Towards Understanding the Complexity of Fragments of Word Equations

Joel D. Day, Florin Manea and Dirk Nowotka

May 7, 2019

Word Equations

Notation:

- Let $X := \{x, y, z, \ldots\}$ be a set of variables.

Word Equations

Notation:

- Let $X := \{x, y, z, \ldots\}$ be a set of variables.
- Let $A := \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ be a set of terminal symbols.

Word Equations

Notation:

- Let $X := \{x, y, z, \ldots\}$ be a set of variables.
- Let $A := \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ be a set of terminal symbols.
- Let $U, V \in (X \cup A)^*$. Then $U = V$ is a word equation.

Word Equations

Notation:

- Let $X := \{x, y, z, \ldots\}$ be a set of variables.
- Let $A := \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ be a set of terminal symbols.
- Let $U, V \in (X \cup A)^*$. Then $U = V$ is a word equation.
- Solutions are substitutions of terminal words for the variables such that the LHS and RHS become identical.

Word Equations

Notation:

- Let $X := \{x, y, z, \ldots\}$ be a set of variables.
- Let $A := \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots\}$ be a set of terminal symbols.
- Let $U, V \in (X \cup A)^*$. Then $U = V$ is a word equation.
- Solutions are substitutions of terminal words for the variables such that the LHS and RHS become identical.
- In other words, solutions are terminal-preserving homomorphisms $h : (X \cup A)^* \to A^*$ such that $h(U) = h(V)$.

The Satisfiability Problem:

Given a word equation $U = V$, does there exist a solution h?

The Satisfiability Problem:

Given a word equation $U = V$, does there exist a solution h? Does there exist a solution h satisfying some additional constraints?

Fragments of Word Equations

Motivation:

- Understanding the complexity of the satisfiability problem is important both for understanding the theory of word equations and for practical applications and as such, the exact complexity remains an important long-standing open problem.

Fragments of Word Equations

Motivation:

- Understanding the complexity of the satisfiability problem is important both for understanding the theory of word equations and for practical applications and as such, the exact complexity remains an important long-standing open problem.
- The premise of this talk is that it is also worth understanding the complexity for smaller fragments.

Fragments of Word Equations

Motivation:

- Understanding the complexity of the satisfiability problem is important both for understanding the theory of word equations and for practical applications and as such, the exact complexity remains an important long-standing open problem.
- The premise of this talk is that it is also worth understanding the complexity for smaller fragments.
- From a theoreticians point of view, this is a natural tactic for improving understanding overall.

Fragments of Word Equations

Motivation:

- Understanding the complexity of the satisfiability problem is important both for understanding the theory of word equations and for practical applications and as such, the exact complexity remains an important long-standing open problem.
- The premise of this talk is that it is also worth understanding the complexity for smaller fragments.
- From a theoreticians point of view, this is a natural tactic for improving understanding overall.
- Some fragments may be more relevant to practical applications than the general case anyway.

Fragments of Word Equations

Motivation:

- Understanding the complexity of the satisfiability problem is important both for understanding the theory of word equations and for practical applications and as such, the exact complexity remains an important long-standing open problem.
- The premise of this talk is that it is also worth understanding the complexity for smaller fragments.
- From a theoreticians point of view, this is a natural tactic for improving understanding overall.
- Some fragments may be more relevant to practical applications than the general case anyway.
- We need tools for showing upper bounds in particular.

An Interesting Fragment

Quadratic word equations (QWEs):

- QWEs are equations $U = V$ in which each variable x may occur at most twice in UV.

An Interesting Fragment

Quadratic word equations (QWEs):

- QWEs are equations $U = V$ in which each variable x may occur at most twice in UV.
- Satisfiability of quadratic equations remains NP-hard [Diekert, Robson '99].

An Interesting Fragment

Quadratic word equations (QWEs):

- QWEs are equations $U = V$ in which each variable x may occur at most twice in UV.
- Satisfiability of quadratic equations remains NP-hard [Diekert, Robson '99].
- There is simple proof of decidability (via Nielson Transformations).

An Interesting Fragment

Quadratic word equations (QWEs):

- QWEs are equations $U = V$ in which each variable x may occur at most twice in UV.
- Satisfiability of quadratic equations remains NP-hard [Diekert, Robson '99].
- There is simple proof of decidability (via Nielson Transformations).
- As with the general case, inclusion in NP remains a long-standing open problem.

A Hard/Simple Fragment

Strictly Regular-Ordered Equations (SROWEs)

- SROWEs are equations $U = V$ which have the form

$$u_0 x_1 u_1 x_2 u_2 \dots x_n u_n = v_0 x_1 v_1 x_2 v_2 \dots x_n v_n.$$

where $u_i, v_i \in A^*$ and the x_is are (distinct) variables.

A Hard/Simple Fragment

Strictly Regular-Ordered Equations (SROWEs)

- SROWEs are equations $U = V$ which have the form

$$u_0 x_1 u_1 x_2 u_2 \dots x_n u_n = v_0 x_1 v_1 x_2 v_2 \dots x_n v_n.$$

where $u_i, v_i \in A^*$ and the x_is are (distinct) variables.

