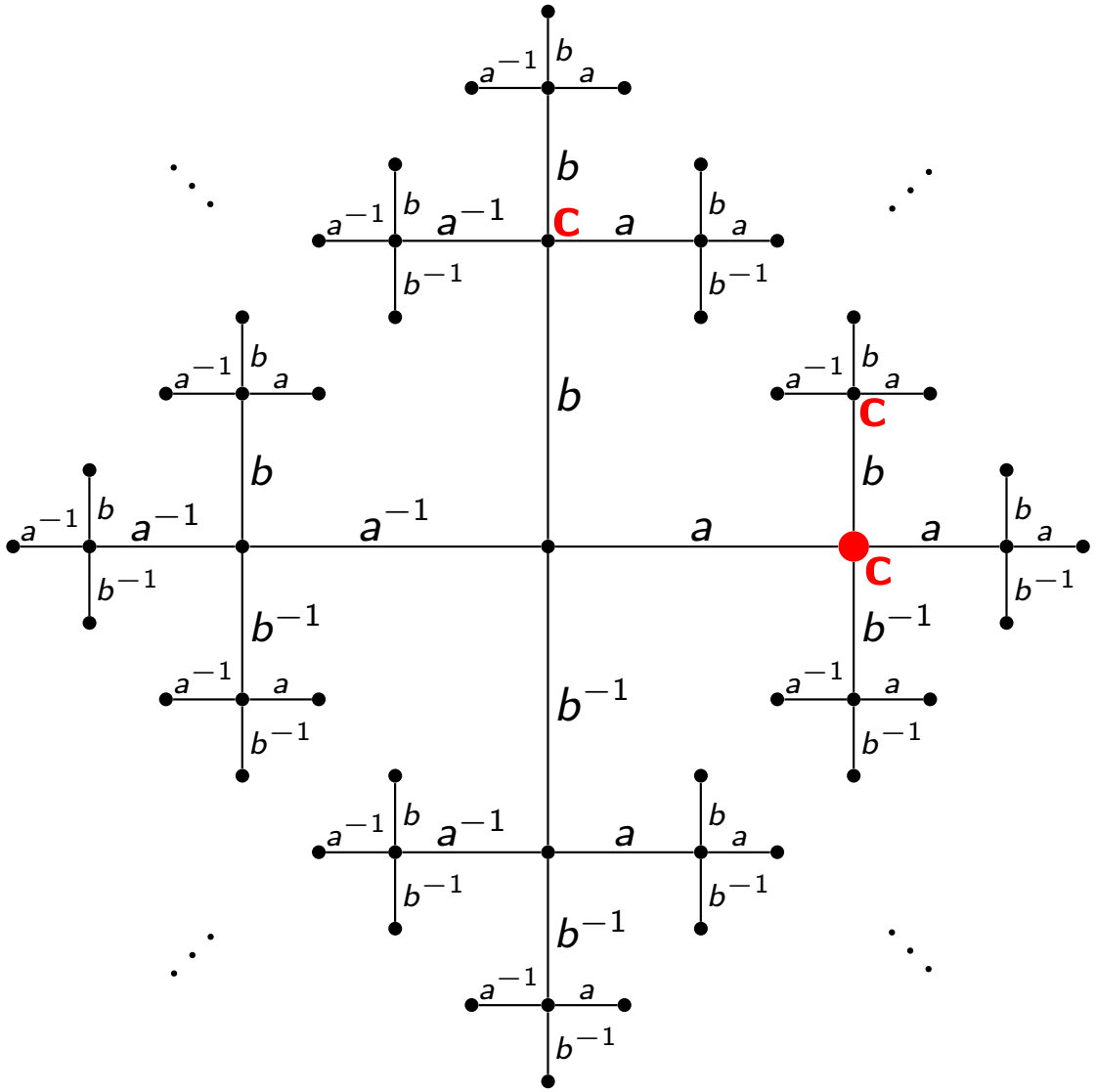
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



