On Recompression for Word Equations

Artur Jeż

Meeting on String Constraints and Applications (MOSCA)
07.05.2019
Definition (Satisfiability of word equations)

Given equation $U = V$, where $U, V \in (\Sigma \cup \mathcal{X})^*$. Is there a substitution $S : \mathcal{X} \rightarrow \Sigma^*$ satisfying the equation?

(Also more general: finitely many solutions, representation of all, ... )
Word Equations

Definition (Satisfiability of word equations)

Given equation $U = V$, where $U, V \in (\Sigma \cup X)^*$. Is there a substitution $S : X \to \Sigma^*$ satisfying the equation? (Also more general: finitely many solutions, representation of all, ...)

$$a X b X Y bbb = X abaa Y bY \quad S(X) = aa, S(Y) = bb$$
Definition (Satisfiability of word equations)

Given equation $U = V$, where $U, V \in (\Sigma \cup \mathcal{X})^*$. Is there a substitution $S : \mathcal{X} \rightarrow \Sigma^*$ satisfying the equation? (Also more general: finitely many solutions, representation of all, ...)

$$aXbXYbbb = XabaaYbY \quad S(X) = aa, S(Y) = bb$$

$$aaaabaaabbb = aaaabaaabbb$$
Word Equations

Definition (Satisfiability of word equations)

Given equation $U = V$, where $U, V \in (\Sigma \cup \mathcal{X})^*$. Is there a substitution $S : \mathcal{X} \to \Sigma^*$ satisfying the equation? (Also more general: finitely many solutions, representation of all, ...)

We extend $S$ to a $S : (\Sigma \cup \mathcal{X})^* \to \Sigma^*$; identity on $\Sigma$. $S(U)$ is a solution word. Length-minimal $S$: minimises $|S(U)|$. Usually: no $S(X) = \epsilon$, i.e. $S : \mathcal{X} \to \Sigma^+$. 
Makanin’s algorithm 1977

High complexity [EXPSPACE ’98], difficult proof.
### Makanin’s algorithm 1977

High complexity [EXPSPACE ’98], difficult proof.

### Compression and word equations

Length minimal solution (length $N$): compressible to poly($\log N$). 2NEXPTIME [Plandowski and Rytter, 1998]

The size $N$ of the minimal solution is at most doubly exponential. NEXPTIME [Plandowski 1999]

PSPACE [Plandowski 1999]

The same, but simpler. [J. 2013]

Only NP-hard. And believed to be in NP.

Solutions at most exponential?
<table>
<thead>
<tr>
<th><strong>Makanin’s algorithm 1977</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>High complexity [EXPSPACE ’98], difficult proof.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Compression and word equations</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Length minimal solution (length $N$): compressible to $\text{poly}(\log N)$. $2\text{NEXPTIME}$ [Plandowski and Rytter, 1998]</td>
</tr>
</tbody>
</table>
Makanin’s algorithm 1977
High complexity [EXPSPACE ’98], difficult proof.

Compression and word equations
- Length minimal solution (length $N$): compressible to poly($\log N$). \textbf{2NEXPTIME} [Plandowski and Rytter, 1998]
- The size $N$ of the minimal solution is at most doubly exponential. \textbf{NEXPTIME} [Plandowski 1999]
Makanin’s algorithm 1977

High complexity [EXPSPACE ’98], difficult proof.

Compression and word equations

- Length minimal solution (length $N$): compressible to poly$(\log N)$. \textbf{2NEXPTIME} [Plandowski and Rytter, 1998]
- The size $N$ of the minimal solution is at most doubly exponential. \textbf{NEXPTIME} [Plandowski 1999]
- \textbf{PSPACE} [Plandowski 1999]
Makanin’s algorithm 1977

High complexity [EXPSPACE ’98], difficult proof.

Compression and word equations

- Length minimal solution (length $N$): compressible to poly($\log N$). $2\text{NEXPTIME}$ [Plandowski and Rytter, 1998]
- The size $N$ of the minimal solution is at most doubly exponential. $\text{NEXPTIME}$ [Plandowski 1999]
- $\text{PSPACE}$ [Plandowski 1999]
- The same, but simpler. [J. 2013]
Makanin’s algorithm 1977

High complexity [EXPSPACE ’98], difficult proof.

