The power word problem in free groups

Armin Weiß[1]

Universität Stuttgart, FMI

Schloss Dagstuhl, March, 2019

[1] Joint work with Markus Lohrey

Dehn's fundamental problems and others

Let G be a f. g. group, generated by a finite set $\Sigma = \Sigma^{-1} \subseteq G$.

- Word problem (WP): Given $w \in \Sigma^*$. Question: Is $w = 1$ in G?
- Conjugacy problem: Given $v, w \in \Sigma^*$.
  Question: $\exists\, z \in G$ such that $zvz^{-1} = w$?

Dehn's fundamental problems and others

Let G be a f. g. group, generated by a finite set $\Sigma = \Sigma^{-1} \subseteq G$.

- Word problem (WP): Given $w \in \Sigma^*$. Question: Is $w = 1$ in G?
- Conjugacy problem: Given $v, w \in \Sigma^*$.
  Question: $\exists\, z \in G$ such that $zvz^{-1} = w$?
- Compressed word problem: Given a straight-line program $\mathbb{G}$ which produces a word $w \in \Sigma^*$.
  Question: Is $w = 1$ in G?

Dehn's fundamental problems and others

Let G be a f. g. group, generated by a finite set $\Sigma = \Sigma^{-1} \subseteq G$.

- Word problem (WP): Given $w \in \Sigma^*$. Question: Is $w = 1$ in G?
- Conjugacy problem: Given $v, w \in \Sigma^*$.
  Question: $\exists\, z \in G$ such that $zvz^{-1} = w$?
- Compressed word problem: Given a straight-line program $\mathbb{G}$ which produces a word $w \in \Sigma^*$.
  Question: Is $w = 1$ in G?
- Knapsack problem: Given $p_1, \ldots, p_k, w \in \Sigma^*$.
  Question: $\exists\, x_1, \ldots, x_k \in \mathbb{N}$ such that $p_1^{x_1} \cdots p_k^{x_k} = w$?
- ...

Let G be a f. g. group, generated by a finite set $\Sigma = \Sigma^{-1} \subseteq G$.

- Word problem (WP): Given $w \in \Sigma^*$. Question: Is $w = 1$ in G?
- Conjugacy problem: Given $v, w \in \Sigma^*$.
  Question: $\exists\, z \in G$ such that $zvz^{-1} = w$?
- Compressed word problem: Given a straight-line program $\mathbb{G}$ which produces a word $w \in \Sigma^*$.
  Question: Is $w = 1$ in G?
- Knapsack problem: Given $p_1, \ldots, p_k, w \in \Sigma^*$.
  Question: $\exists\, x_1, \ldots, x_k \in \mathbb{N}$ such that $p_1^{x_1} \cdots p_k^{x_k} = w$?
- ...
- Power word problem (PowerWP):
  Given $p_1, \ldots, p_k \in \Sigma^*$ and $x_1, \ldots, x_k \in \mathbb{Z}$.
  Question: $p_1^{x_1} \cdots p_k^{x_k} = 1$ in G?

Why is the power word problem interesting?

The power word problem is natural:

- straightforward way of compression

Why is the power word problem interesting?

The power word problem is natural:

- straightforward way of compression
- for abelian groups this is the usual way of encoding

Why is the power word problem interesting?

The power word problem is natural:

- straightforward way of compression
- for abelian groups this is the usual way of encoding
- in nilpotent groups, every element can be expressed by a power word of logarithmic length

Why is the power word problem interesting?

The power word problem is natural:

- straightforward way of compression
- for abelian groups this is the usual way of encoding
- in nilpotent groups, every element can be expressed by a power word of logarithmic length
- binary encoded matrices in $\mathrm{SL}(2, \mathbb{Z})$ yield power words over the generators (Gurevich, Schupp 07)

Why is the power word problem interesting?

The power word problem is natural:

- straightforward way of compression
- for abelian groups this is the usual way of encoding
- in nilpotent groups, every element can be expressed by a power word of logarithmic length
- binary encoded matrices in $\mathrm{SL}(2, \mathbb{Z})$ yield power words over the generators (Gurevich, Schupp 07)

$$\begin{pmatrix} -499 & 5000 \\ -50 & 501 \end{pmatrix}$$

Why is the power word problem interesting?

The power word problem is natural:

- straightforward way of compression
- for abelian groups this is the usual way of encoding
- in nilpotent groups, every element can be expressed by a power word of logarithmic length
- binary encoded matrices in $\mathrm{SL}(2,\mathbb{Z})$ yield power words over the generators (Gurevich, Schupp 07)

$$\begin{pmatrix} -499 & 5000 \\ -50 & 501 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{10} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{50} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-10}$$

Why is the power word problem interesting?

The power word problem is natural:

- straightforward way of compression
- for abelian groups this is the usual way of encoding
- in nilpotent groups, every element can be expressed by a power word of logarithmic length
- binary encoded matrices in $\mathrm{SL}(2, \mathbb{Z})$ yield power words over the generators (Gurevich, Schupp 07)

$$\begin{pmatrix} -499 & 5000 \\ -50 & 501 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{10} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{50} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}^{-10}$$

The power word problem helps

- to solve the knapsack problem in RAAGS (Lohrey, Zetsche, 15), ...
- to understand the compressed word problem better:
  - lower bounds
  - better upper bounds in the special case.

Why parallel complexity?

- Finer classification of problems inside polynomial time.

Why parallel complexity?

- Finer classification of problems inside polynomial time.
- We cannot be faster than linear time on one processor, but we can on many processors.

Why parallel complexity?

- Finer classification of problems inside polynomial time.
- We cannot be faster than linear time on one processor, but we can on many processors.
- Parallel computing is more and more important in the "real world".

Machine models:

- PRAMs (parallel random access machines)
- (Boolean) circuits

Parallel Complexity

Machine models:

- PRAMs (parallel random access machines)
- (Boolean) circuits

Circuit = directed acyclic graph where each vertex is either:

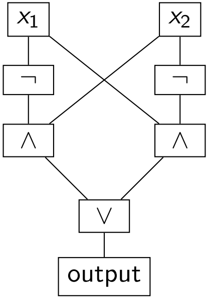
- input gates (has only outgoing edges)
- Boolean gates (and $\wedge$, or $\vee$, not $\neg$ having incoming and outgoing edges)
- output gates (only incoming edges)

Machine models:

- PRAMs (parallel random access machines)
- (Boolean) circuits

Circuit = directed acyclic graph where each vertex is either:

- input gates (has only outgoing edges)
- Boolean gates (and $\wedge$, or $\vee$, not $\neg$ having incoming and outgoing edges)
- output gates (only incoming edges)



Parallel Complexity

Machine models:

- PRAMs (parallel random access machines)
- (Boolean) circuits

Circuit = directed acyclic graph where each vertex is either:

- input gates (has only outgoing edges)
- Boolean gates (and $\wedge$, or $\vee$, not $\neg$ having incoming and outgoing edges)
- output gates (only incoming edges)

size = number of gates
depth = longest path from input to output gate
fan-in = number of input-wires per gate

NC = problems which can be solved by a family of circuits of polynomial size and polylogarithmic depth and bounded fan-in.

Inside NC:

- NC^i = solved by a family of circuits of depth $\mathcal{O}(\log^i n)$ and polynomial size with bounded fan-in (= in-degree) $\neg$, $\wedge$, $\vee$ gates.

Parallel Complexity

Inside NC:

- NC^i = solved by a family of circuits of depth $\mathcal{O}(\log^i n)$ and polynomial size with bounded fan-in (= in-degree) $\neg$, $\wedge$, $\vee$ gates.

Infinite hierarchy:

$$\mathsf{NC}^1 \qquad\qquad \subseteq \mathsf{NC}^2 \subseteq \mathsf{NC}^3 \subseteq \cdots \subseteq \mathsf{NC} \subseteq \mathsf{P}.$$

Inside NC:

- NC^i = solved by a family of circuits of depth $\mathcal{O}(\log^i n)$ and polynomial size with bounded fan-in (= in-degree) $\neg$, $\wedge$, $\vee$ gates.