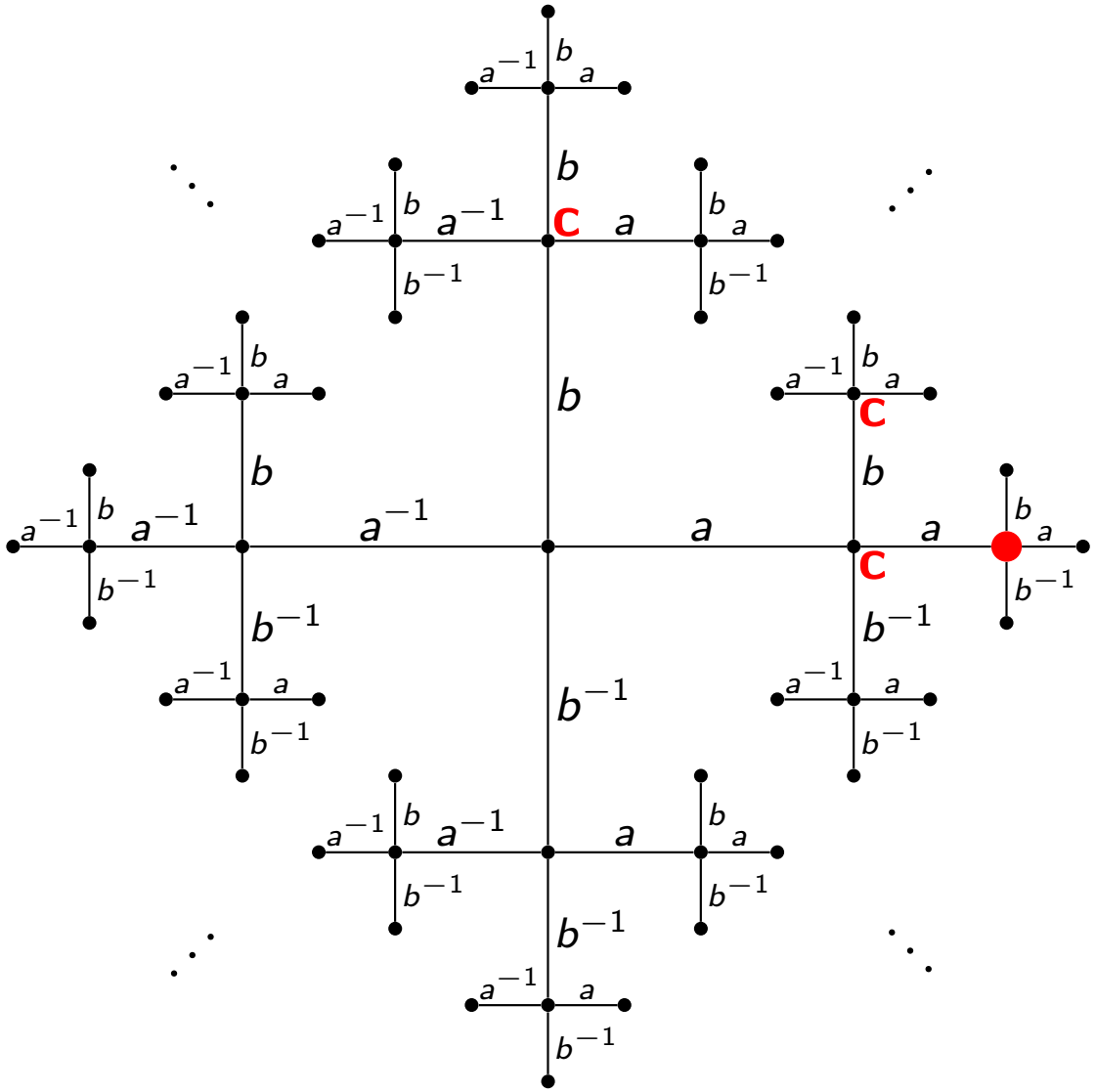
Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



Wreath product $\mathbb{Z}_2 \wr F(a,b)$ with $\mathbb{Z}_2 = \langle c \mid c^2 = 1 \rangle$

$cbcb^{-1}cabcb^{-1}ca$:



Easy word problem but difficult compressed word problem

Let A be any non-Abelian group. Then $\mathrm{CWP}(A \wr \mathbb{Z})$ is coNP-hard.

Easy word problem but difficult compressed word problem

Let A be any non-Abelian group. Then $\mathrm{CWP}(A \wr \mathbb{Z})$ is coNP-hard.

Remark: If A is finite then $\mathrm{WP}(A \wr \mathbb{Z})$ can be solved in logspace.

Easy word problem but difficult compressed word problem

Let A be any non-Abelian group. Then $\mathrm{CWP}(A \wr \mathbb{Z})$ is coNP-hard.

Remark: If A is finite then $\mathrm{WP}(A \wr \mathbb{Z})$ can be solved in logspace.

Proof sketch: Reduction from **coSUBSETSUM:**

Easy word problem but difficult compressed word problem

Let A be any non-Abelian group. Then $\mathrm{CWP}(A \wr \mathbb{Z})$ is coNP-hard.

