Finding all solutions of equations in free groups

Volker Diekert Artur Jeż Wojciech Plandowski

Les Diablerets, 08.03.2016

Equations in Free Groups

Definition (Free group)

Free group generated by Γ

- words over $\Gamma \cup \Gamma^{-1}$
- no cancellation except trivial one ($aa^{-1} = 1$)

Wlog $\Gamma = \Gamma^{-1}$.

Equations in Free Groups

Definition (Free group)

Free group generated by Γ

- words over $\Gamma \cup \Gamma^{-1}$
- no cancellation except trivial one ($aa^{-1} = 1$)

Wlog $\Gamma = \Gamma^{-1}$.

Definition (Word equations in Free group)

Given equations $U_i = V_i$, where $U_i, V_i \in (\Gamma \cup \mathcal{X})^*$.
Is there an assignment $S : \mathcal{X} \mapsto \Gamma^*$ satisfying the equations in a free group?

Equations in Free Semigroups

Definition

Free semigroup generated by Γ

- words over Γ
- no cancellation at all

Definition (Word equation in a Free Semigroup)

Given equation $U = V$, where $U, V \in (\Gamma \cup \mathcal{X})^*$.
Is there an assignment $S : \mathcal{X} \mapsto \Gamma^*$ satisfying the equation in a free semigroup?

Makanin's algorithm

Makanin 1977

Decidability in free semigroups: Rewriting procedure for free semigroups.

Makanin's algorithm

Makanin 1977

Decidability in free semigroups: Rewriting procedure for free semigroups.

Makanin 1985

Extension to free groups.

Makanin's algorithm

Makanin 1977

Decidability in free semigroups: Rewriting procedure for free semigroups.

Makanin 1985

Extension to free groups.

Only satisfiability, not all solutions.

Representation of all solutions

Razborov 1987

Representation of all solutions (free group).
(Makanin-Razborov diagrams)

Representation of all solutions

Razborov 1987

Representation of all solutions (free group).
(Makanin-Razborov diagrams)

This is important

First step in Kharlampovich & Myasnikov proof of Tarski's Conjecture

Progress in semigroups

Schulz 1990

Regular constraints for free semigroup.

Progress in semigroups

Schulz 1990

Regular constraints for free semigroup.

- description of regular languages $R_1, R_2, \ldots$ as part of the input
- conditions $X \in R_i$ or $X \notin R_i$

Description: by a monoid M:

- homomorphism ρ from Γ to M
- $\rho(S(X))$ is fixed for each X

Involution and free groups

Diekert, Gutiérrez, Hagenah 2001

Solving equations in free groups (with regular constraints)
reduces to
Solving equations in free semigroups with regular constraints and involution

Involution and free groups

Diekert, Gutiérrez, Hagenah 2001

Solving equations in free groups (with regular constraints)
reduces to
Solving equations in free semigroups with regular constraints and involution

- purely syntactical
- bijection on the solutions (in case of free groups: reduced)

In groups

- triangulate the equation
- remove cancellation: $XY = Z \to X = X'R,\ Y = R^{-1}Y',\ X'Y' = Z$
- look only at reduced solutions

In groups

- triangulate the equation
- remove cancellation: $XY = Z \to X = X'R,\ Y = R^{-1}Y',\ X'Y' = Z$
- look only at reduced solutions

Definition (Involution)

Think of it as the inverse in a free group.
Bijection $\bar{\cdot}$ on Γ^* and on $\mathcal{X}$ such that

- $\overline{\overline{a}} = a$
- $\overline{a_1 a_2 \cdots a_k} = \overline{a_k} \cdots \overline{a_2}\,\overline{a_1}$

In groups

- triangulate the equation
- remove cancellation: $XY = Z \to X = X'R,\ Y = R^{-1}Y',\ X'Y' = Z$
- look only at reduced solutions

Definition (Involution)

Think of it as the inverse in a free group.
Bijection $\bar{\cdot}$ on Γ^* and on $\mathcal{X}$ such that

- $\overline{\overline{a}} = a$
- $\overline{a_1 a_2 \cdots a_k} = \overline{a_k} \cdots \overline{a_2}\, \overline{a_1}$

Regular constraints

Enforce that everything is reduced (no aa^{-1})

Compression and word equations in semigroups

Plandowski & Rytter 1998

Length minimal solution of length N is compressible into $\mathrm{poly}(\log N)$.