Infinite hierarchy:

$$\mathsf{NC}^1 \subseteq \mathsf{LOGSPACE} \subseteq \mathsf{NC}^2 \subseteq \mathsf{NC}^3 \subseteq \cdots \subseteq \mathsf{NC} \subseteq \mathsf{P}.$$

Parallel Complexity

Inside NC:

- NC^i = solved by a family of circuits of depth $\mathcal{O}(\log^i n)$ and polynomial size with bounded fan-in (= in-degree) $\neg$, $\wedge$, $\vee$ gates.

Infinite hierarchy:

$$\mathsf{NC}^1 \subseteq \mathsf{LOGSPACE} \subseteq \mathsf{NC}^2 \subseteq \mathsf{NC}^3 \subseteq \cdots \subseteq \mathsf{NC} \subseteq \mathsf{P}.$$

Theorem (Lipton, Zalcstein, 1977 / Simon, 1979)

The word problem of linear groups is in LOGSPACE.

Parallel Complexity

Inside NC:

- NC^i = solved by a family of circuits of depth $\mathcal{O}(\log^i n)$ and polynomial size with bounded fan-in (= in-degree) $\neg$, $\wedge$, $\vee$ gates.

Infinite hierarchy:

$$\mathrm{AC}^0 \subsetneq \qquad \mathrm{NC}^1 \subseteq \mathrm{LOGSPACE} \subseteq \mathrm{NC}^2 \subseteq \mathrm{NC}^3 \subseteq \cdots \subseteq \mathrm{NC} \subseteq \mathrm{P}.$$

Theorem (Lipton, Zalcstein, 1977 / Simon, 1979)

The word problem of linear groups is in LOGSPACE.

Inside NC^1:

- AC^0 = solved by a family of circuits of constant depth and polynomial size with unbounded fan-in $\neg$, $\wedge$, $\vee$ gates.

Parallel Complexity

Inside NC:

- NC^i = solved by a family of circuits of depth $\mathcal{O}(\log^i n)$ and polynomial size with bounded fan-in (= in-degree) $\neg$, $\wedge$, $\vee$ gates.

Infinite hierarchy:

$$\mathsf{AC}^0 \subsetneq \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{LOGSPACE} \subseteq \mathsf{NC}^2 \subseteq \mathsf{NC}^3 \subseteq \cdots \subseteq \mathsf{NC} \subseteq \mathsf{P}.$$

Theorem (Lipton, Zalcstein, 1977 / Simon, 1979)

The word problem of linear groups is in LOGSPACE.

Inside NC^1:

- AC^0 = solved by a family of circuits of constant depth and polynomial size with unbounded fan-in $\neg$, $\wedge$, $\vee$ gates.
- TC^0 allows additionally majority gates: $\mathrm{Maj}(w) = 1$ iff $|w|_1 \geq |w|_0$ for $w \in \{0, 1\}^*$.

Word problem of $\mathbb{Z}$

The word problem of $\mathbb{Z}$ with generators $\{ +1, -1 \}$ is in TC^0.

Word problem of $\mathbb{Z}$

The word problem of $\mathbb{Z}$ with generators $\{+1, -1\}$ is in TC^0.

Use 0 to encode -1 and 1 for 1.

Word problem of $\mathbb{Z}$

The word problem of $\mathbb{Z}$ with generators $\{+1, -1\}$ is in TC^0.

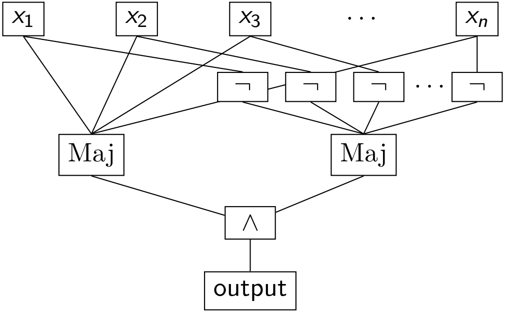
Use 0 to encode -1 and 1 for 1. Let $w \in \{0, 1\}^*$,

$$\begin{aligned} w \text{ represents } 0 \text{ in } \mathbb{Z} &\iff |w|_1 = |w|_0 \\ &\iff \mathrm{Maj}(w) \wedge \mathrm{Maj}(\neg w) \end{aligned}$$

The word problem of $\mathbb{Z}$ with generators $\{+1, -1\}$ is in TC^0.

Use 0 to encode -1 and 1 for 1. Let $w \in \{0,1\}^*$,

$$\begin{aligned} w \text{ represents } 0 \text{ in } \mathbb{Z} &\iff |w|_1 = |w|_0 \\ &\iff \mathrm{Maj}(w) \wedge \mathrm{Maj}(\neg w) \end{aligned}$$



Word problem of $\mathbb{Z}$

The word problem of $\mathbb{Z}$ with generators $\{+1, -1\}$ is in TC^0.

Use 0 to encode -1 and 1 for 1. Let $w \in \{0, 1\}^*$,

$$\begin{aligned} w \text{ represents } 0 \text{ in } \mathbb{Z} &\iff |w|_1 = |w|_0 \\ &\iff \mathrm{Maj}(w) \wedge \mathrm{Maj}(\neg w) \end{aligned}$$

Theorem (Myasnikov, W. 2017, Lohrey, W.)

If G is f.g. nilpotent or $G = H \wr \mathbb{Z}$ for H f.g. abelian, then $\textsc{PowerWP}(G)$ *is in* TC^0.

Reductions

- For a formal language $L \subseteq \{0, 1\}^*$, $\mathsf{AC}^0(L)$ allows additionally oracle gates for L.
- $L' \in \mathsf{AC}^0(L)$ means L' is AC^0-(Turing)-reducible to L.

Reductions

- For a formal language $L \subseteq \{0, 1\}^*$, $\mathsf{AC}^0(L)$ allows additionally oracle gates for L.
- $L' \in \mathsf{AC}^0(L)$ means L' is AC^0-(Turing)-reducible to L.
- Every problem in TC^0 is AC^0-reducible to Majority.

  $\rightsquigarrow$ Majority is TC^0-complete.

Reductions

- For a formal language $L \subseteq \{0,1\}^*$, $\mathsf{AC}^0(L)$ allows additionally oracle gates for L.
- $L' \in \mathsf{AC}^0(L)$ means L' is AC^0-(Turing)-reducible to L.
- Every problem in TC^0 is AC^0-reducible to Majority.
  $\rightsquigarrow$ Majority is TC^0-complete.

- $\mathsf{TC}^0 = \mathsf{AC}^0(\mathrm{WP}(\mathbb{Z})) \subseteq \mathsf{AC}^0(\mathrm{WP}(F_2))$
- $\mathsf{AC}^0(\mathrm{WP}(F_2)) \subseteq \mathsf{LOGSPACE}$

Word problem of free groups

- The word problem of free groups is in LOGSPACE (Lipton, Zalcstein, 1977).
- $\mathrm{WP}(F_k)$ is NC^1-hard for $k \geq 2$ (Robinson, 1993).

Word problem of free groups

- The word problem of free groups is in LOGSPACE (Lipton, Zalcstein, 1977).
- $\mathrm{WP}(F_k)$ is NC^1-hard for $k \geq 2$ (Robinson, 1993).
- The compressed word problem is P-complete for $k \geq 2$ (Lohrey).

Word problem of free groups

- The word problem of free groups is in LOGSPACE (Lipton, Zalcstein, 1977).
- $\mathrm{WP}(F_k)$ is NC^1-hard for $k \geq 2$ (Robinson, 1993).
- The compressed word problem is P-complete for $k \geq 2$ (Lohrey).

Theorem (Lohrey, W.)

The power word problem for free groups is in $\mathrm{AC}^0(\mathrm{WP}(F_2))$.