Remark: If A is finite then $\mathrm{WP}(A \wr \mathbb{Z})$ can be solved in logspace.

Proof sketch: Reduction from **coSUBSETSUM:**

INPUT: Binary coded weight vector $\overline{w} \in \mathbb{N}^n$ and a target $z \in \mathbb{N}$.
QUESTION: Does for all $\overline{x} \in \{0,1\}^n$, $\overline{x} \cdot \overline{w} \neq z$ hold?

Easy word problem but difficult compressed word problem

Let A be any non-Abelian group. Then $\mathrm{CWP}(A \wr \mathbb{Z})$ is coNP-hard.

Remark: If A is finite then $\mathrm{WP}(A \wr \mathbb{Z})$ can be solved in logspace.

Proof sketch: Reduction from **coSUBSETSUM:**

INPUT: Binary coded weight vector $\overline{w} \in \mathbb{N}^n$ and a target $z \in \mathbb{N}$.
QUESTION: Does for all $\overline{x} \in \{0,1\}^n$, $\overline{x} \cdot \overline{w} \neq z$ hold?

Let $\overline{w} = (w_1, \ldots, w_n)$ and $s = w_1 + \cdots + w_n$.

Easy word problem but difficult compressed word problem

Let A be any non-Abelian group. Then $\mathrm{CWP}(A \wr \mathbb{Z})$ is coNP-hard.

Remark: If A is finite then $\mathrm{WP}(A \wr \mathbb{Z})$ can be solved in logspace.

Proof sketch: Reduction from **coSUBSETSUM:**

INPUT: Binary coded weight vector $\overline{w} \in \mathbb{N}^n$ and a target $z \in \mathbb{N}$.
QUESTION: Does for all $\overline{x} \in \{0,1\}^n$, $\overline{x} \cdot \overline{w} \neq z$ hold?

Let $\overline{w} = (w_1, \ldots, w_n)$ and $s = w_1 + \cdots + w_n$.

From $\overline{w}$, z we can construct in poly. time SLPs $\mathbb{A}$, $\mathbb{B}$ such that

$$\mathrm{val}(\mathbb{A}) = \prod_{\overline{x} \in \{0,1\}^n} (t^{\overline{x}\cdot\overline{w}-1}\, c\, t^{s-\overline{x}\cdot\overline{w}}) \quad \text{and} \quad \mathrm{val}(\mathbb{B}) = (t^{z-1}\, c\, t^{s-z})^{2^n}.$$

Easy word problem but difficult compressed word problem

Let A be any non-Abelian group. Then $\mathrm{CWP}(A \wr \mathbb{Z})$ is coNP-hard.

Remark: If A is finite then $\mathrm{WP}(A \wr \mathbb{Z})$ can be solved in logspace.

Proof sketch: Reduction from **coSUBSETSUM:**

INPUT: Binary coded weight vector $\overline{w} \in \mathbb{N}^n$ and a target $z \in \mathbb{N}$.
QUESTION: Does for all $\overline{x} \in \{0,1\}^n$, $\overline{x} \cdot \overline{w} \neq z$ hold?

Let $\overline{w} = (w_1, \ldots, w_n)$ and $s = w_1 + \cdots + w_n$.

From $\overline{w}$, z we can construct in poly. time SLPs $\mathbb{A}$, $\mathbb{B}$ such that

$$\mathrm{val}(\mathbb{A}) = \prod_{\overline{x} \in \{0,1\}^n} (t^{\overline{x}\cdot\overline{w}-1}\, c\, t^{s-\overline{x}\cdot\overline{w}}) \quad \text{and} \quad \mathrm{val}(\mathbb{B}) = (t^{z-1}\, c\, t^{s-z})^{2^n}.$$

$$\rightsquigarrow \quad \exists p \in \mathbb{N} : p\text{-th symbol of } \mathrm{val}(\mathbb{A}) = c = p\text{-th symbol of } \mathrm{val}(\mathbb{B})$$

$$\Leftrightarrow$$

$$\exists \overline{x} \in \{0,1\}^n : \overline{x} \cdot \overline{w} = z$$

Easy word problem but difficult compressed word problem

Let $\mathbb{Z} = \langle t \rangle$.

Easy word problem but difficult compressed word problem

Let $\mathbb{Z} = \langle t \rangle$.

Choose two elements $a, b \in A$ with $[a, b] \neq 1$.