Compression and word equations in semigroups

Plandowski & Rytter 1998

Length minimal solution of length N is compressible into poly($\log N$).

Plandowski

- PSPACE algorithm [1999]
- generation of all solutions in PSPACE [2006]

Compression and word equations in semigroups

Plandowski & Rytter 1998

Length minimal solution of length N is compressible into poly($\log N$).

Plandowski

- PSPACE algorithm [1999]
- generation of all solutions in PSPACE [2006]

J. 2013

The same, but simpler.

Generation of all solutions?

Adding involution and regular constraints to Plandowski's construction:

- works for satisfiability [Diekert, Gutiérrez, Hagenah 2001]
- does not work for generation of all solutions (problems with involution)

Generation of all solutions?

Adding involution and regular constraints to Plandowski's construction:

- works for satisfiability [Diekert, Gutiérrez, Hagenah 2001]
- does not work for generation of all solutions (problems with involution)

We do this for Jeż's solution.

Result

Representation of all solutions

- Graph (exponential size)
- Nodes: equations of size length $\mathcal{O}(n)$, at most n occurrences of variables
- Edges between nodes, labelled with (families of) morphisms

Result

Representation of all solutions

- Graph (exponential size)
- Nodes: equations of size length $\mathcal{O}(n)$, at most n occurrences of variables
- Edges between nodes, labelled with (families of) morphisms

Preserving solutions

If S is a solution of $U = V$ then

- the equation is trivial or
- there is ϕ-labelled edge to an equation $U' = V'$ with a solution S' such that $S = \phi(S')$

Result

Representation of all solutions

- Graph (exponential size)
- Nodes: equations of size length $\mathcal{O}(n)$, at most n occurrences of variables
- Edges between nodes, labelled with (families of) morphisms

Preserving solutions

If S is a solution of $U = V$ then

- the equation is trivial or
- there is ϕ-labelled edge to an equation $U' = V'$ with a solution S' such that $S = \phi(S')$

If $U' = V'$ has a solution S' and there is ϕ-labelled edge from $U = V$ then $\phi(S')$ is a solution of $U = V$.

Result

Representation of all solutions

- Graph (exponential size)
- Nodes: equations of size length $\mathcal{O}(n)$, at most n occurrences of variables
- Edges between nodes, labelled with (families of) morphisms

Preserving solutions

If S is a solution of $U = V$ then

- the equation is trivial or
- there is ϕ-labelled edge to an equation $U' = V'$ with a solution S' such that $S = \phi(S')$

If $U' = V'$ has a solution S' and there is ϕ-labelled edge from $U = V$ then $\phi(S')$ is a solution of $U = V$.

Each solution is obtained on a path from original equation to a trivial one.

Equality and Compression of Words

$$a\ a\ a\ b\ a\ b\ c\ a\ b\ a\ b\ b\ a\ b\ c\ b\ a$$

$$a\ a\ a\ b\ a\ b\ c\ a\ b\ a\ b\ b\ a\ b\ c\ b\ a$$

Equality and Compression of Words

$$a\ a\ a\ b\ a\ b\ c\ a\ b\ a\ b\ b\ a\ b\ c\ b\ a$$

$$a\ a\ a\ b\ a\ b\ c\ a\ b\ a\ b\ b\ a\ b\ c\ b\ a$$

Equality and Compression of Words

$$a_3 \quad b\ a\ b\ c\ a\ b\ a\ b\ b\ a\ b\ c\ b\ a$$

$$a_3 \quad b\ a\ b\ c\ a\ b\ a\ b\ b\ a\ b\ c\ b\ a$$

Equality and Compression of Words

$$a_3 \quad b \; a \; b \; c \; a \; b \; a \quad b_2 \; a \; b \; c \; b \; a$$

$$a_3 \quad b \; a \; b \; c \; a \; b \; a \quad b_2 \; a \; b \; c \; b \; a$$

Equality and Compression of Words

$$a_3 \quad b \quad d \quad c \quad d \quad a \quad b_2 \quad d \quad c \; b \; a$$

$$a_3 \quad b \quad d \quad c \quad d \quad a \quad b_2 \quad d \quad c \; b \; a$$

Equality and Compression of Words

$$a_3 \quad b \quad \color{green}{d} \quad c \quad \color{green}{d} \quad a \quad \color{blue}{b_2} \quad \color{green}{d} \quad c \quad \color{red}{e}$$

$$a_3 \quad b \quad \color{green}{d} \quad c \quad \color{green}{d} \quad a \quad \color{blue}{b_2} \quad \color{green}{d} \quad c \quad \color{red}{e}$$

Equality and Compression of Words

$$a_3 \quad b \quad d \quad c \quad d \quad a \quad b_2 \quad d \quad c \quad e$$

$$a_3 \quad b \quad d \quad c \quad d \quad a \quad b_2 \quad d \quad c \quad e$$

Iterate!