Overview: small circuit classes

AC^0	$\mathbb{Z}/n\mathbb{Z}$ with one monoid generator
TC^0	$\mathbb{Z}$, linear solvable, free solvable $\textsc{PowerWP}(\mathsf{Ab} \wr \mathbb{Z})$, $\textsc{PowerWP}(\text{nilpotent})$
$\mathsf{NC}^1 = \mathsf{AC}^0(\mathrm{WP}(A_5))$	finite non-solvable, regular languages

Overview: small circuit classes

AC^0	$\mathbb{Z}/n\mathbb{Z}$ with one monoid generator
TC^0	$\mathbb{Z}$, linear solvable, free solvable $\textsc{PowerWP}(\mathsf{Ab} \wr \mathbb{Z})$, $\textsc{PowerWP}(\text{nilpotent})$
$\mathsf{NC}^1 = \mathsf{AC}^0(\mathrm{WP}(A_5))$	finite non-solvable, regular languages
$\mathsf{AC}^0(\mathrm{WP}(F_2))$	virtually free, Baumslag-Solitar groups, RAAGs, free products, graph products $\textsc{PowerWP}(\text{free})$

Overview: small circuit classes

AC^0	$\mathbb{Z}/n\mathbb{Z}$ with one monoid generator
TC^0	$\mathbb{Z}$, linear solvable, free solvable $\textsc{PowerWP}(\mathrm{Ab} \wr \mathbb{Z})$, $\textsc{PowerWP}(\text{nilpotent})$
$\mathsf{NC}^1 = \mathsf{AC}^0(\mathrm{WP}(A_5))$	finite non-solvable, regular languages
$\mathsf{AC}^0(\mathrm{WP}(F_2))$	virtually free, Baumslag-Solitar groups, RAAGs, free products, graph products $\textsc{PowerWP}(\text{free})$
LOGSPACE	linear groups, Grigorchuk group (not know to be complete)
NC^2	hyperbolic groups (not know to be complete)

Overview: small circuit classes

AC^0	$\mathbb{Z}/n\mathbb{Z}$ with one monoid generator
TC^0	$\mathbb{Z}$, linear solvable, free solvable $\mathrm{PowerWP}(\mathsf{Ab} \wr \mathbb{Z})$, $\mathrm{PowerWP}(\text{nilpotent})$
$\mathsf{NC}^1 = \mathsf{AC}^0(\mathrm{WP}(A_5))$	finite non-solvable, regular languages
$\mathsf{AC}^0(\mathrm{WP}(F_2))$	virtually free, Baumslag-Solitar groups, RAAGs, free products, graph products $\mathrm{PowerWP}(\text{free})$
LOGSPACE	linear groups, Grigorchuk group (not know to be complete)
NC^2	hyperbolic groups (not know to be complete)
P polynomial time	compressed word problem of free groups,...

Open Questions I

- Is there a natural (non-group theoretic) problem which is $\mathsf{AC}^0(\mathrm{WP}(F_2))$-complete?
- Is $\mathrm{WP}(F_2)$ complete for $\mathsf{AC}^0(\mathrm{WP}(F_2))$ under many-one reductions?
- Is there a $\mathsf{AC}^0(\mathrm{WP}(F_2))$-complete problem under many-one reductions?

Open Questions I

- Is there a natural (non-group theoretic) problem which is $\mathsf{AC}^0(\mathrm{WP}(F_2))$-complete?
- Is $\mathrm{WP}(F_2)$ complete for $\mathsf{AC}^0(\mathrm{WP}(F_2))$ under many-one reductions?
- Is there a $\mathsf{AC}^0(\mathrm{WP}(F_2))$-complete problem under many-one reductions?
- How does the word problem of the Grigorchuk group relate to this class?
- Precise complexity for hyperbolic groups.

Open Questions I

- Is there a natural (non-group theoretic) problem which is $\mathsf{AC}^0(\mathrm{WP}(F_2))$-complete?
- Is $\mathrm{WP}(F_2)$ complete for $\mathsf{AC}^0(\mathrm{WP}(F_2))$ under many-one reductions?
- Is there a $\mathsf{AC}^0(\mathrm{WP}(F_2))$-complete problem under many-one reductions?

- How does the word problem of the Grigorchuk group relate to this class?
- Precise complexity for hyperbolic groups.

Or even more challenging:

- Separation results: $\mathsf{TC}^0 \neq \mathsf{NC}^1$? $\mathsf{AC}^0(\mathrm{WP}(F_2)) \neq \mathsf{NC}^1$?...
- Can a non-solvable group have word problem in TC^0?

Power word problem in free groups

Power word problem: Given $p_1, \ldots, p_k \in \Sigma^*$ and $x_1, \ldots, x_k \in \mathbb{Z}$.
Question: $p_1^{x_1} \cdots p_k^{x_k} = 1$ in G?

Theorem (Lohrey, W.)

The power word problem for free groups is in $\mathrm{AC}^0(\mathrm{WP}(F_2))$.

Theorem (Lohrey, W.)

$\textsc{PowerWP}(G * H) \in \mathrm{AC}^0(\textsc{PowerWP}(G), \textsc{PowerWP}(H), \mathrm{WP}(F_2))$.

Power word problem in free groups

Power word problem: Given $p_1, \ldots, p_k \in \Sigma^*$ and $x_1, \ldots, x_k \in \mathbb{Z}$.
Question: $p_1^{x_1} \cdots p_k^{x_k} = 1$ in G?

Theorem (Lohrey, W.)

The power word problem for free groups is in $\mathrm{AC}^0(\mathrm{WP}(F_2))$.

Theorem (Lohrey, W.)

$\textsc{PowerWP}(G * H) \in \mathrm{AC}^0(\textsc{PowerWP}(G), \textsc{PowerWP}(H), \mathrm{WP}(F_2))$.

Three steps:

- Preprocessing
- Make exponents small
- Solve regular word problem

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}\,a\,b^{-100}\,b^{100}\,a\,b^{-100}\,b^{100}\,\overline{a}\,\overline{a}\,(a\,b)^{-1000}$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$
\begin{aligned}
(a\, b)^{1000} a\, b^{-100} b^{100} a\, b^{-100} b^{100} \overline{a}\, \overline{a}\, (a\, b)^{-1000} \\
= (a\, b)^{1000} a\, a\, b^{-100} b^{100} \overline{a}\, \overline{a}\, (a\, b)^{-1000}
\end{aligned}
$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}\,a\,b^{-100}\,b^{100}\,a\,b^{-100}\,b^{100}\,\overline{a}\,\overline{a}\,(a\,b)^{-1000}$$
$$= (a\,b)^{1000}\,a\,a\,\overline{a}\,\overline{a}\,(a\,b)^{-1000}$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}\,a\,b^{-100}\,b^{100}\,a\,b^{-100}\,b^{100}\,\overline{a}\,\overline{a}\,(a\,b)^{-1000}$$
$$= (a\,b)^{1000}(a\,b)^{-1000}$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}\,a\,b^{-100}\,b^{100}\,a\,b^{-100}\,b^{100}\,\overline{a}\,\overline{a}\,(a\,b)^{-1000}$$
$$= 1$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\bar{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}\,a\,b^{-100}\,b^{100}\,a\,b^{-100}\,b^{100}\,\bar{a}\,\bar{a}\,(a\,b)^{-1000}$$
$$= 1$$

Example 2

$$b^{123}(b\,a\,a)^{123}\,a^{-246}\,b^{-123}(\bar{b}\,\bar{a})^{123}\,a^{123}$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}a\,b^{-100}b^{100}a\,b^{-100}b^{100}\overline{a}\,\overline{a}\,(a\,b)^{-1000}$$
$$= 1$$

Example 2

$$b^{123}(b\,a\,a)^{123}a^{-246}b^{-123}(\overline{b}\,\overline{a})^{123}a^{123} \neq 1$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}\,a\,b^{-100}\,b^{100}\,a\,b^{-100}\,b^{100}\,\overline{a}\,\overline{a}\,(a\,b)^{-1000}$$
$$= 1$$

Example 2

$$b^{123}(b\,a\,a)^{123}\,a^{-246}\,b^{-123}(\overline{b}\,\overline{a})^{123}\,a^{123} \neq 1$$

Example 3

$$(a\,a)^{500}\,(\overline{a})^{999}\,\overline{a}$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}\,a\,b^{-100}\,b^{100}\,a\,b^{-100}\,b^{100}\,\overline{a}\,\overline{a}\,(a\,b)^{-1000}$$
$$= 1$$