Easy word problem but difficult compressed word problem

Let $\mathbb{Z} = \langle t \rangle$.

Choose two elements $a, b \in A$ with $[a, b] \neq 1$.

For $x \in \{a, b, a^{-1}, b^{-1}\}$ let $\mathbb{A}_x$ ($\mathbb{B}_x$) be the SLP that is obtained from $\mathbb{A}$ ($\mathbb{B}$) by replacing every occurrence of the letter c by x.

Easy word problem but difficult compressed word problem

Let $\mathbb{Z} = \langle t \rangle$.

Choose two elements $a, b \in A$ with $[a, b] \neq 1$.

For $x \in \{a, b, a^{-1}, b^{-1}\}$ let $\mathbb{A}_x$ ($\mathbb{B}_x$) be the SLP that is obtained from $\mathbb{A}$ ($\mathbb{B}$) by replacing every occurrence of the letter c by x.

We can construct in poly. time an SLP $\mathbb{C}$ such that

$$\mathrm{val}(\mathbb{C}) = \mathrm{val}(\mathbb{A}_a)t^{-s\cdot 2^n}\mathrm{val}(\mathbb{B}_b)t^{-s\cdot 2^n}\mathrm{val}(\mathbb{A}_{a^{-1}})t^{-s\cdot 2^n}\mathrm{val}(\mathbb{B}_{b^{-1}})t^{-s\cdot 2^n}.$$

Easy word problem but difficult compressed word problem

Let $\mathbb{Z} = \langle t \rangle$.

Choose two elements $a, b \in A$ with $[a, b] \neq 1$.

For $x \in \{a, b, a^{-1}, b^{-1}\}$ let $\mathbb{A}_x$ ($\mathbb{B}_x$) be the SLP that is obtained from $\mathbb{A}$ ($\mathbb{B}$) by replacing every occurrence of the letter c by x.

We can construct in poly. time an SLP $\mathbb{C}$ such that

$$\mathrm{val}(\mathbb{C}) = \mathrm{val}(\mathbb{A}_a)t^{-s\cdot 2^n}\mathrm{val}(\mathbb{B}_b)t^{-s\cdot 2^n}\mathrm{val}(\mathbb{A}_{a^{-1}})t^{-s\cdot 2^n}\mathrm{val}(\mathbb{B}_{b^{-1}})t^{-s\cdot 2^n}.$$

Then we have:

$$\mathrm{val}(\mathbb{C}) \neq 1 \text{ in } A \wr \mathbb{Z}$$

$$\Leftrightarrow$$

$$\exists p \in \mathbb{N} : p\text{-th symbol of } \mathrm{val}(\mathbb{A}) = c = p\text{-th symbol of } \mathrm{val}(\mathbb{B}).$$

Other wreath products

If G and H are finitely generated abelian, then $H \wr G$ is finitely generated metabelian (2-step solvable).

Other wreath products

If G and H are finitely generated abelian, then $H \wr G$ is finitely generated metabelian (2-step solvable).

Wehrfritz 1980

Every finitely generated metabelian group embedds into a direct product of finitely generated linear groups.

Other wreath products

If G and H are finitely generated abelian, then $H \wr G$ is finitely generated metabelian (2-step solvable).

Wehrfritz 1980

Every finitely generated metabelian group embedds into a direct product of finitely generated linear groups.

Hence, $\mathrm{CWP}(H \wr G)$ (with G and H finitely generated abelian) reduces to the CWP for finitely generated linear groups.

Randomized complexity classes

A language L belongs to the class **RP** (randomized polynomial time) if there exists a nondeterministic polynomial time bounded Turing machine M such that for every input x:

- If $x \notin L$ then Prob[M accepts x] $= 0$.
- If $x \in L$ then Prob[M accepts x] $\geq 1/2$.

Randomized complexity classes

A language L belongs to the class **RP** (randomized polynomial time) if there exists a nondeterministic polynomial time bounded Turing machine M such that for every input x:

- If $x \notin L$ then Prob[M accepts x] $= 0$.
- If $x \in L$ then Prob[M accepts x] $\geq 1/2$.

A language L belongs to the class **coRP** if there exists a nondeterministic polynomial time bounded Turing machine M such that for every input x:

- If $x \in L$ then Prob[M accepts x] $= 1$.
- If $x \notin L$ then Prob[M accepts x] $\leq 1/2$.

Randomized complexity classes

A language L belongs to the class **RP** (randomized polynomial time) if there exists a nondeterministic polynomial time bounded Turing machine M such that for every input x:

- If $x \notin L$ then Prob[M accepts x] $= 0$.
- If $x \in L$ then Prob[M accepts x] $\geq 1/2$.