Regular constraints

- When c replaces w then $\rho(c) \leftarrow \rho(w)$.

Involution

When we replace ab with c, we should also replace $\overline{b}\overline{a}$ by $\overline{c}$.

Involution

When we replace ab with c, we should also replace $\overline{b}\overline{a}$ by $\overline{c}$.

What if they overlap?

Involution

When we replace ab with c, we should also replace $\overline{b}\overline{a}$ by $\overline{c}$.

What if they overlap?

Can happen only when $a = \overline{a}$ or $b = \overline{b}$.

Involution

When we replace ab with c, we should also replace $\overline{b}\overline{a}$ by $\overline{c}$.

What if they overlap?

Can happen only when $a = \overline{a}$ or $b = \overline{b}$.

Replace maximal such overlapping factors.

ab-blocks

$a = b$

a^i for $i \geq 2$ and $\overline{a}^i$ for $i \geq 2$

ab-blocks

$a = b$

a^i for $i \geq 2$ and $\overline{a}^i$ for $i \geq 2$

$a \neq b$

$\overline{a} \neq a$, $\overline{b} \neq b$ ab and $\overline{ab} = \overline{b}\overline{a}$

ab-blocks

$a = b$

a^i for $i \geq 2$ and $\overline{a}^i$ for $i \geq 2$

$a \neq b$

$\overline{a} \neq a$, $\overline{b} \neq b$ ab and $\overline{ab} = \overline{b}\overline{a}$

$\overline{a} = a$, $\overline{b} \neq b$ ab, $\overline{ab} = \overline{b}a$ and $\overline{b}ab$

ab-blocks

$a = b$

a^i for $i \geq 2$ and $\overline{a}^i$ for $i \geq 2$

$a \neq b$

$\overline{a} \neq a$, $\overline{b} \neq b$ ab and $\overline{ab} = \overline{b}\,\overline{a}$

$\overline{a} = a$, $\overline{b} \neq b$ ab, $\overline{ab} = \overline{b}a$ and $\overline{b}ab$

$\overline{a} \neq a$, $\overline{b} = b$ ab, $\overline{ab} = b\overline{a}$ and $ab\overline{a}$

ab-blocks

$a = b$

a^i for $i \geq 2$ and $\overline{a}^i$ for $i \geq 2$

$a \neq b$

$\overline{a} \neq a$, $\overline{b} \neq b$ ab and $\overline{ab} = \overline{b}\overline{a}$

$\overline{a} = a$, $\overline{b} \neq b$ ab, $\overline{ab} = \overline{b}a$ and $\overline{b}ab$

$\overline{a} \neq a$, $\overline{b} = b$ ab, $\overline{ab} = b\overline{a}$ and $ab\overline{a}$

$\overline{a} = a$, $\overline{b} = b$ $(ba)^i$, $a(ba)^i$, $(ba)^i b$ and $(ab)^i$ $(i \geq 1)$

ab-blocks

$a = b$

$$a^i \text{ for } i \geq 2 \text{ and } \overline{a}^i \text{ for } i \geq 2$$

$a \neq b$

$\overline{a} \neq a$, $\overline{b} \neq b$ ab and $\overline{ab} = \overline{b}\,\overline{a}$

$\overline{a} = a$, $\overline{b} \neq b$ ab, $\overline{ab} = \overline{b}a$ and $\overline{b}ab$

$\overline{a} \neq a$, $\overline{b} = b$ ab, $\overline{ab} = b\overline{a}$ and $ab\overline{a}$

$\overline{a} = a$, $\overline{b} = b$ $(ba)^i$, $a(ba)^i$, $(ba)^i b$ and $(ab)^i$ $(i \geq 1)$

Lemma

Maximal ab-factors do not overlap.

Compress in parallel the maximal ab-blocks.