Compression and word equations

- Length minimal solution (length $N$): compressible to poly(log $N$). 2NEXPTIME [Plandowski and Rytter, 1998]
- The size $N$ of the minimal solution is at most doubly exponential. NEXPTIME [Plandowski 1999]
- PSPACE [Plandowski 1999]
- The same, but simpler. [J. 2013]

Only NP-hard. And believed to be in NP.
Solutions at most exponential?
Simple is good on its own.
Simple is good on its own.

Easier to generalize

- Regular constraints [Diekert, J., Plandowski]
- Involution ($aw = wa$) [Diekert, J., Plandowski]
- Free groups [Diekert, J., Plandowski]
- Generation of all solutions [J.]
  for free groups [Diekert, J., Plandowski]
- Partial commutation [Diekert, J., Kufleitner]
- All solutions are EDT0L language [Ciobanu, Diekert, Elder]
- Nondeterministic linear space = context sensitive language [J.]
- Twisted word equations (permutation of letters) [Diekert, Elder]
- Linear time for one variable [J.]
- Context unification (terms) [J.]
Intuition: recompression
▶ Think of new letters as nonterminals of a grammar
▶ We build a grammar for both strings, bottom-up.
▶ Everything is compressed in the same way!

Comparison with Plandowski’s approach
Top-down, creates many problems.
Intuition: recompression
▶ Think of new letters as nonterminals of a grammar
▶ We build a grammar for both strings, bottom-up.
▶ Everything is compressed in the same way!

Comparison with Plandowski’s approach
Top-down, creates many problems.
Equality and Compression of Strings

Intuition: recompresion

- Think of new letters as nonterminals of a grammar
- We build a grammar for both strings, bottom-up.
- Everything is compressed in the same way!

Comparison with Pladowski's approach

top-down, creates many problems.
Equality and Compression of Strings

Intuition: recompression
▶ Think of new letters as nonterminals of a grammar
▶ We build a grammar for both strings, bottom-up.
▶ Everything is compressed in the same way!

Comparison with Plandowski’s approach
Top-down creates many problems.
Intuition: recompression

Think of new letters as nonterminals of a grammar

We build a grammar for both strings, bottom-up.

Everything is compressed in the same way!

Comparison with Plandowski's approach

Top-down, creates many problems.
Intuition: recompression
▶ Think of new letters as nonterminals of a grammar
▶ We build a grammar for both strings, bottom-up.
▶ Everything is compressed in the same way!

Comparison with Plandowski’s approach
Top-down, creates many problems.
Equality and Compression of Strings

Intuition: recompression

- Think of new letters as nonterminals of a grammar
- We build a grammar for both strings, bottom-up.
- Everything is compressed in the same way!
Intuition: recompression

- Think of new letters as nonterminals of a grammar
- We build a grammar for both strings, bottom-up.
- Everything is compressed in the same way!

Comparison with Plandowski’s approach

Top-down, creates many problems.
For both solution words choose a pair (or letter) and compress it.
For both solution words choose a pair (or letter) and compress it.

while $U \notin \Sigma$ and $V \notin \Sigma$ do
  L ← letters from $S(U) = S(V)$
  for choose $ab \in L^2$ or $a \in L$ do
    replace all occurrences of $ab$ in $S(U)$ and $S(V)$
    (or replace all occurrences of blocks of $a$)
For both solution words choose a pair (or letter) and compress it.

```
while U \notin \Sigma \text{ and } V \notin \Sigma \text{ do }
L \leftarrow \text{letters from } S(U) = S(V)
for \text{ choose } ab \in L^2 \text{ or } a \in L \text{ do }
    \text{replace all occurrences of } ab \text{ in } S(U) \text{ and } S(V)
    \text{(or replace all occurrences of blocks of } a\text{)}
```

How to do this for equations?
**Working example**

*\(XbaYb = baaababbab\) has a solution \(S(X) = baaa, S(Y) = bba\)*
Working example

$XbaYb = baaaababbab$ has a solution $S(X) = baaa$, $S(Y) = bba$

We want to replace pair $ba$ by a new letter $c$. Then

$$XbaYb = baaaababbab \quad \text{for} \quad S(X) = baaa \quad S(Y) = bba$$
Working example