A language L belongs to the class **coRP** if there exists a nondeterministic polynomial time bounded Turing machine M such that for every input x:

- If $x \in L$ then Prob[M accepts x] $= 1$.
- If $x \notin L$ then Prob[M accepts x] $\leq 1/2$.

Impagliazzo, Wigderson 1997

If there exists a language in $\mathbf{DTIME}(2^{\mathcal{O}(n)})$ that has circuit complexity $2^{\Omega(n)}$ (seems to be plausible) then $\mathbf{P} = \mathbf{RP} = \mathbf{coRP}$ (actually, $\mathbf{P} = \mathbf{BPP}$).

Polynomial identity testing

An arithmetic circuit is a directed acyclic graph $\mathcal{C}$ such that:

- Every node (gate) is labelled with either 1, -1, a variable $x_1, \ldots, x_n$, or an operator $+$, $\cdot$.
- Nodes labelled with 1, -1, or a variable x_i have no incoming edges.
- There is a distinguished gate o (the output gate).

Polynomial identity testing

An arithmetic circuit is a directed acyclic graph $\mathcal{C}$ such that:

- Every node (gate) is labelled with either 1, -1, a variable $x_1, \ldots, x_n$, or an operator $+$, $\cdot$.
- Nodes labelled with 1, -1, or a variable x_i have no incoming edges.
- There is a distinguished gate o (the output gate).

$\mathcal{C}$ defines a polynomial $p_{\mathcal{C}}(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$.

Polynomial identity testing

An arithmetic circuit is a directed acyclic graph $\mathcal{C}$ such that:

- Every node (gate) is labelled with either 1, -1, a variable $x_1, \ldots, x_n$, or an operator $+$, $\cdot$.
- Nodes labelled with 1, -1, or a variable x_i have no incoming edges.
- There is a distinguished gate o (the output gate).

$\mathcal{C}$ defines a polynomial $p_{\mathcal{C}}(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$.

An arithmetic circuit variable-free if there is no node labeled with a variable x_i (hence, $p_{\mathcal{C}} \in \mathbb{Z}$).

Polynomial identity testing

An arithmetic circuit is a directed acyclic graph $\mathcal{C}$ such that:

- Every node (gate) is labelled with either 1, -1, a variable $x_1, \ldots, x_n$, or an operator $+$, $\cdot$.
- Nodes labelled with 1, -1, or a variable x_i have no incoming edges.
- There is a distinguished gate o (the output gate).

$\mathcal{C}$ defines a polynomial $p_{\mathcal{C}}(x_1, \ldots, x_n) \in \mathbb{Z}[x_1, \ldots, x_n]$.

An arithmetic circuit variable-free if there is no node labeled with a variable x_i (hence, $p_{\mathcal{C}} \in \mathbb{Z}$).

Polynomial identity testing over the ring $R \in \{\mathbb{Z}\} \cup \{\mathbb{Z}_n \mid n \geq 2\}$

INPUT: An arithmetic circuit $\mathcal{C}$.
QUESTION: Is $p_{\mathcal{C}}$ the zero polynomial in $R[x_1, \ldots, x_n]$?

Complexity of polynomial identity testing

Ibarra, Moran 1983; Agrawal, Biswas 2003

For every ring $R \in \{\mathbb{Z}\} \cup \{\mathbb{Z}_n \mid n \geq 2\}$, polynomial identity testing over R belongs to **coRP**.

Complexity of polynomial identity testing

Ibarra, Moran 1983; Agrawal, Biswas 2003

For every ring $R \in \{\mathbb{Z}\} \cup \{\mathbb{Z}_n \mid n \geq 2\}$, polynomial identity testing over R belongs to **coRP**.

Allender, Bürgisser, Kjeldgaard-Pedersen, Miltersen 2008

Polynomial identity testing over $\mathbb{Z}$ is equivalent w.r.t. polynomial time many-one reductions) to polynomial identity testing over $\mathbb{Z}$, restricted to variable-free arithmetic circuits.

Complexity of polynomial identity testing

Ibarra, Moran 1983; Agrawal, Biswas 2003

For every ring $R \in \{\mathbb{Z}\} \cup \{\mathbb{Z}_n \mid n \geq 2\}$, polynomial identity testing over R belongs to **coRP**.