Block types

Definition (Factor and block types)

Factor ab is

- **explicit** it comes from U or V;
- **implicit** comes solely from $S(X)$;
- **crossing** in other case.

ab is crossing if ab or $\overline{ab}$ have a crossing occurrence,
non-crossing otherwise.

Block types

Definition (Factor and block types)

Factor ab is

explicit it comes from U or V;

implicit comes solely from $S(X)$;

crossing in other case.

ab is crossing if ab or $\overline{ab}$ have a crossing occurrence,
non-crossing otherwise.

Alphabet: $a = \overline{a}, b = \overline{b}, c \neq \overline{c}$
Equation: $baXbc\overline{c}ab\overline{X}ab = babac\overline{c}abc\overline{c}abac\overline{c}abab$
Solution: $S(X) = bac\overline{c}a$

- $babac\overline{c}abc\overline{c}abac\overline{c}abab$ $[baXbc\overline{c}ab\overline{X}ab]$
- $babac\overline{c}abc\overline{c}abac\overline{c}abab$ $[baXbc\overline{c}ab\overline{X}ab]$
- $babac\overline{c}abc\overline{c}abac\overline{c}abab$ $[baXbc\overline{c}ab\overline{X}ab]$

Compression of non-crossing *ab*-blocks

When *ab* is non-crossing.

Compression of non-crossing ab-blocks

When ab is non-crossing.

CompNCr

1: let $f_1, f_2, \ldots, f_k$ be different ab-blocks in the equation
2: replace each explicit factor f_i in U and V by c_{f_i}

$$f_i = \overline{f_j} \iff c_i = \overline{c_j}$$

3: set $\rho(c_i) = \rho(f_i)$

Lemma

When ab is non-crossing for S then $\mathrm{CompNCr}(a, b)$ *returns an equation with a corresponding solution S'.*

Lemma

When ab is non-crossing for S then CompNCr(a, b) *returns an equation with a corresponding solution S'.*

Consider an occurrence of ab-block

explicit replaced explicitly

implicit we change S to S'

crossing there is none

Lemma

When ab is non-crossing for S then CompNCr(a, b) *returns an equation with a corresponding solution* S'.

Consider an occurrence of ab-block

- explicit: replaced explicitly
- implicit: we change S to S'
- crossing: there is none

$baXbc\overline{c}ab\overline{X}ab = babac\overline{c}abc\overline{c}abac\overline{c}abab$
$S(X) = bac\overline{c}a$

Lemma

When ab is non-crossing for S then CompNCr(a, b) *returns an equation with a corresponding solution S'.*

Consider an occurrence of ab-block

- **explicit** replaced explicitly
- **implicit** we change S to S'
- **crossing** there is none

$$baXbc\overline{c}ab\overline{X}ab = babac\overline{c}abc\overline{c}abac\overline{c}abab$$
$$S(X) = bac\overline{c}a$$
$$baXbdab\overline{X}ab = babadabdabadabab$$
$$S(X) = bada$$

Crossing ab-blocks

Most difficult

Consider $a = \overline{a}$, $b = \overline{b}$.
ab-blocks: $(ab)^i, b(ab)^i, (ab)^i a, (ba)^i$

Crossing ab-blocks

Most difficult

Consider $a = \overline{a}$, $b = \overline{b}$.
ab-blocks: $(ab)^i, b(ab)^i, (ab)^i a, (ba)^i$

ab is crossing

- There is ab-factor partially in X and partially outside.

Crossing ab-blocks

Most difficult

Consider $a = \overline{a}$, $b = \overline{b}$.
ab-blocks: $(ab)^i, b(ab)^i, (ab)^i a, (ba)^i$

ab is crossing

- There is ab-factor partially in X and partially outside.
- What is the part inside?

Crossing *ab*-blocks

Most difficult

Consider $a = \overline{a}$, $b = \overline{b}$.
ab-blocks: $(ab)^i, b(ab)^i, (ab)^i a, (ba)^i$

ab is crossing

- There is *ab*-factor partially in X and partially outside.
- What is the part inside?