$X_{ba}Y_{b} = baaaababbab$ has a solution $S(X) = baaa$, $S(Y) = bba$

We want to replace pair $ba$ by a new letter $c$. Then

$X_{ba}Y_{b} = baaaababbab$ for $S(X) = baaa$, $S(Y) = bba$

$X\ c\ Y_{b} = c\ aa\ c\ b\ c\ b$ for $S'(X) = caa$, $S'(Y) = bc$
Working example

\[ XbaYb = baaaababbbab \] has a solution \( S(X) = baaa, S(Y) = bba \)

We want to replace pair \( ba \) by a new letter \( c \). Then

\[
\begin{align*}
XbaYb &= baaaababbbab \\
XcYb &= c\text{ }aa\text{ }c\text{ }b\text{ }c\text{ }b
\end{align*}
\]
for \( S(X) = baaa \) \( S(Y) = bba \)

And what about replacing \( ab \) by \( d \)?

\[
\begin{align*}
XbaYb &= baaaababbbab \\
&\text{for } S(X) = baaa \ S(Y) = bba
\end{align*}
\]
Working example

$XbaYb = baaababbab$ has a solution $S(X) = baaa$, $S(Y) = bba$

We want to replace pair $ba$ by a new letter $c$. Then

$XbaYb = baacababbab$ for $S(X) = baaa$ $S(Y) = bba$

$XcYb = cacaacbca$ for $S'(X) = caa$ $S'(Y) = bc$

And what about replacing $ab$ by $d$?

$XbaYb = baaababbbbab$ for $S(X) = baaa$ $S(Y) = bba$

There is a problem with ‘crossing pairs’. We will fix!
Definition (Pair types)

Occurrence of $ab$ in a solution word (so for a fixed solution) is

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

$ab$ is **crossing** if it has a crossing occurrence, non-crossing otherwise.
Definition (Pair types)

Occurrence of $ab$ in a solution word (so for a fixed solution) is
- **explicit** it comes from $U$ or $V$;
- **implicit** comes solely from $S(X)$;
- **crossing** in other case.

$ab$ is **crossing** if it has a crossing occurrence, non-crossing otherwise.

$$X = \text{baa} \quad Y = \text{bbaabaaabbab} \quad S(X) = \text{baa}a \quad S(Y) = \text{bba}$$
**Definition (Pair types)**

Occurrence of \( ab \) in a solution word (so for a fixed solution) is

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

\( ab \) is **crossing** if it has a crossing occurrence, non-crossing otherwise.

\[
X \quad baa \quad Y \quad b = baaabaabbab \\
S(X) = baaa, S(Y) = bba
\]

\[
baaa \quad baa \quad bba \quad b = baaabaabbab
\]

**explicit**

\[
baaa \quad baa \quad bba \quad b = baaabaabbab
\]

**implicit**

\[
baaa \quad baa \quad bba \quad b = baaabaabbab
\]

**crossing**

\[
baaa \quad baa \quad bba \quad b = baaabaabbab
\]
Compression of non-crossing pairs

**PairComp** \((a, b)\)

1: let \(c \in \Sigma\) be an unused letter
2: replace each explicit \(ab\) in \(U\) and \(V\) by \(c\)

**Lemma**
The \(\text{PairComp}(a, b)\) properly compresses noncrossing pairs.

complete if the old equation has a solution then the new one has
sound if the new equation has a solution then the old one has
Compression of non-crossing pairs

**PairComp**($a, b$)

1. let $c \in \Sigma$ be an unused letter
2. replace each explicit $ab$ in $U$ and $V$ by $c$

**Lemma**

*The PairComp*($a, b$) *properly compresses noncrossing pairs.*
Compression of non-crossing pairs

**PairComp** \((a, b)\)

1. let \(c \in \Sigma\) be an unused letter
2. replace each explicit \(ab\) in \(U\) and \(V\) by \(c\)

**Lemma**

The **PairComp** \((a, b)\) properly compresses noncrossing pairs.

**complete** if the old equation has a solution then the new one has
**sound** if the new equation has a solution then the old one has
Correctness.

Complete

$S'(U')$ is $S(U)$ with every $ab$ replaced; similarly $S'(V')$:

- **explicit pairs** replaced explicitly
- **implicit pairs** replaced implicitly (in the solution)
- **crossing** there are none
Correctness.