Allender, Bürgisser, Kjeldgaard-Pedersen, Miltersen 2008

Polynomial identity testing over $\mathbb{Z}$ is equivalent w.r.t. polynomial time many-one reductions) to polynomial identity testing over $\mathbb{Z}$, restricted to variable-free arithmetic circuits.

Kabanets, Impagliazzo 2004

If polynomial identity testing over $\mathbb{Z}$ belongs to **P**, then one of the following conclusions holds:

- There is a language in **NEXPTIME** that does not have polynomial size boolean circuits.
- The permanent is not computable by polynomial size arithmetic circuits.

Polynomial identity testing and the compressed word problem

If G is finitely generated linear over field of characteristic 0 (resp. $p \in$ Primes), then $\mathrm{CWP}(G)$ can be reduced to polynomial identity testing over $\mathbb{Z}$ (resp. $\mathbb{Z}_p$).

Polynomial identity testing and the compressed word problem

If G is finitely generated linear over field of characteristic 0 (resp. $p \in$ Primes), then $\mathrm{CWP}(G)$ can be reduced to polynomial identity testing over $\mathbb{Z}$ (resp. $\mathbb{Z}_p$).

In particular, $\mathrm{CWP}(G)$ belongs to **coRP**.

Proof: G can be embedded into $\mathrm{GL}_n(\mathbb{Q}(x_1, \ldots, x_n))$ (resp. $\mathrm{GL}_n(\mathbb{F}_p(x_1, \ldots, x_n))$ for some n (Lipton, Zalcstein 1975).

Polynomial identity testing and the compressed word problem

If G is finitely generated linear over field of characteristic 0 (resp. $p \in$ Primes), then $\mathrm{CWP}(G)$ can be reduced to polynomial identity testing over $\mathbb{Z}$ (resp. $\mathbb{Z}_p$).

In particular, $\mathrm{CWP}(G)$ belongs to **coRP**.

Proof: G can be embedded into $\mathrm{GL}_n(\mathbb{Q}(x_1, \ldots, x_n))$ (resp. $\mathrm{GL}_n(\mathbb{F}_p(x_1, \ldots, x_n))$ for some n (Lipton, Zalcstein 1975).

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

Polynomial identity testing and the compressed word problem

If G is finitely generated linear over field of characteristic 0 (resp. $p \in$ Primes), then $\mathrm{CWP}(G)$ can be reduced to polynomial identity testing over $\mathbb{Z}$ (resp. $\mathbb{Z}_p$).

In particular, $\mathrm{CWP}(G)$ belongs to **coRP**.

Proof: G can be embedded into $\mathrm{GL}_n(\mathbb{Q}(x_1, \ldots, x_n))$ (resp. $\mathrm{GL}_n(\mathbb{F}_p(x_1, \ldots, x_n))$ for some n (Lipton, Zalcstein 1975).

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

Proof: Uses a construction of Ben-Or, Cleve 1992.

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

Proof: Let $\mathcal{C}$ be a variable-free arithmetic circuit $\mathcal{C}$ over $\mathbb{Z}$.

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

Proof: Let $\mathcal{C}$ be a variable-free arithmetic circuit $\mathcal{C}$ over $\mathbb{Z}$.

Construct an SLP $\mathbb{A}$ over generators of $\mathrm{SL}_3(\mathbb{Z})$ such that:
$p_{\mathcal{C}} = 0 \Leftrightarrow \mathrm{val}(\mathbb{A}) = I_3$.

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

Proof: Let $\mathcal{C}$ be a variable-free arithmetic circuit $\mathcal{C}$ over $\mathbb{Z}$.

Construct an SLP $\mathbb{A}$ over generators of $\mathrm{SL}_3(\mathbb{Z})$ such that: $p_{\mathcal{C}} = 0 \Leftrightarrow \mathrm{val}(\mathbb{A}) = I_3$.

The SLP $\mathbb{A}$ contains for every $\mathcal{C}$-gate A and all $b \in \{-1, 1\}$ and $1 \leq i, j \leq 3$ with $i \neq j$ a variable $A_{i,j,b}$ such that: If $\overline{y} = A_{i,j,b} \cdot \overline{x}$ then $y_i = x_i + b \cdot A \cdot x_j$ and $y_k = x_k$ for $k \in \{1, 2, 3\} \setminus \{j\}$.

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

Proof: Let $\mathcal{C}$ be a variable-free arithmetic circuit $\mathcal{C}$ over $\mathbb{Z}$.