Definition

ab-prefix of $S(X)$ a, b or *ab*-block
ab-suffix of $S(X)$ a, b or *ab*-block

Dealing with crossing *ab*

f is the ab-prefix of $S(X)$

- replace X with fX and $\overline{X}$ by $\overline{X}\,\overline{f}$
  (implicitly change solution $S(X) = fw$ to $S(X) = w$
  $S(\overline{X}) = \overline{w}\overline{f}$ to $S(\overline{X}) = \overline{w}$)

Dealing with crossing *ab*

f is the *ab*-prefix of $S(X)$

- replace X with fX and $\overline{X}$ by $\overline{X}\,\overline{f}$
  (implicitly change solution $S(X) = fw$ to $S(X) = w$
  $S(\overline{X}) = \overline{w}\overline{f}$ to $S(\overline{X}) = \overline{w}$)
- change ρ_X, $\rho_{\overline{X}}$ so that : $\rho_X = \rho(f)\rho'_X$, $\rho_{\overline{X}} = \rho_{\overline{X'}}\rho(\overline{f})$

Dealing with crossing ab

f is the ab-prefix of $S(X)$

- replace X with fX and $\overline{X}$ by $\overline{X}\,\overline{f}$
  (implicitly change solution $S(X) = fw$ to $S(X) = w$
  $S(\overline{X}) = \overline{w}\overline{f}$ to $S(\overline{X}) = \overline{w}$)
- change ρ_X, $\rho_{\overline{X}}$ so that : $\rho_X = \rho(f)\rho'_X$, $\rho_{\overline{X}} = \rho_{\overline{X'}}\rho(\overline{f})$
- if $S(X) = 1$ and $\rho_X = \rho(1)$ then remove X.

Dealing with crossing *ab*

f is the *ab*-prefix of $S(X)$

- replace X with fX and $\overline{X}$ by $\overline{X}\,\overline{f}$
  (implicitly change solution $S(X) = fw$ to $S(X) = w$
  $S(\overline{X}) = \overline{w}\overline{f}$ to $S(\overline{X}) = \overline{w}$)
- change ρ_X, $\rho_{\overline{X}}$ so that : $\rho_X = \rho(f)\rho'_X$, $\rho_{\overline{X}} = \rho_{\overline{X'}}\rho(\overline{f})$
- if $S(X) = 1$ and $\rho_X = \rho(1)$ then remove X.

Lemma

Uncrossing preserves solutions

Dealing with crossing *ab*

f is the *ab*-prefix of $S(X)$

- replace X with fX and $\overline{X}$ by $\overline{X}\,\overline{f}$ (implicitly change solution $S(X) = fw$ to $S(X) = w$ $S(\overline{X}) = \overline{w}\overline{f}$ to $S(\overline{X}) = \overline{w}$)
- change ρ_X, $\rho_{\overline{X}}$ so that : $\rho_X = \rho(f)\rho'_X$, $\rho_{\overline{X}} = \rho_{\overline{X'}}\rho(\overline{f})$
- if $S(X) = 1$ and $\rho_X = \rho(1)$ then remove X.

Lemma

Uncrossing preserves solutions

Lemma

After performing this for all variables, ab is no longer crossing.

Dealing with crossing ab

f is the ab-prefix of $S(X)$

- replace X with fX and $\overline{X}$ by $\overline{X}\,\overline{f}$
  (implicitly change solution $S(X) = fw$ to $S(X) = w$
  $S(\overline{X}) = \overline{w}\overline{f}$ to $S(\overline{X}) = \overline{w}$)
- change ρ_X, $\rho_{\overline{X}}$ so that : $\rho_X = \rho(f)\rho'_X$, $\rho_{\overline{X}} = \rho_{\overline{X'}}\rho(\overline{f})$
- if $S(X) = 1$ and $\rho_X = \rho(1)$ then remove X.

Lemma

Uncrossing preserves solutions

Lemma

After performing this for all variables, ab is no longer crossing.

Compress the ab-blocks!

Alphabet: $a = \overline{a}, b = \overline{b}, c \neq \overline{c}$
Equation: $baXbc\overline{c}ab\overline{X}ab = babac\overline{c}abc\overline{c}abac\overline{c}abab$
Solution: $S(X) = {\color{red}ba}c\overline{c}{\color{red}a}$

Alphabet: $a = \overline{a}, b = \overline{b}, c \neq \overline{c}$
Equation: $baXbc\overline{c}ab\overline{X}ab = babac\overline{c}abc\overline{c}abac\overline{c}abab$
Solution: $S(X) = bac\overline{c}a$
Equation: $babaXabc\overline{c}aba\overline{X}abab = babac\overline{c}abc\overline{c}abac\overline{c}abab$
Solution: $S'(X) = c\overline{c}$

Algorithm

while $U \notin \Gamma$ and $V \notin \Gamma$ **do**
  choose ab from the equation
  uncross the ab-blocks ▷ If needed
  compress ab-blocks

Algorithm

while $U \notin \Gamma$ and $V \notin \Gamma$ **do**
choose ab from the equation
uncross the ab-blocks ▷ If needed
compress ab-blocks

Theorem

This preserves solution.
This shortens the solution.