Complete

\[ S'(U') \text{ is } S(U) \text{ with every } ab \text{ replaced; similarly } S'(V') : \]

**explicit pairs** replaced explicitly

**implicit pairs** replaced implicitly (in the solution)

**crossing** there are none

\[ X \text{ baa Y b=}baaabaabbab \quad S(X) = baaa \quad S(Y) = bba \]

baaabaabbab = baaabaabbab
Correctness.

Complete

$S'(U')$ is $S(U)$ with every $ab$ replaced; similarly $S'(V')$:

- **explicit pairs** replaced explicitly
- **implicit pairs** replaced implicitly (in the solution)
- **crossing** there are none

\[
X \quad \text{baa} \quad Y \quad \text{b=baaabaabbab} \quad S(X) = \text{baaa} \quad S(Y) = \text{bba} \\
\text{baaabaabbab=baaabaabbab} \quad S'(X) = \text{caa} \quad S'(Y) = \text{bc}
\]
Correctness.

Complete

$S'(U')$ is $S(U)$ with every $ab$ replaced; similarly $S'(V')$:
- **explicit pairs** replaced explicitly
- **implicit pairs** replaced implicitly (in the solution)
  - **crossing** there are none

Sound

If the new equation is satisfiable: roll back the changes.
Correctness.

Complete

$S'(U')$ is $S(U)$ with every $ab$ replaced; similarly $S'(V')$:

- **explicit pairs** replaced explicitly
- **implicit pairs** replaced implicitly (in the solution)
- **crossing** there are none

Sound

If the new equation is satisfiable: roll back the changes.

\[
X \ baa \ Y \ b = baaabaabbab
\]
\[
c \ \text{aa} \ c \ \text{ab} \ c \ b = c \ \text{aa} \ c \ \text{ab} \ c \ b
\]
\[
X \ c \ a \ Y \ b = c \ \text{aa} \ c \ \text{ab} \ c \ b \quad S'(X) = c\text{aa} \\ S'(Y) = bc
\]
Correctness.

**Complete**

\( S'(U') \) is \( S(U) \) with every \( ab \) replaced; similarly \( S'(V') \):

- **explicit pairs** replaced explicitly
- **implicit pairs** replaced implicitly (in the solution)
- **crossing** there are none

**Sound**

If the new equation is satisfiable: roll back the changes.

\[
X \ baa \ Y \ b = baaabaabbab \quad S(X) = baaa \quad S(Y) = bba
\]

\[
\begin{align*}
  c \ aa & \quad c \ ab \ c \ b = c \ aa \ c \ ab \ c \ b \\
  X \ c \ a \ Y \ b & = c \ aa \ c \ ab \ c \ b \\
  S'(X) & = caa \quad S'(Y) = bc
\end{align*}
\]
Correctness.

Complete

$S'(U')$ is $S(U)$ with every $ab$ replaced; similarly $S'(V')$:

- **explicit pairs** replaced explicitly
- **implicit pairs** replaced implicitly (in the solution)
- **crossing** there are none

Sound

If the new equation is satisfiable: roll back the changes.

\[
X \ ba a \ Y \ b = baaaabaabbab \quad S(X) = baaa \quad S(Y) = bba \\
baaaabaabbab = baaaabaabbab \\
c aa \ c \ a b \ c \ b = c \ aa \ c \ ab \ c \ b \\
X \ c \ a \ Y \ b = c \ aa \ c \ ab \ c \ b \quad S'(X) = caa \quad S'(Y) = bc
\]
Dealing with crossing pairs

*ab* is a crossing pair
There is $X$ such that $S(X) = bw$ and $aX$ occurs in $U = V$ (or symmetric).
**Dealing with crossing pairs**

*ab* is a crossing pair

There is $X$ such that $S(X) = bw$ and $aX$ occurs in $U = V$ (or symmetric).