Example 2

$$b^{123}(b\,a\,a)^{123}\,a^{-246}\,b^{-123}(\overline{b}\,\overline{a})^{123}\,a^{123} \neq 1$$

Example 3

$$(a\,a)^{500}\,(\overline{a})^{999}\,\overline{a} = 1$$

Examples: Power word problem in free groups

Let $F = F(\{a, b\})$ be the free group. Write $\overline{a}$ for a^{-1}

Example 1

$$(a\,b)^{1000}\,a\,b^{-100}\,b^{100}\,a\,b^{-100}\,b^{100}\,\overline{a}\,\overline{a}\,(a\,b)^{-1000} = 1$$

Example 2

$$b^{123}(b\,a\,a)^{123}a^{-246}\,b^{-123}(\overline{b}\,\overline{a})^{123}a^{123} \neq 1$$

Example 3

$$(a\,a)^{500}\,(\overline{a})^{999}\,\overline{a} = 1$$

Example 4

$$(b\,a\,a\,\overline{a}\,b\,a)^{500}\,(b)^{2}\,(\overline{b}\,\overline{b}\,\overline{a}b)^{999}\,(\overline{b}\,\overline{a}\,\overline{b}\,\overline{b}\,a\,b)^{1}(a\,b)^{-1}$$

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,
(2) p is primitive,
(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\left\{\, uv \;\middle|\; vu = p \text{ or } vu = p^{-1} \,\right\}$).

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,

(2) p is primitive,

(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\{ uv \mid vu = p \text{ or } vu = p^{-1} \}$).

$$\Omega = \{ a, b, ab, a\overline{b}, aab, aa\overline{b}, \dots \}$$

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,

(2) p is primitive,

(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\{\, uv \mid vu = p \text{ or } vu = p^{-1} \,\}$).

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,

(2) p is primitive,

(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\{ uv \mid vu = p \text{ or } vu = p^{-1} \}$).

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,

(2) p is primitive,

(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\{ uv \mid vu = p \text{ or } vu = p^{-1} \}$).

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Proof.

- By (1), $v = w^{-1}$ as words.

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,

(2) p is primitive,

(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\{ uv \mid vu = p \text{ or } vu = p^{-1} \}$).

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Proof.

- By (1), $v = w^{-1}$ as words. $\rightsquigarrow$ v has periods $|p|$ and $|q|$.

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,

(2) p is primitive,

(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\{\, uv \;\middle|\; vu = p \text{ or } vu = p^{-1} \,\}$).

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Proof.

- By (1), $v = w^{-1}$ as words. $\rightsquigarrow$ v has periods $|p|$ and $|q|$.
- By Fine and Wilf's theorem v has period $\gcd(|p|, |q|)$.

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,
(2) p is primitive,
(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\left\{ uv \;\middle|\; vu = p \text{ or } vu = p^{-1} \right\}$).

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Proof.

- By (1), $v = w^{-1}$ as words. $\rightsquigarrow$ v has periods $|p|$ and $|q|$.
- By Fine and Wilf's theorem v has period $\gcd(|p|, |q|)$.
  $\rightsquigarrow$ also p and q.

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,

(2) p is primitive,

(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\{ uv \mid vu = p \text{ or } vu = p^{-1} \}$).

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Proof.

- By (1), $v = w^{-1}$ as words. $\rightsquigarrow$ v has periods $|p|$ and $|q|$.
- By Fine and Wilf's theorem v has period $\gcd(|p|, |q|)$.
  $\rightsquigarrow$ also p and q.
- By (2), $|p| = |q|$.

Preprocessing

$\Omega \subseteq \Sigma^+$ is set of non-empty words p with

(1) p is cyclically reduced,

(2) p is primitive,

(3) p is lexicographically minimal among all cyclic permutations of p and p^{-1} (i. e., in $\left\{ uv \;\middle|\; vu = p \text{ or } vu = p^{-1} \right\}$).

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Proof.

- By (1), $v = w^{-1}$ as words. $\rightsquigarrow$ v has periods $|p|$ and $|q|$.
- By Fine and Wilf's theorem v has period $\gcd(|p|, |q|)$.
  $\rightsquigarrow$ also p and q.
- By (2), $|p| = |q|$.
- By (3), since p is a factor of w^{-1}, we get $p = q$. $\square$

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$(b\,a\,a\,\overline{a}\,b\,a)^{500}\,(b)^2\,(\overline{b}\,\overline{b}\,\overline{a}b)^{999}\,(\overline{b}\,\overline{a}\,\overline{b}\,\overline{b}\,a\,b)^1(a\,b\,)^{-1}$$

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$(b\, a\, a\, \overline{a}\, b\, a)^{500}\, (b)^2\, (\overline{b}\, \overline{b}\, \overline{a} b)^{999}\, (\overline{b}\, \overline{a}\, \overline{b}\, \overline{b}\, a\, b)^1 (a\, b\,)^{-1}$$

- Freely reduce the q_i.

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$(b\,a\,b\,a)^{500}\,(b)^2\,(\overline{b}\,\overline{b}\,\overline{a}b)^{999}\,(\overline{b}\,\overline{a}\,\overline{b}\,\overline{b}\,a\,b)^1(a\,b\,)^{-1}$$

- Freely reduce the q_i.

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$(b\,a\,b\,a)^{500}\,(b)^2\,(\overline{b}\,\overline{b}\,\overline{a}b)^{999}\,(\overline{b}\,\overline{a}\,\overline{b}\,\overline{b}\,a\,b)^1(a\,b\,)^{-1}$$

- Freely reduce the q_i.
- Make each q_i cyclically reduced.

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$(b\,a\,b\,a)^{500}\,(b)^2\,\overline{b}\,(\overline{b}\,\overline{a})^{999}\,b\,\overline{b}\,\overline{a}\,(\overline{b}\,\overline{b})^1\,a\,b\,(a\,b\,)^{-1}$$

- Freely reduce the q_i.
- Make each q_i cyclically reduced.

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$(b\,a\,b\,a)^{500}\,(b)^2\,\overline{b}\,(\overline{b}\,\overline{a})^{999}\,b\,\overline{b}\,\overline{a}\,(\overline{b}\,\overline{b})^1\,a\,b\,(a\,b\,)^{-1}$$

- Freely reduce the q_i.
- Make each q_i cyclically reduced.
- Make each q_i primitive.

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$(b\,a)^{1000}\,(b)^2\,\overline{b}\,(\overline{b}\,\overline{a})^{999}\,b\,\overline{b}\,\overline{a}\,(\overline{b})^2\,a\,b\,(a\,b)^{-1}$$

- Freely reduce the q_i.
- Make each q_i cyclically reduced.
- Make each q_i primitive.

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$(b\,a)^{1000}\,(b)^2\,\overline{b}\,(\overline{b}\,\overline{a})^{999}\,b\,\overline{b}\,\overline{a}\,(\overline{b})^2 a\,b\,(a\,b)^{-1}$$

- Freely reduce the q_i.
- Make each q_i cyclically reduced.
- Make each q_i primitive.
- Make q_i lex. minimal in $\{\, uv \mid vu = q_i \text{ or } vu = q_i^{-1} \,\}$

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \qquad (1)$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$b\,(a\,b)^{1000}\,\overline{b}\,(b)^2\,\overline{b}\,(a\,b)^{-999}\,b\,\overline{b}\,\overline{a}\,(b)^{-2}\,a\,b\,(a\,b)^{-1}$$

- Freely reduce the q_i.
- Make each q_i cyclically reduced.
- Make each q_i primitive.
- Make q_i lex. minimal in $\left\{ uv \;\middle|\; vu = q_i \text{ or } vu = q_i^{-1} \right\}$

This yields $\qquad s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n$

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$b\,(a\,b)^{1000}\,\overline{b}\,(b)^2\,\overline{b}\,(a\,b)^{-999}\,b\,\overline{b}\,\overline{a}\,(b)^{-2}\,a\,b\,(a\,b)^{-1}$$

- Freely reduce the q_i.
- Make each q_i cyclically reduced.
- Make each q_i primitive.
- Make q_i lex. minimal in $\left\{ uv \;\middle|\; vu = q_i \text{ or } vu = q_i^{-1} \right\}$

This yields $\qquad s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n$

- Freely reduce the s_i.