Crucial property

Theorem (Main property: keeps the equation short)

For any solution there is always ab such that after ab-compression the equation has size $\mathcal{O}(n^2)$.

Crucial property

Theorem (Main property: keeps the equation short)

For any solution there is always ab such that after ab-compression the equation has size $\mathcal{O}(n^2)$.

Proof.

Take $c = 18$

- if there is non-crossing ab: do it!

Crucial property

Theorem (Main property: keeps the equation short)

For any solution there is always ab such that after ab-compression the equation has size $\mathcal{O}(n^2)$.

Proof.

Take $c = 18$

- if there is non-crossing ab: do it!
- each crossing ab-compression: $+2n$ letters

Crucial property

Theorem (Main property: keeps the equation short)

For any solution there is always ab such that after ab-compression the equation has size $\mathcal{O}(n^2)$.

Proof.

Take $c = 18$

- if there is non-crossing ab: do it!
- each crossing ab-compression: $+2n$ letters
- if size of the equation $\leq 18n^2 - 2n$: do it!

Crucial property

Theorem (Main property: keeps the equation short)

For any solution there is always ab such that after ab-compression the equation has size $\mathcal{O}(n^2)$.

Proof.

Take $c = 18$

- if there is non-crossing ab: do it!
- each crossing ab-compression: $+2n$ letters
- if size of the equation $\leq 18n^2 - 2n$: do it!
- there are $\leq 4n$ crossing ab

Crucial property

Theorem (Main property: keeps the equation short)

For any solution there is always ab such that after ab-compression the equation has size $\mathcal{O}(n^2)$.

Proof.

Take $c = 18$

- if there is non-crossing ab: do it!
- each crossing ab-compression: $+2n$ letters
- if size of the equation $\leq 18n^2 - 2n$: do it!
- there are $\leq 4n$ crossing ab
- so some covers $\geq (18n^2 - 2n)/4n \geq 4n$ letters

Crucial property

Theorem (Main property: keeps the equation short)

For any solution there is always ab such that after ab-compression the equation has size $\mathcal{O}(n^2)$.

Proof.

Take $c = 18$

- if there is non-crossing ab: do it!
- each crossing ab-compression: $+2n$ letters
- if size of the equation $\leq 18n^2 - 2n$: do it!
- there are $\leq 4n$ crossing ab
- so some covers $\geq (18n^2 - 2n)/4n \geq 4n$ letters
- so $4n/2 = 2n$ letters are removed

Crucial property

Theorem (Main property: keeps the equation short)

For any solution there is always ab such that after ab-compression the equation has size $\mathcal{O}(n^2)$.

Proof.

Take $c = 18$

- if there is non-crossing ab: do it!
- each crossing ab-compression: $+2n$ letters
- if size of the equation $\leq 18n^2 - 2n$: do it!
- there are $\leq 4n$ crossing ab
- so some covers $\geq (18n^2 - 2n)/4n \geq 4n$ letters
- so $4n/2 = 2n$ letters are removed
- the equation does not grow

□

Satisfiability

Corollary

Satisfiability of word equations in free semigroups with regular constraints and involution is in PSPACE*.*

Satisfiability

Corollary

Satisfiability of word equations in free semigroups with regular constraints and involution is in PSPACE*.*

Proof.

Just guess so that the equation stays polynomial. □

Satisfiability

Corollary

Satisfiability of word equations in free semigroups with regular constraints and involution is in PSPACE*.*

Proof.

Just guess so that the equation stays polynomial. □

Corollary

The graph representing all solutions of word equations in free semigroups with regular constraints and involution can be constructed in PSPACE*.*

Satisfiability

Corollary

Satisfiability of word equations in free semigroups with regular constraints and involution is in PSPACE*.*

Proof.

Just guess so that the equation stays polynomial. □

Corollary

The graph representing all solutions of word equations in free semigroups with regular constraints and involution can be constructed in PSPACE*.*

Proof.