Construct an SLP $\mathbb{A}$ over generators of $\mathrm{SL}_3(\mathbb{Z})$ such that: $p_{\mathcal{C}} = 0 \Leftrightarrow \mathrm{val}(\mathbb{A}) = I_3$.

The SLP $\mathbb{A}$ contains for every $\mathcal{C}$-gate A and all $b \in \{-1, 1\}$ and $1 \leq i, j \leq 3$ with $i \neq j$ a variable $A_{i,j,b}$ such that: If $\overline{y} = A_{i,j,b} \cdot \overline{x}$ then $y_i = x_i + b \cdot A \cdot x_j$ and $y_k = x_k$ for $k \in \{1, 2, 3\} \setminus \{j\}$.

Consider a $\mathcal{C}$-gate A.

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

Proof: Let $\mathcal{C}$ be a variable-free arithmetic circuit $\mathcal{C}$ over $\mathbb{Z}$.

Construct an SLP $\mathbb{A}$ over generators of $\mathrm{SL}_3(\mathbb{Z})$ such that: $p_{\mathcal{C}} = 0 \Leftrightarrow \mathrm{val}(\mathbb{A}) = I_3$.

The SLP $\mathbb{A}$ contains for every $\mathcal{C}$-gate A and all $b \in \{-1, 1\}$ and $1 \leq i, j \leq 3$ with $i \neq j$ a variable $A_{i,j,b}$ such that: If $\overline{y} = A_{i,j,b} \cdot \overline{x}$ then $y_i = x_i + b \cdot A \cdot x_j$ and $y_k = x_k$ for $k \in \{1, 2, 3\} \setminus \{j\}$.

Consider a $\mathcal{C}$-gate A.

Case 1. $A := c \in \{-1, 1\}$. Set for instance

$$A_{1,2,1} := \begin{pmatrix} 1 & c & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$\mathrm{CWP}(\mathrm{SL}_3(\mathbb{Z}))$ is equivalent w.r.t. polynomial time many-one reductions to polynomial identity testing over $\mathbb{Z}$.

Proof: Let $\mathcal{C}$ be a variable-free arithmetic circuit $\mathcal{C}$ over $\mathbb{Z}$.

Construct an SLP $\mathbb{A}$ over generators of $\mathrm{SL}_3(\mathbb{Z})$ such that: $p_{\mathcal{C}} = 0 \Leftrightarrow \mathrm{val}(\mathbb{A}) = I_3$.

The SLP $\mathbb{A}$ contains for every $\mathcal{C}$-gate A and all $b \in \{-1, 1\}$ and $1 \leq i, j \leq 3$ with $i \neq j$ a variable $A_{i,j,b}$ such that: If $\overline{y} = A_{i,j,b} \cdot \overline{x}$ then $y_i = x_i + b \cdot A \cdot x_j$ and $y_k = x_k$ for $k \in \{1, 2, 3\} \setminus \{j\}$.

Consider a $\mathcal{C}$-gate A.

Case 1. $A := c \in \{-1, 1\}$. Set for instance

$$A_{1,2,1} := \begin{pmatrix} 1 & c & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Case 2. $A := B + C$. Set $A_{i,j,b} := B_{i,j,b} + C_{i,j,b}$.

Case 3. $A := B \cdot C$. Let $\{k\} = \{1,2,3\} \setminus \{i,j\}$. Then we set

$$\begin{aligned} A_{i,j,1} &:= B_{k,j,-1} C_{i,k,1} B_{k,j,1} C_{i,k,-1} \\ A_{i,j,-1} &:= B_{k,j,-1} C_{i,k,-1} B_{k,j,1} C_{i,k,1} \end{aligned}$$

Case 3. $A := B \cdot C$. Let $\{k\} = \{1,2,3\} \setminus \{i,j\}$. Then we set

$$\begin{aligned} A_{i,j,1} &:= B_{k,j,-1} C_{i,k,1} B_{k,j,1} C_{i,k,-1} \\ A_{i,j,-1} &:= B_{k,j,-1} C_{i,k,-1} B_{k,j,1} C_{i,k,1} \end{aligned}$$

If $\overline{y} = A_{i,j,1} \cdot \overline{x}$, then $y_j = x_j$, $y_k = x_k + B \cdot x_j - B \cdot x_j = x_k$, and

$$y_i = x_i - C \cdot x_k + C \cdot (x_k + B \cdot x_j) = x_i + C \cdot B \cdot x_j.$$