Preprocessing

The first aim is to rewrite an input word $q_1^{y_1} \cdots q_n^{y_n}$ in the form

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.} \tag{1}$$

Lemma

Given a power word v, a power word w of the form (1) with $v =_F w$ can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

$$b\,(a\,b)^{1000}\,\overline{b}\,(b)^2\,\overline{b}\,(a\,b)^{-999}\,\overline{a}\,(b)^{-2}\,a\,b\,(a\,b)^{-1}$$

- Freely reduce the q_i.
- Make each q_i cyclically reduced.
- Make each q_i primitive.
- Make q_i lex. minimal in $\{\, uv \mid vu = q_i \text{ or } vu = q_i^{-1} \,\}$

This yields $s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n$

- Freely reduce the s_i.

Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$.

Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.

Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$*.*

Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

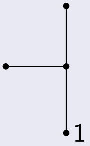
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

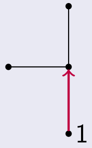
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \dots, n\} \times \{1, \dots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

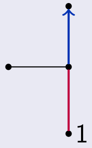
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

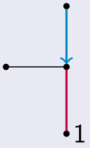
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

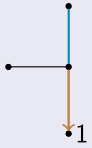
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

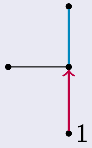
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

$$\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ b & \underbrace{b & \overline{b} & \overline{b} & b}_{w_{1,5} = 1} & \overline{a} & a & \overline{b} & b \end{array}$$



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

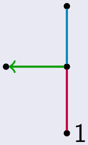
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

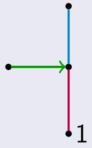
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

Proof.

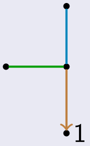
Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\bar{b}$	$\bar{b}$	b	$\bar{a}$	a	$\bar{b}$	b

$w_{4,8} = 1$



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

Proof.

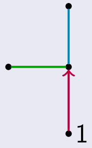
Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

$$\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ b & b & \overline{b} & \overline{b} & b & \overline{a} & a & \overline{b} & b \end{array}$$

$w_{1,9} = w_{5,9} = 1$



Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathrm{AC}^0(\mathrm{WP}(F))$.

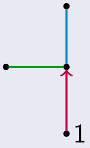
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



$w =_F b$

Computing freely reduced words

Proposition (W., 2016)

Freely reduced words can be computed in $\mathsf{AC}^0(\mathrm{WP}(F))$.

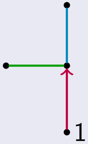
Proof.

Input: $w = w_1 \cdots w_n$ with $w_i \in \Sigma \cup \Sigma^{-1}$. Set $w_{i,j} = w_{i+1} \cdots w_j$.
Define an equivalence relation $\approx \subseteq \{1, \ldots, n\} \times \{1, \ldots, n\}$ by

$$i \approx j \iff w_i = w_j \text{ and } \begin{cases} w_{i,j} =_F 1 & \text{if } i < j, \\ w_{j,i} =_F 1 & \text{if } j < i. \end{cases}$$

$\rightsquigarrow$ $i \approx j$ iff w_i and w_j are the same edge in the Cayley graph

1	2	3	4	5	6	7	8	9
b	b	$\overline{b}$	$\overline{b}$	b	$\overline{a}$	a	$\overline{b}$	b



$w =_F b$

Can be checked in $\mathsf{AC}^0(\mathrm{WP}(F))$ for all pairs i, j whether $i \approx j$.

Proof. (Contd.)

Define a partial map

$$\overline{\cdot} : \{1, \ldots, n\}/\!\approx \;\to \{1, \ldots, n\}/\!\approx$$

$$[i] \mapsto [j] \qquad \text{if there is some } j \text{ with } w_i = \overline{w}_j \text{ and } w_{i,j-1} =_F 1 \text{ (resp. } w_{j,i-1} =_F 1\text{).}$$

We have

- $[i] = \overline{[j]} \iff w_i$ and w_j are inverse edges in the Cayley graph.

Proof. (Contd.)

Define a partial map

$$\overline{\cdot} : \{1, \ldots, n\}/{\approx} \to \{1, \ldots, n\}/{\approx}$$

$$[i] \mapsto [j] \qquad \text{if there is some } j \text{ with } w_i = \overline{w}_j \text{ and } w_{i,j-1} =_F 1 \text{ (resp. } w_{j,i-1} =_F 1\text{).}$$

We have

- $[i] = \overline{[j]} \iff w_i$ and w_j are inverse edges in the Cayley graph.
- $\left| |[i]| - |\overline{[i]}| \right| \leq 1$ for all i

Computing freely reduced words

Proof. (Contd.)

Define a partial map

$$\overline{\cdot} : \{1, \ldots, n\}/{\approx} \to \{1, \ldots, n\}/{\approx}$$
$$[i] \mapsto [j] \qquad \text{if there is some } j \text{ with } w_i = \overline{w}_j \text{ and } w_{i,j-1} =_F 1 \text{ (resp. } w_{j,i-1} =_F 1\text{).}$$

We have

- $[i] = \overline{[j]} \iff w_i$ and w_j are inverse edges in the Cayley graph.
- $\left| |[i]| - |\overline{[i]}| \right| \leq 1$ for all i
- if $|[i]| = |\overline{[i]}|$, all letters in $[i]$ cancel

Computing freely reduced words

Proof. (Contd.)

Define a partial map

$$\overline{\cdot} : \{\, 1, \ldots, n \,\}/{\approx} \to \{\, 1, \ldots, n \,\}/{\approx}$$

$$[i] \mapsto [j] \qquad \text{if there is some } j \text{ with } w_i = \overline{w}_j \text{ and } w_{i,j-1} =_F 1 \text{ (resp. } w_{j,i-1} =_F 1\text{).}$$

We have

- $[i] = \overline{[j]} \iff w_i$ and w_j are inverse edges in the Cayley graph.
- $\left| |[i]| - |\overline{[i]}| \right| \leq 1$ for all i
- if $|[i]| = |\overline{[i]}|$, all letters in $[i]$ cancel
- if $|[i]| > |\overline{[i]}|$, after any sequence of free reductions, there remains one letter w_j for some $j \in [i]$.

Computing freely reduced words

Proof. (Contd.)

Define a partial map

$$\begin{aligned}\overline{\cdot} : \{1, \ldots, n\}/{\approx} &\to \{1, \ldots, n\}/{\approx}\\ [i] &\mapsto [j] \qquad \text{if there is some } j \text{ with } w_i = \overline{w}_j \text{ and}\\ & \qquad\qquad w_{i,j-1} =_F 1 \text{ (resp. } w_{j,i-1} =_F 1).\end{aligned}$$

We have

- $[i] = \overline{[j]} \iff w_i$ and w_j are inverse edges in the Cayley graph.
- $\Big| |[i]| - |\overline{[i]}| \Big| \leq 1$ for all i
- if $|[i]| = |\overline{[i]}|$, all letters in $[i]$ cancel
- if $|[i]| > |\overline{[i]}|$, after any sequence of free reductions, there remains one letter w_j for some $j \in [i]$.

Output all w_j with $j = \max [i]$ for some i with $|[i]| > |\overline{[i]}|$ and delete the other letters.

Make exponents small

Now we have a "nice" instance

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.}$$

We know that

- if a long factor of $p_i^{x_i}$ cancels with a factor of $p_j^{x_j}$, then $p_i = p_j$

Make exponents small

Now we have a "nice" instance

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.}$$

We know that

- if a long factor of $p_i^{x_i}$ cancels with a factor of $p_j^{x_j}$, then $p_i = p_j$

Idea:

- Decrease all exponents of p_i simultaneously.

Make exponents small

Now we have a "nice" instance

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.}$$

We know that

- if a long factor of $p_i^{x_i}$ cancels with a factor of $p_j^{x_j}$, then $p_i = p_j$

Idea:

- Decrease all exponents of p_i simultaneously.