- produce all nodes
- verify, whether we can go from an equation to an equation □

Satisfiability

Corollary

Satisfiability of word equations in free semigroups with regular constraints and involution is in PSPACE*.*

Proof.

Just guess so that the equation stays polynomial. □

Corollary

The graph representing all solutions of word equations in free semigroups with regular constraints and involution can be constructed in PSPACE*.*

Proof.

- produce all nodes
- verify, whether we can go from an equation to an equation □

Almost

What about lengths of *ab*-blocks?

Long *ab*-blocks

Idea

- *ab* factors can be arbitrarily long

Long *ab*-blocks

Idea

- *ab* factors can be arbitrarily long
- when we replace *ab*-blocks, only equality matters, not length
- pop $(ab)^{\ell_X}$ and $(ba)^{r_X}$ from X but treat ℓ_X, r_X as integer variables

Long ab-blocks

Idea

- ab factors can be arbitrarily long
- when we replace ab-blocks, only equality matters, not length
- pop $(ab)^{\ell_X}$ and $(ba)^{r_X}$ from X but treat ℓ_X, r_X as integer variables
- guess the equal blocks
- check if they can be equal
- replace them

Long *ab*-blocks

Idea

- *ab* factors can be arbitrarily long
- when we replace *ab*-blocks, only equality matters, not length
- pop $(ab)^{\ell_X}$ and $(ba)^{r_X}$ from X but treat ℓ_X, r_X as integer variables
- guess the equal blocks
- check if they can be equal
- replace them

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$
- $3\ell_X = 2\ell_Y$

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$
- $3\ell_X = 2\ell_Y$
- turn to $c_{f_1} c_{f_2} = c_{f_1} c_{f_2}$

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$
- $3\ell_X = 2\ell_Y$
- turn to $c_{f_1}c_{f_2} = c_{f_1}c_{f_2}$
- Lengths: linear combination of $\{\ell_X, r_X\}_{X\in\mathcal{X}}$ and constants.

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$
- $3\ell_X = 2\ell_Y$
- turn to $c_{f_1}c_{f_2} = c_{f_1}c_{f_2}$
- Lengths: linear combination of $\{\ell_X, r_X\}_{X\in\mathcal{X}}$ and constants.
- Equality: equation in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$
- $3\ell_X = 2\ell_Y$
- turn to $c_{f_1}c_{f_2} = c_{f_1}c_{f_2}$
- Lengths: linear combination of $\{\ell_X, r_X\}_{X\in\mathcal{X}}$ and constants.
- Equality: equation in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

Verification

Guessed equalities $\iff$ linear Diophantine equations in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$
- $3\ell_X = 2\ell_Y$
- turn to $c_{f_1}c_{f_2} = c_{f_1}c_{f_2}$
- Lengths: linear combination of $\{\ell_X, r_X\}_{X\in\mathcal{X}}$ and constants.
- Equality: equation in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

Verification

Guessed equalities $\iff$ linear Diophantine equations in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

- has size proportional to equation

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$
- $3\ell_X = 2\ell_Y$
- turn to $c_{f_1}c_{f_2} = c_{f_1}c_{f_2}$
- Lengths: linear combination of $\{\ell_X, r_X\}_{X\in\mathcal{X}}$ and constants.
- Equality: equation in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

Verification

Guessed equalities $\iff$ linear Diophantine equations in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

- has size proportional to equation
- can be verified in polynomial space

Lengths of *ab*-blocks

- $aXbXXX = \overline{X}abYY$
- $S(X) = (ba)^{\ell_X}, S(Y) = (ba)^{\ell_Y}$
- $(ab)^{\ell_X+1}(ba)^{3\ell_X} = (ab)^{\ell_X+1}(ba)^{2\ell_Y}$
- $3\ell_X = 2\ell_Y$
- turn to $c_{f_1}c_{f_2} = c_{f_1}c_{f_2}$
- Lengths: linear combination of $\{\ell_X, r_X\}_{X\in\mathcal{X}}$ and constants.
- Equality: equation in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

Verification

Guessed equalities $\iff$ linear Diophantine equations in $\{\ell_X, r_X\}_{X\in\mathcal{X}}$

- has size proportional to equation
- can be verified in polynomial space
- one solution: one morphism

Diophantine equations and graph representation

- Edges can be labelled with (polynomial-size) system of linear Diophantine equations.
- Each solution gives one morphism.

Open questions, related research, etc.

Open questions

- Is it more general?
- To what groups it generalises?