Case 3. $A := B \cdot C$. Let $\{k\} = \{1,2,3\} \setminus \{i,j\}$. Then we set

$$\begin{aligned} A_{i,j,1} &:= B_{k,j,-1} C_{i,k,1} B_{k,j,1} C_{i,k,-1} \\ A_{i,j,-1} &:= B_{k,j,-1} C_{i,k,-1} B_{k,j,1} C_{i,k,1} \end{aligned}$$

If $\overline{y} = A_{i,j,1} \cdot \overline{x}$, then $y_j = x_j$, $y_k = x_k + B \cdot x_j - B \cdot x_j = x_k$, and

$$y_i = x_i - C \cdot x_k + C \cdot (x_k + B \cdot x_j) = x_i + C \cdot B \cdot x_j.$$

If $\overline{y} = A_{i,j,-1}\overline{x}$, then $y_j = x_j$, $y_k = x_k + B \cdot x_j - B \cdot x_j = x_k$, and

$$y_i = x_i + C \cdot x_k - C \cdot (x_k + B \cdot x_j) = x_i - C \cdot B \cdot x_j.$$

Case 3. $A := B \cdot C$. Let $\{k\} = \{1,2,3\} \setminus \{i,j\}$. Then we set

$$\begin{aligned} A_{i,j,1} &:= B_{k,j,-1} C_{i,k,1} B_{k,j,1} C_{i,k,-1} \\ A_{i,j,-1} &:= B_{k,j,-1} C_{i,k,-1} B_{k,j,1} C_{i,k,1} \end{aligned}$$

If $\overline{y} = A_{i,j,1} \cdot \overline{x}$, then $y_j = x_j$, $y_k = x_k + B \cdot x_j - B \cdot x_j = x_k$, and

$$y_i = x_i - C \cdot x_k + C \cdot (x_k + B \cdot x_j) = x_i + C \cdot B \cdot x_j.$$

If $\overline{y} = A_{i,j,-1}\overline{x}$, then $y_j = x_j$, $y_k = x_k + B \cdot x_j - B \cdot x_j = x_k$, and

$$y_i = x_i + C \cdot x_k - C \cdot (x_k + B \cdot x_j) = x_i - C \cdot B \cdot x_j.$$

Let $S_{1,2,1}$ be the start variable of $\mathbb{A}$. $\rightsquigarrow$

Case 3. $A := B \cdot C$. Let $\{k\} = \{1,2,3\} \setminus \{i,j\}$. Then we set

$$\begin{aligned} A_{i,j,1} &:= B_{k,j,-1} C_{i,k,1} B_{k,j,1} C_{i,k,-1} \\ A_{i,j,-1} &:= B_{k,j,-1} C_{i,k,-1} B_{k,j,1} C_{i,k,1} \end{aligned}$$

If $\overline{y} = A_{i,j,1} \cdot \overline{x}$, then $y_j = x_j$, $y_k = x_k + B \cdot x_j - B \cdot x_j = x_k$, and

$$y_i = x_i - C \cdot x_k + C \cdot (x_k + B \cdot x_j) = x_i + C \cdot B \cdot x_j.$$

If $\overline{y} = A_{i,j,-1}\overline{x}$, then $y_j = x_j$, $y_k = x_k + B \cdot x_j - B \cdot x_j = x_k$, and

$$y_i = x_i + C \cdot x_k - C \cdot (x_k + B \cdot x_j) = x_i - C \cdot B \cdot x_j.$$

Let $S_{1,2,1}$ be the start variable of $\mathbb{A}$. $\rightsquigarrow$

$$p_{\mathcal{C}} = 0 \;\Leftrightarrow\; \forall \overline{x} \in \mathbb{Z}^3 : \mathrm{val}(\mathbb{A}) \cdot \overline{x} = \overline{x} \;\Leftrightarrow\; \mathrm{val}(\mathbb{A}) = I_3.$$

Open problems

- What is the precise complexity of $\mathrm{CWP}(A \wr \mathbb{Z})$ for A finite non-Abelian (coNP-hard, in PSPACE).

Open problems

- What is the precise complexity of $\mathrm{CWP}(A \wr \mathbb{Z})$ for A finite non-Abelian (coNP-hard, in PSPACE).
- Compressed word problem for $A \wr F_2$.

  Might be related to polynomial identity testing for non-commuting variables.

Open problems

- What is the precise complexity of $\mathrm{CWP}(A \wr \mathbb{Z})$ for A finite non-Abelian (coNP-hard, in PSPACE).
- Compressed word problem for $A \wr F_2$.

  Might be related to polynomial identity testing for non-commuting variables.
- Compressed word problem for braid groups (they are linear).