But: cannot delete them entirely:

$$a^{100}\, b\, a^{-100}\, \overline{b} \neq 1, \text{ but } a^{0}\, b\, a^{0}\, \overline{b} = 1$$

Make exponents small

Now we have a "nice" instance

$$w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n \qquad \text{with } p_i \in \Omega \text{ and } s_i \text{ freely reduced.}$$

We know that

- if a long factor of $p_i^{x_i}$ cancels with a factor of $p_j^{x_j}$, then $p_i = p_j$

Idea:

- Decrease all exponents of p_i simultaneously.

But: cannot delete them entirely:

$$a^{100} b\, a^{-100} \overline{b} \neq 1, \text{ but } a^{0} b\, a^{0} \overline{b} = 1$$

Nor down to 1:

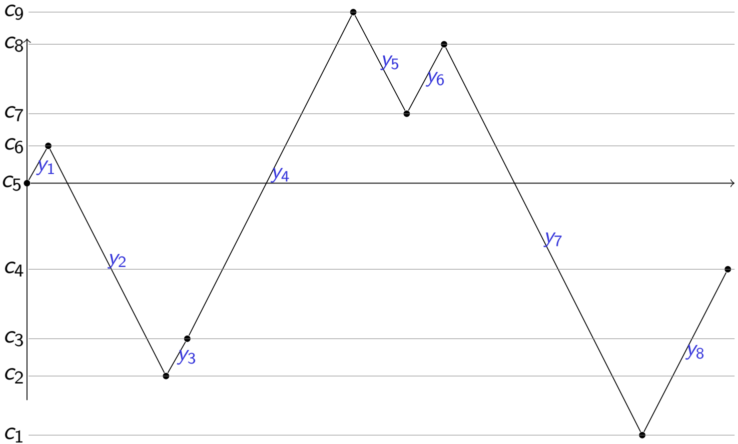
$$a^{100} (\overline{a}\, b\, a)^1 a^{-100} \overline{b} \neq 1 \text{ but } a^{1} (\overline{a}\, b\, a)^1 a^{-1} \overline{b} = 1$$

Make exponents small

Write $w = u_0 p^{y_1} u_1 \cdots p^{y_m} u_m$ for some $p \in \Omega$ such that u_i does not contain p with exponents.

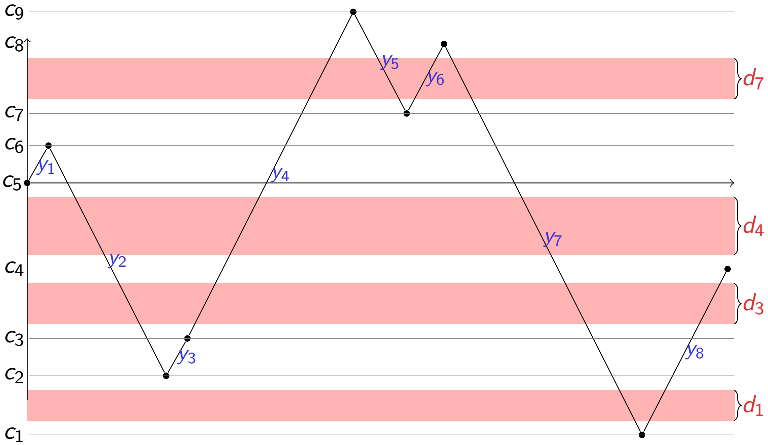
Make exponents small

Write $w = u_0 p^{y_1} u_1 \cdots p^{y_m} u_m$ for some $p \in \Omega$ such that u_i does not contain p with exponents.



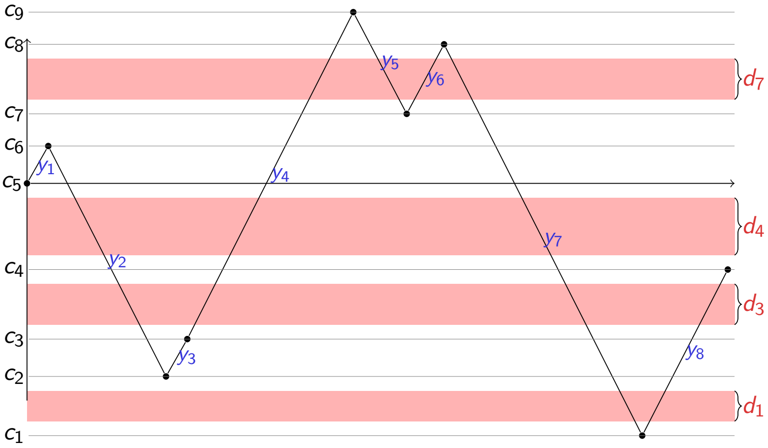
Make exponents small

Write $w = u_0 p^{y_1} u_1 \cdots p^{y_m} u_m$ for some $p \in \Omega$ such that u_i does not contain p with exponents.



Make exponents small

Write $w = u_0 p^{y_1} u_1 \cdots p^{y_m} u_m$ for some $p \in \Omega$ such that u_i does not contain p with exponents.



Define $\mathcal{S}(w) = u_0 p^{z_1} u_1 \cdots p^{z_m} u_m$ where $z_i = y_i - \mathrm{sign}(y_i) \cdot \sum_{j \in C_i} d_j$

Make exponents small

Proposition

$$w =_F 1 \iff \mathcal{S}(w) =_F 1.$$

Make exponents small

Proposition

$w =_F 1 \iff \mathcal{S}(w) =_F 1.$

Proof of the main theorem.

- Preprocessing gives a "nice word" $w = s_0 p_1^{x_1} s_1 \cdots p_n^{x_n} s_n$.
- For all $p \in \Omega$ which appear in w, compute $\mathcal{S}(w)$ in parallel (iterated addition $\rightsquigarrow$ in TC^0).
- Yields a word of polynomial length $\rightsquigarrow$ apply the ordinary word problem.

Further results on the power word problem

Theorem (Lohrey, W.)

Let G be f.g. and $H \leq G$ of finite index. Then $\textsc{PowerWP}(G)$ *is* NC^1*-many-one-reducible to* $\textsc{PowerWP}(H)$.

Further results on the power word problem

Theorem (Lohrey, W.)

Let G be f.g. and $H \leq G$ of finite index. Then $\textsc{PowerWP}(G)$ *is* NC^1*-many-one-reducible to* $\textsc{PowerWP}(H)$*.*

Corollary

The power word problem of f.g. virtually free groups is in $\mathsf{AC}^0(\mathrm{WP}(F_2))$*.*

Further results on the power word problem

Theorem (Lohrey, W.)

Let G be f.g. and $H \leq G$ of finite index. Then $\textsc{PowerWP}(G)$ *is* NC^1*-many-one-reducible to* $\textsc{PowerWP}(H)$.

Corollary

The power word problem of f.g. virtually free groups is in $\mathsf{AC}^0(\mathrm{WP}(F_2))$.

Theorem (Lohrey, W.)