<table>
<thead>
<tr>
<th>Uncrossing($a, b$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1: for $X \in \mathcal{X}$ do</td>
</tr>
<tr>
<td>2: if first letter of $S(X)$ is $b$ then</td>
</tr>
<tr>
<td>3: replace each occurrence of $X$ by $bX$</td>
</tr>
<tr>
<td>4: if $S(X) = \epsilon$ then remove $X$ from the equation</td>
</tr>
<tr>
<td>5: perform symmetrically for the last letter and $a$</td>
</tr>
</tbody>
</table>

**Lemma**

After uncrossing *ab* is no longer crossing. We can compress it.
Dealing with crossing pairs

*ab* is a crossing pair

There is $X$ such that $S(X) = bw$ and $aX$ occurs in $U = V$ (or symmetric).

**Uncrossing** $(a, b)$

1: **for** $X \in X$ **do**
2: **if** first letter of $S(X)$ is $b$ **then**
3:   replace each occurrence of $X$ by $bX$ ▷ Pop ▷ Change $S$ accordingly
4: **if** $S(X) = \epsilon$ **then** remove $X$ from the equation
5:   ▷ perform symmetrically for the last letter and $a$

**Lemma**

*After uncrossing* $ab$ *is no longer crossing.*
Dealing with crossing pairs

*ab* is a crossing pair

There is *X* such that *S(X) = bw* and *aX* occurs in *U = V* (or symmetric).

Uncrossing(*a, b*)

1: for *X ∈ X* do
2: if first letter of *S(X) is b* then
3: replace each occurrence of *X* by *bX* ▶ Pop ▶ Change *S* accordingly
4: if *S(X) = ε* then remove *X* from the equation
5: ▶ perform symmetrically for the last letter and *a*

Lemma

*After uncrossing* *ab* is no longer crossing.

We can compress it.
Uncrossing: example

Uncrossing $ab$

$X \ baa \ Y \ b = baaabaababbab \quad S(X) = baaa \quad S(Y) = bba$
Uncrossing $ab$

$X = \text{baa} \quad Y = \text{bbaababaabbab} \quad S(X) = \text{baaa} \quad S(Y) = \text{bba}$

$\text{baaabaaa bba b} = \text{baaabaabbab}$
Uncrossing $ab$

$X$ baa $Y$ b = baaabaaabbab $S(X) = baaa$ $S(Y) = bba$

baaabaa bba b = baaabaaabbab

$bXabaabYab = baaabaaabbab$ $S'(X) = aa$ $S'(Y) = b$
Uncrossing $ab$

$$X \ baa \ Y \ b = baaabaabbab \quad S(X) = baaa \quad S(Y) = bba$$

$$baaabaab \ bba \ b = baaabaabbab$$

$$baaabaab \ ba \ b = baaabaabbab$$

$$bX \ abaab \ Y \ ab = baaabaabbab \quad S'(X) = aa \quad S'(Y) = b$$
Definition (maximal block of $a$)

When $a^\ell$ occurs in $S(U) = S(V)$ and cannot be extended.
Maximal blocks

Definition (maximal block of $a$)

When $a^\ell$ occurs in $S(U) = S(V)$ and cannot be extended.

Equivalents of pairs.
Maximal blocks

**Definition (maximal block of \(a\))**

When \(a^\ell\) occurs in \(S(U) = S(V)\) and cannot be extended.

Equivalents of pairs.

- Block occurrence can be explicit, implicit or crossing.
- Letter \(a\) is crossing (has a crossing block) if there is a crossing block of \(a\).
Maximal blocks

**Definition (maximal block of $a$)**

When $a^\ell$ occurs in $S(U) = S(V)$ and cannot be extended.

**Equivalents of pairs.**

- Block occurrence can be explicit, implicit or crossing.
- Letter $a$ is crossing (has a crossing block) if there is a crossing block of $a$.

\[
\begin{align*}
X \ baa \ Y \ b &= baabbaabbb \\
S(X) &= baab \\
S(Y) &= bb \\
baab \ baa \ bb \ b &= baabbaabbb
\end{align*}
\]
Maximal blocks

**Definition (maximal block of \( a \))**

When \( a^\ell \) occurs in \( S(U) = S(V) \) and cannot be extended.

**Equivalents of pairs.**

- Block occurrence can be explicit, implicit or crossing.
- Letter \( a \) is crossing (has a crossing block) if there is a crossing block of \( a \).

\[
\begin{align*}
X \ baa \ Y \ b &= baabbaabbb \\
S(X) &= baab \quad S(Y) = bb \\
baab \ baa \ bb \ b &= baabbaabbb
\end{align*}
\]

**Lemma (Length-minimal solutions)**

*If \( a^\ell \) is a maximal block in a length-minimal solution of \( U = V \) then \( \ell \leq 2^c|UV| \).*
When $a$ has no crossing block

1: for all maximal blocks $a^\ell$ of $a$ and $\ell > 1$ do
2: let $a_\ell \in \Sigma$ be an unused letter
3: replace each explicit maximal $a^\ell$ in $U = V$ by $a_\ell$
**Blocks compression**

When \( a \) has no crossing block

1. **for** all maximal blocks \( a^\ell \) of \( a \) and \( \ell > 1 \) do
2. let \( a^\ell \in \Sigma \) be an unused letter
3. replace each explicit maximal \( a^\ell \) in \( U = V \) by \( a^\ell \)

**Lemma**

The BlockComp\((a)\) properly compresses noncrossing blocks of \( a \).
**Blocks compression**

**When \(a\) has no crossing block**

1. **for** all maximal blocks \(a^\ell\) of \(a\) and \(\ell > 1\) **do**
2. let \(a_\ell \in \Sigma\) be an unused letter
3. replace each explicit maximal \(a^\ell\) in \(U = V\) by \(a_\ell\)

**Lemma**

*The BlockComp\((a)\) properly compresses noncrossing blocks of \(a\).*

\[
X \quad baaYbaaa = baabbaabbbbaaa \quad S(X) = baab \quad S(Y) = bb
\]
### Blocks compression

#### When $a$ has no crossing block

1. **for** all maximal blocks $a^\ell$ of $a$ and $\ell > 1$ **do**
2. let $a_\ell \in \Sigma$ be an unused letter
3. replace each explicit maximal $a^\ell$ in $U = V$ by $a_\ell$

#### Lemma

*The BlockComp$(a)$ properly compresses noncrossing blocks of $a$.***

\[
X \overset{\text{baa}Y\text{baaa}}{=\text{baabbaabbbbaaa}} \quad S(X) = \text{baab} \quad S(Y) = \text{bb}
\]

\[
\text{baabbaabbbbaaa} = \overset{\text{baabbaabbbbaaa}}{=\text{baabbaabbbbaaa}}
\]
Blocks compression

When $a$ has no crossing block

1: for all maximal blocks $a^\ell$ of $a$ and $\ell > 1$ do
2: let $a^\ell \in \Sigma$ be an unused letter
3: replace each explicit maximal $a^\ell$ in $U = V$ by $a^\ell$

Lemma

The BlockComp$(a)$ properly compresses noncrossing blocks of $a$.

$$X \ baaYbaaa = baabbaabbbaaaa \quad S(X) = baab \quad S(Y) = bb$$

$$baabbaabbbaaaa = baabbaabbbaaaa$$

$$X \ ba_2Yb \ a_3 = ba_2bbba_2bbb \ a_3 \quad S'(X) = ba_2b \quad S'(Y) = bb$$
When $a$ has no crossing block

1: for all maximal blocks $a^\ell$ of $a$ and $\ell > 1$ do
2: let $a_\ell \in \Sigma$ be an unused letter
3: replace each explicit maximal $a^\ell$ in $U = V$ by $a_\ell$

Lemma

The BlockComp$(a)$ properly compresses noncrossing blocks of $a$.

\[
\begin{align*}
X \quad baaY baaa &= baabbaabbbbaaa \\
baabbbabbbbbaaa &= baabbbabbbbaaa \\
ba_2 bb a_2 bbb a_3 &= ba_2 bba_2 bbb a_3 \\
X \quad ba_2 Y b a_3 &= ba_2 bba_2 bbb a_3 \\
S(X) &= baab \quad S(Y) = bb \\
S'(X) &= ba_2 b \quad S'(Y) = bb
\end{align*}
\]
Crossing $a$-chains?

- Crossing $a$-chain: similar to crossing $ab$. 

Lemma

After uncrossing $a$ is no longer crossing.
Crossing $a$-chains?

- Crossing $a$-chain: similar to crossing $ab$.
- **pop whole $a$-prefix and $a$-suffix:**
  $S(X) = a^\ell_x w a^r_x$: change it to $S(X) = w$
Crossing $a$-chains?