Let G be either

- *finite non-solvable*
- *f.g. free of rank ≥ 2.*

Then $\textsc{PowerWP}(G \wr \mathbb{Z})$ *is* coNP*-complete.*

Proof: coNP hardness

$\textsc{CNF-UnSAT} \leq \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Proof: coNP hardness

$\textsc{CNF-UnSAT} \leq \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Let $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$; follow Robinson's proof that $\mathrm{WP}(F_2)$ is NC^1-hard:

Proof: coNP hardness

$\text{CNF-}\textsc{UnSAT} \leq \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Let $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$; follow Robinson's proof that $\mathrm{WP}(F_2)$ is NC^1-hard:

- every CNF formula is an NC^1 circuit (logarithmic depth)

Proof: coNP hardness

$\textsc{CNF-UnSAT} \leq \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Let $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$; follow Robinson's proof that $\mathrm{WP}(F_2)$ is NC^1-hard:

- every CNF formula is an NC^1 circuit (logarithmic depth)

Given a formula F over variables $\{X_1, \dots, X_m\}$, construct a word $w_F \in \left(\{a^{\pm 1}, b^{\pm 1}\} \cup \{Y_1^{\pm 1}, \dots, Y_m^{\pm 1}, \widetilde{Y}_1^{\pm 1}, \dots, \widetilde{Y}_m^{\pm 1}\}\right)^*$ such that for any valuation $\sigma : \{X_1, \dots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

Proof: coNP hardness

$\textsc{CNF-UnSAT} \le \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Let $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$; follow Robinson's proof that $\mathrm{WP}(F_2)$ is NC^1-hard:

- every CNF formula is an NC^1 circuit (logarithmic depth)

Given a formula F over variables $\{X_1, \ldots, X_m\}$, construct a word $w_F \in \left(\{a^{\pm 1}, b^{\pm 1}\} \cup \{Y_1^{\pm 1}, \ldots, Y_m^{\pm 1}, \widetilde{Y}_1^{\pm 1}, \ldots, \widetilde{Y}_m^{\pm 1}\}\right)^*$ such that for any valuation $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

where $\sigma'(Y_i) = \begin{cases} 1 & \text{if } \sigma(X_i) = 0 \\ a & \text{if } \sigma(X_i) = 1 \end{cases}$ and $\sigma'(\widetilde{Y}_i) = \begin{cases} a & \text{if } \sigma(X_i) = 0, \\ 1 & \text{if } \sigma(X_i) = 1. \end{cases}$

Proof: coNP hardness

$\textsc{CNF-UnSAT} \leq \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Let $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$; follow Robinson's proof that $\mathrm{WP}(F_2)$ is NC^1-hard:

- every CNF formula is an NC^1 circuit (logarithmic depth)

Given a formula F over variables $\{X_1, \ldots, X_m\}$, construct a word $w_F \in \left(\{a^{\pm 1}, b^{\pm 1}\} \cup \{Y_1^{\pm 1}, \ldots, Y_m^{\pm 1}, \widetilde{Y}_1^{\pm 1}, \ldots, \widetilde{Y}_m^{\pm 1}\}\right)^*$ such that for any valuation $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

where $\sigma'(Y_i) = \begin{cases} 1 & \text{if } \sigma(X_i) = 0 \\ a & \text{if } \sigma(X_i) = 1 \end{cases}$ and $\sigma'(\widetilde{Y}_i) = \begin{cases} a & \text{if } \sigma(X_i) = 0, \\ 1 & \text{if } \sigma(X_i) = 1. \end{cases}$

- $F \vee G \rightsquigarrow w_F w_G$ + padding

Proof: coNP hardness

$\textsc{CNF-UnSAT} \leq \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Let $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$; follow Robinson's proof that $\mathrm{WP}(F_2)$ is NC^1-hard:

- every CNF formula is an NC^1 circuit (logarithmic depth)

Given a formula F over variables $\{X_1, \dots, X_m\}$, construct a word $w_F \in \left(\{a^{\pm 1}, b^{\pm 1}\} \cup \{Y_1^{\pm 1}, \dots, Y_m^{\pm 1}, \widetilde{Y}_1^{\pm 1}, \dots, \widetilde{Y}_m^{\pm 1}\}\right)^*$ such that for any valuation $\sigma : \{X_1, \dots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

where $\sigma'(Y_i) = \begin{cases} 1 & \text{if } \sigma(X_i) = 0 \\ a & \text{if } \sigma(X_i) = 1 \end{cases}$ and $\sigma'(\widetilde{Y}_i) = \begin{cases} a & \text{if } \sigma(X_i) = 0, \\ 1 & \text{if } \sigma(X_i) = 1. \end{cases}$

- $F \vee G \rightsquigarrow w_F w_G$ + padding $\qquad \rightsquigarrow a\, b\, w_F\, b\, w_G\, \overline{b}\, \overline{b}\, \overline{a}$

Proof: coNP hardness

$\textsc{CNF-UnSAT} \le \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Let $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$; follow Robinson's proof that $\mathrm{WP}(F_2)$ is NC^1-hard:

- every CNF formula is an NC^1 circuit (logarithmic depth)

Given a formula F over variables $\{X_1, \ldots, X_m\}$, construct a word $w_F \in \left(\{a^{\pm 1}, b^{\pm 1}\} \cup \{Y_1^{\pm 1}, \ldots, Y_m^{\pm 1}, \widetilde{Y}_1^{\pm 1}, \ldots, \widetilde{Y}_m^{\pm 1}\}\right)^*$ such that for any valuation $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

where $\sigma'(Y_i) = \begin{cases} 1 & \text{if } \sigma(X_i) = 0 \\ a & \text{if } \sigma(X_i) = 1 \end{cases}$ and $\sigma'(\widetilde{Y}_i) = \begin{cases} a & \text{if } \sigma(X_i) = 0, \\ 1 & \text{if } \sigma(X_i) = 1. \end{cases}$

- $F \vee G \rightsquigarrow w_F w_G$ + padding $\rightsquigarrow a\, b\, w_F\, b\, w_G\, \overline{b}\, \overline{b}\, \overline{a}$
- $F \wedge G \rightsquigarrow [w_F, w_G]$ + padding $\rightsquigarrow a\, [b\, w_F \overline{b},\ b\, b\, w_G \overline{b}\, \overline{b}] \overline{a}$

Proof: coNP hardness

$\textsc{CNF-UnSAT} \leq \textsc{PowerWP}(F_2 \wr \mathbb{Z})$:

Let $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$; follow Robinson's proof that $\mathrm{WP}(F_2)$ is NC^1-hard:

- every CNF formula is an NC^1 circuit (logarithmic depth)

Given a formula F over variables $\{X_1, \ldots, X_m\}$, construct a word $w_F \in \left(\{a^{\pm 1}, b^{\pm 1}\} \cup \{Y_1^{\pm 1}, \ldots, Y_m^{\pm 1}, \widetilde{Y}_1^{\pm 1}, \ldots, \widetilde{Y}_m^{\pm 1}\}\right)^*$ such that for any valuation $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

where $\sigma'(Y_i) = \begin{cases} 1 & \text{if } \sigma(X_i) = 0 \\ a & \text{if } \sigma(X_i) = 1 \end{cases}$ and $\sigma'(\widetilde{Y}_i) = \begin{cases} a & \text{if } \sigma(X_i) = 0, \\ 1 & \text{if } \sigma(X_i) = 1. \end{cases}$

- $F \vee G \rightsquigarrow w_F w_G$ + padding $\rightsquigarrow a\, b\, w_F\, b\, w_G\, \overline{b}\, \overline{b}\, \overline{a}$
- $F \wedge G \rightsquigarrow [w_F, w_G]$ + padding $\rightsquigarrow a\, [b\, w_F \overline{b}\,,\; b\, b\, w_G \overline{b}\, \overline{b}] \overline{a}$
- logarithmic depth $\rightsquigarrow$ polynomial size

Proof: coNP hardness

- $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$.
- For any assignment $\sigma : \{X_1, \ldots, X_m\} \to \{0,1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

Evaluate w_F for all valuations "in parallel":

Proof: coNP hardness

- $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$.
- For any assignment $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

Evaluate w_F for all valuations "in parallel":

- Let $p_1, \ldots, p_m \in \mathbb{N}$ be pairwise coprime, $M = \prod p_i$, $M_i = M/p_i$

Proof: coNP hardness

- $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$.
- For any assignment $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

Evaluate w_F for all valuations "in parallel":

- Let $p_1, \ldots, p_m \in \mathbb{N}$ be pairwise coprime, $M = \prod p_i$, $M_i = M/p_i$
- $Y_i \mapsto (a\, \underbrace{t \cdots t}_{p_i})^{M_i} t^{-M} = \underbrace{(a, \underbrace{1, \ldots, 1}_{p_i - 1}, \ldots, a, \underbrace{1, \ldots, 1}_{p_i - 1})}_{M_i \text{ times}}$

Proof: coNP hardness

- $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$.
- For any assignment $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

Evaluate w_F for all valuations "in parallel":

- Let $p_1, \ldots, p_m \in \mathbb{N}$ be pairwise coprime, $M = \prod p_i$, $M_i = M/p_i$
- $Y_i \mapsto (a \underbrace{t \cdots t}_{p_i})^{M_i} t^{-M} = \underbrace{(a, \underbrace{1, \ldots, 1}_{p_i - 1}, \ldots, a, \underbrace{1, \ldots, 1}_{p_i - 1})}_{M_i \text{ times}}$