- Crossing $a$-chain: similar to crossing $ab$.
- pop whole $a$-prefix and $a$-suffix:
  \[
  S(X) = a^{\ell_X}wa^{r_X} : \text{change it to } S(X) = w
  \]

1. \textbf{for } $X \in \mathcal{X}$ \textbf{do}
2. \quad replace each occurrence of $X$ by $a^{\ell_X}Xa^{r_X}$ \hspace{1cm} $\triangleright \ell_X, r_X \geq 0$
3. \quad $\triangleright a^{\ell_X}$ and $a^{r_X}$ are the $a$-prefix and suffix of $S(X)$
4. \quad \textbf{if } $S(X) = \epsilon$ \textbf{then}
5. \quad remove $X$ from the equation
Crossing $a$-chains?

- Crossing $a$-chain: similar to crossing $ab$.
- pop whole $a$-prefix and $a$-suffix:
  \[ S(X) = a^{\ell_X}w a^{r_X} : \text{change it to } S(X) = w \]

```
1: for $X \in \mathcal{X}$ do
2:   replace each occurrence of $X$ by $a^{\ell_X} X a^{r_X} \triangleright \ell_X, r_X \geq 0$
3:   \triangleright $a^{\ell_X}$ and $a^{r_X}$ are the $a$-prefix and suffix of $S(X)$
4: if $S(X) = \epsilon$ then
5:   remove $X$ from the equation
```

Lemma

After uncrossing $a$ is no longer crossing.
Algorithm

while $U \notin \Sigma$ and $V \notin \Sigma$ do
    L ← letters from $U = V$
    choose a pair of letters or a block from L
    if it is crossing then
        Uncross it
    Compress it
If the new equation has a solution, then also the original one had.
If the new equation has a solution, then also the original one had.

Just roll back the changes.
If the new equation has a solution, then also the original one had.

Just roll back the changes.

\[ X \ baa \ Y \ b = baaabaabbab \]
\[ X \ c \ a \ Y \ b = c \ aa \ c \ ab \ c \ b \]
\[ S(X) = caa \ S(Y) = bc \]
If the new equation has a solution, then also the original one had.

Just roll back the changes.

\[ X \ baa \ Y \ b = baaabaabbab \]
\[ c \ aa \ c \ ab \ c \ b = c \ aa \ c \ ab \ c \ b \]
\[ X \ c \ a \ Y \ b = c \ aa \ c \ ab \ c \ b \]
\[ S(X) = caa \ S(Y) = bc \]
If the new equation has a solution, then also the original one had.

Just roll back the changes.

\[ \begin{align*}
X &= baa 
Y &= baaaabaabbab \\
S(X) &= baaa 
S(Y) &= bba \\
\end{align*} \]

\[ \begin{align*}
X &= ca \quad Y = c 
X &= ca 
S(X) &= caa 
S(Y) &= bc
\end{align*} \]
Soundness

If the new equation has a solution, then also the original one had.

Just roll back the changes.

\[
\begin{align*}
X & \quad baa \quad Y \quad b = baaabaabbab \\
baaaabaabbab & = baaabaabbab \\
S(X) & = baaa \quad S(Y) = bba
\end{align*}
\]

\[
\begin{align*}
\text{c aa} & \quad \text{c ab} \quad \text{c b} = \text{c aa} \quad \text{c ab} \quad \text{c b} \\
X & \quad \text{c a} \quad Y \quad b = \text{c aa} \quad \text{c ab} \quad \text{c b} \\
S(X) & = \text{caa} \quad S(Y) = \text{bc}
\end{align*}
\]
If the equation has the solution, then for some nondeterministic choices the new equation has a corresponding one.
If the equation has the solution, then for some nondeterministic choices the new equation has a corresponding one.

Make the choices according to the solution.
Completeness

If the equation has the solution, then for some nondeterministic choices the new equation has a corresponding one.

Make the choices according to the solution.

What about termination?
Termination

We show that

- we stay in $O(n^2)$ space.
- After each operation the length-minimal solution shortens.
We show that

- we stay in $O(n^2)$ space.
- After each operation the length-minimal solution shortens.

So we terminate on positive instances.
We show that

- we stay in $O(n^2)$ space.
- After each operation the length-minimal solution shortens.

So we terminate on positive instances.

**Lemma**

*Each compression decreases the length of the length-minimal solution.*

**Proof.**

We perform the compression on the solution word.
Strategy

Lemma

*Compression of a non-crossing pair/block decreases equation’s size.*

Proof.

Something is compressed in the equation.
Lemma

*Compression of a non-crossing pair/block decreases equation’s size.*

Proof.

Something is compressed in the equation.

Strategy

- If there is something non-crossing: compress it.
- If there are only crossing: choose one that minimises the equation.
Lemma (Fixed solution)

There are at most $2n$ different crossing pairs and blocks.
Lemma (Fixed solution)

There are at most $2n$ different crossing pairs and blocks.

Each is associated with a side of an occurrence of a variable.
Lemma (Fixed solution)

There are at most \(2n\) different crossing pairs and blocks.

Each is associated with a side of an occurrence of a variable.

Lemma (Fixed solution)

Uncrossing introduces at most \(2n\) letters to the equation.
Lemma (Fixed solution)

There are at most $2n$ different crossing pairs and blocks.

Each is associated with a side of an occurrence of a variable.

Lemma (Fixed solution)

Uncrossing introduces at most $2n$ letters to the equation.

Each variable pops left and right one letter for $a$-chains: it is compressed immediately afterwards.
Lemma (Fixed solution)

*There are at most* $2n$ *different crossing pairs and blocks.*

Each is associated with a side of an occurrence of a variable.

Lemma (Fixed solution)

*Uncrossing introduces at most* $2n$ *letters to the equation.*

Each variable pops left and right one letter for $a$-chains: it is compressed immediately afterwards.

Lemma

*There is always some choice to be* $\leq 8n^2$. 
Lemma (Fixed solution)

There are at most $2n$ different crossing pairs and blocks.

Each is associated with a side of an occurrence of a variable.

Lemma (Fixed solution)

Uncrossing introduces at most $2n$ letters to the equation.

Each variable pops left and right one letter for $a$-chains: it is compressed immediately afterwards.

Lemma

There is always some choice to be $\leq 8n^2$.

There are $m \leq 8n^2$ letters and $k \leq 2n$ different crossing blocks/pairs. Some covers $\geq m/k$ letters. Its compression removes $\geq m/2k$ letters and introduces $2n$ letters. We are left with at most

$$(1 - 1/2k) \cdot m + 2n \leq (1 - 1/4n) \cdot 8n^2 + 2n = 8n^2.$$
Conclusions and Open questions

Conclusions

» The representation can be more important than the combinatorics.

Open questions

» Are word equations in NP? (Are solutions at most exponential?)
» To which problems can we generalise this approach?
Conclusions and Open questions

Conclusions

- The representation can be more important than the combinatorics.

Open questions

- Are word equations in NP? (Are solutions at most exponential?)
- To which problems can we generalise this approach?
Regular constraints

For each variable: constraints of the form $X \in R, X \notin R'$
Regular constraints

For each variable: constraints of the form $X \in R, X \notin R'$

$\rho$: homomorphism from letters to transition matrices of NFAs extend also to variables: $\rho_X$, require $\rho(S(X)) = \rho_X$
Regular constraints

For each variable: constraints of the form $X \in R, X \notin R'$

$\rho$: homomorphism from letters to transition matrices of NFAs extend also to variables: $\rho_X$, require $\rho(S(X)) = \rho_X$

when $w$ is replaced by $c$: $\rho(c) \leftarrow \rho(w)$
when $X$ is replaced with $wX$: $\rho_X \leftarrow \rho'_X$ such that $\rho_X = \rho(w)\rho'_X$
when $X$ is removed: check $\rho_X = \rho(\epsilon)$
(some extra tricks in the analysis)
Using parallel compression: length $\mathcal{O}(n) \implies \mathcal{O}(n \log n)$ bits

Using Huffman coding: linear-size (in terms of bits)
Space saving

Using parallel compression: length $\mathcal{O}(n) \implies \mathcal{O}(n \log n)$ bits

Using Huffman coding: linear-size (in terms of bits)
Even if input is Huffman-coded.