$\rightsquigarrow$ a at positions $\equiv 0 \mod p_i$

Proof: coNP hardness

- $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$.
- For any assignment $\sigma : \{X_1, \ldots, X_m\} \to \{0,1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

Evaluate w_F for all valuations "in parallel":

- Let $p_1, \ldots, p_m \in \mathbb{N}$ be pairwise coprime, $M = \prod p_i$, $M_i = M/p_i$
- $Y_i \mapsto (a\underbrace{t \cdots t}_{p_i})^{M_i} t^{-M} = \underbrace{(a, \underbrace{1, \ldots, 1}_{p_i - 1}, \ldots, a, \underbrace{1, \ldots, 1}_{p_i - 1})}_{M_i \text{ times}}$

$\rightsquigarrow$ a at positions $\equiv 0 \mod p_i$

$$\tilde{Y}_i \mapsto (t\underbrace{at \cdots at}_{p_i - 1})^{M_i} t^{-M} = (1, \underbrace{a, \ldots, a}_{p_i - 1}, \ldots, 1, \underbrace{a, \ldots, a}_{p_i - 1})$$

Proof: coNP hardness

- $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$.
- For any assignment $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

Evaluate w_F for all valuations "in parallel":

- Let $p_1, \ldots, p_m \in \mathbb{N}$ be pairwise coprime, $M = \prod p_i$, $M_i = M/p_i$
- $Y_i \mapsto (a \underbrace{t \cdots t}_{p_i})^{M_i} t^{-M} = \underbrace{(a, \underbrace{1, \ldots, 1}_{p_i - 1}, \ldots, a, \underbrace{1, \ldots, 1}_{p_i - 1})}_{M_i \text{ times}}$

  $\rightsquigarrow a$ at positions $\equiv 0 \mod p_i$

  $\tilde{Y}_i \mapsto (t \underbrace{at \cdots at}_{p_i - 1})^{M_i} t^{-M} = (1, \underbrace{a, \ldots, a}_{p_i - 1}, \ldots, 1, \underbrace{a, \ldots, a}_{p_i - 1})$

  $\rightsquigarrow a$ at positions $\not\equiv 0 \mod p_i$

Proof: coNP hardness

- $F_2 \wr \mathbb{Z} = \langle a, b, t \rangle$.
- For any assignment $\sigma : \{X_1, \ldots, X_m\} \to \{0, 1\}$

$$\sigma(F) = 0 \iff \sigma'(w_F) =_{F_2} 1$$

Evaluate w_F for all valuations "in parallel":

- Let $p_1, \ldots, p_m \in \mathbb{N}$ be pairwise coprime, $M = \prod p_i$, $M_i = M/p_i$
- $Y_i \mapsto (a\underbrace{t \cdots t}_{p_i})^{M_i} t^{-M} = \underbrace{(a, \underbrace{1, \ldots, 1}_{p_i - 1}, \ldots, a, \underbrace{1, \ldots, 1}_{p_i - 1})}_{M_i \text{ times}}$

  $\rightsquigarrow$ a at positions $\equiv 0 \mod p_i$

  $\tilde{Y}_i \mapsto (t\underbrace{at \cdots at}_{p_i - 1})^{M_i} t^{-M} = (1, \underbrace{a, \ldots, a}_{p_i - 1}, \ldots, 1, \underbrace{a, \ldots, a}_{p_i - 1})$

  $\rightsquigarrow$ a at positions $\not\equiv 0 \mod p_i$
- By the Chinese Remainder Theorem, this tests all valuations.

Open Questions II

The proof for free groups should be generalizable to

- RAAGs (= graph groups),
- graph products,
- hyperbolic groups,
- HNN extensions and amalgamated products over finite subgroups.

Open Questions II

The proof for free groups should be generalizable to

- RAAGs (= graph groups),
- graph products,
- hyperbolic groups,
- HNN extensions and amalgamated products over finite subgroups.

Problem:

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y. If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

Open Questions II

The proof for free groups should be generalizable to

- RAAGs (= graph groups),
- graph products,
- hyperbolic groups,
- HNN extensions and amalgamated products over finite subgroups.

Problem:

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

is NOT true anymore!!

Open Questions II

The proof for free groups should be generalizable to

- RAAGs (= graph groups),
- graph products,
- hyperbolic groups,
- HNN extensions and amalgamated products over finite subgroups.

Problem:

Lemma

Let $p, q \in \Omega$ and v a factor of p^x and w a factor of q^y.
If $vw = 1$ in F and $|v| = |w| \geq |p| + |q| - 1$, then $p = q$.

is NOT true anymore!!

Example

Let $p = qa$ with $[q, a] = 1$, then q^x is a factor of p^x and cancels with q^{-x} but $p \neq q$!

$\rightsquigarrow$ need more restrictions on Ω

- What if we allow nested exponents:

$$\left(b^{13}\overline{a}\ \left((b\,a^{8}a)^{13}a^{-26}b^{-13}\right)^{12}\right)^{16}\left((\overline{b}\,\overline{a})^{13}a^{13}\right)^{20}$$

  - Conjecture: for constant nesting depth in $\mathrm{AC}^0(\mathrm{WP}(F_2))$ (same approach).
  - Not clear what happens for unbounded nesting depth:
    …is it P-complete? …or in $\mathrm{AC}^0(\mathrm{WP}(F_2))$?

Open Questions III

- What if we allow nested exponents:

$$\left(b^{13}\bar{a}\ ((b\,a^{8}a)^{13}a^{-26}b^{-13})^{12}\right)^{16}\ ((\bar{b}\,\bar{a})^{13}a^{13})^{20}$$

  - Conjecture: for constant nesting depth in $\mathrm{AC}^0(\mathrm{WP}(F_2))$ (same approach).
  - Not clear what happens for unbounded nesting depth:
    ... is it P-complete? ... or in $\mathrm{AC}^0(\mathrm{WP}(F_2))$?
- Complexity of $\textsc{PowerWP}(G \wr \mathbb{Z})$ for G non-abelian, but not free nor finite, non-solvable (e. g. G nilpotent)?

Open Questions III

- What if we allow nested exponents:

$$\left(b^{13}\bar{a}\;((b\,a^{8}a)^{13}a^{-26}b^{-13})^{12}\right)^{16}((\bar{b}\,\bar{a})^{13}a^{13})^{20}$$

  - Conjecture: for constant nesting depth in $\mathsf{AC}^0(\mathrm{WP}(F_2))$ (same approach).
  - Not clear what happens for unbounded nesting depth:
    ...is it P-complete? ...or in $\mathsf{AC}^0(\mathrm{WP}(F_2))$?
- Complexity of $\textsc{PowerWP}(G \wr \mathbb{Z})$ for G non-abelian, but not free nor finite, non-solvable (e. g. G nilpotent)?
- Complexity of $\textsc{PowerWP}$ in other groups:
  - Grigochuk group – what is the maximal order of an element of length n?
  - other automaton groups?
  - Baumslag-Solitar groups?

- What if we allow nested exponents:

$$\left(b^{13}\overline{a}\ \left((b\,a^{8}a)^{13}a^{-26}b^{-13}\right)^{12}\right)^{16}\left((\overline{b}\,\overline{a})^{13}a^{13}\right)^{20}$$

  - Conjecture: for constant nesting depth in $\mathrm{AC}^0(\mathrm{WP}(F_2))$ (same approach).
  - Not clear what happens for unbounded nesting depth:
    ...is it P-complete? ...or in $\mathrm{AC}^0(\mathrm{WP}(F_2))$?
- Complexity of $\textsc{PowerWP}(G \wr \mathbb{Z})$ for G non-abelian, but not free nor finite, non-solvable (e. g. G nilpotent)?
- Complexity of $\textsc{PowerWP}$ in other groups:
  - Grigochuk group – what is the maximal order of an element of length n?
  - other automaton groups?
  - Baumslag-Solitar groups?

Thank you!