Theorem

The satisfiability Problem is NP*-complete for SROWEs.*

A Hard/Simple Fragment

Strictly Regular-Ordered Equations (SROWEs)

- SROWEs are equations $U = V$ which have the form

$$u_0 x_1 u_1 x_2 u_2 \dots x_n u_n = v_0 x_1 v_1 x_2 v_2 \dots x_n v_n.$$

where $u_i, v_i \in A^*$ and the x_is are (distinct) variables.

Theorem

The satisfiability Problem is NP*-complete for SROWEs.*

- Inclusion in NP is straightforward: minimal solutions will be short (linear).

A Hard/Simple Fragment

Strictly Regular-Ordered Equations (SROWEs)

- SROWEs are equations $U = V$ which have the form

$$u_0 x_1 u_1 x_2 u_2 \dots x_n u_n = v_0 x_1 v_1 x_2 v_2 \dots x_n v_n.$$

where $u_i, v_i \in A^*$ and the x_is are (distinct) variables.

Theorem

The satisfiability Problem is NP*-complete for SROWEs.*

- Inclusion in NP is straightforward: minimal solutions will be short (linear).
- Showing the lower bounds is much more involved, and is done by reduction from 3-Partition.

A Simple NP-hard Fragment

Additional Constraints:

- **DFA Constraints:** for each variable x, $h(x)$ must belong to the language of some DFA A_x.

A Simple NP-hard Fragment

Additional Constraints:

- **DFA Constraints:** for each variable x, $h(x)$ must belong to the language of some DFA A_x.
- **Length Constraints:** $|h(x)| = 3|h(y)| + 2 \textit{ and} |h(z)| \geq 2$.

A Simple NP-hard Fragment

Additional Constraints:

- **DFA Constraints:** for each variable x, $h(x)$ must belong to the language of some DFA A_x.
- **Length Constraints:** $|h(x)| = 3|h(y)| + 2 \; and |h(z)| \geq 2$.
- **Letter Counting Constraints:** $|h(x)|_{\mathtt{b}} + 1 = 2|h(y)|_{\mathtt{a}}$.

A Simple NP-hard Fragment

Additional Constraints:

- **DFA Constraints:** for each variable x, $h(x)$ must belong to the language of some DFA A_x.
- **Length Constraints:** $|h(x)| = 3|h(y)| + 2$ *and*$|h(z)| \geq 2$.
- **Letter Counting Constraints:** $|h(x)|_{\texttt{b}} + 1 = 2|h(y)|_{\texttt{a}}$.
- **Subword Constraints:** $h(x)$ is a scattered subword of $h(y)$.

A Simple NP-hard Fragment

Additional Constraints:

- **DFA Constraints:** for each variable x, $h(x)$ must belong to the language of some DFA A_x.
- **Length Constraints:** $|h(x)| = 3|h(y)| + 2$ *and* $|h(z)| \geq 2$.
- **Letter Counting Constraints:** $|h(x)|_{\mathtt{b}} + 1 = 2|h(y)|_{\mathtt{a}}$.
- **Subword Constraints:** $h(x)$ is a scattered subword of $h(y)$.

Theorem

The satisfiability problem for SROWEs with DFA, length, letter-counting and subword constraints is NP*-complete.*

A Simple NP-hard Fragment

Regular-Ordered Equations (ROWEs):

- If we relax the definition slightly, things start to get slightly harder.

A Simple NP-hard Fragment

Regular-Ordered Equations (ROWEs):

- If we relax the definition slightly, things start to get slightly harder.
- ROWEs have a similar form as SROWEs, but we allow some variables to occur only once (i.e. on one side only).

A Simple NP-hard Fragment

Regular-Ordered Equations (ROWEs):

- If we relax the definition slightly, things start to get slightly harder.
- ROWEs have a similar form as SROWEs, but we allow some variables to occur only once (i.e. on one side only).

$$E.g.\ \ x_1\mathtt{aba}x_2x_3 = \mathtt{b}x_1\mathtt{a}x_3\mathtt{ba}$$

A Simple NP-hard Fragment

Regular-Ordered Equations (ROWEs):

- If we relax the definition slightly, things start to get slightly harder.
- ROWEs have a similar form as SROWEs, but we allow some variables to occur only once (i.e. on one side only).

$$E.g.\ \ x_1 \mathtt{aba} x_2 x_3 = \mathtt{b} x_1 \mathtt{a} x_3 \mathtt{ba}$$

Theorem

The satisfiability problem for the single regular-ordered equation $xy = yz$ with regular constraints is PSPACE-complete.

A Simple NP-hard Fragment

Regular-Ordered Equations (ROWEs):

- If we relax the definition slightly, things start to get slightly harder.
- ROWEs have a similar form as SROWEs, but we allow some variables to occur only once (i.e. on one side only).

$$E.g.\ \ x_1\mathtt{aba}x_2x_3 = \mathtt{b}x_1\mathtt{a}x_3\mathtt{ba}$$

Theorem

The satisfiability problem for the single regular-ordered equation $xy = yz$ with regular constraints is PSPACE-complete.

Theorem

The Satisfiability Problem for ROWEs (without additional constraints) is NP*-complete.*

Upper Bounds

- Moving toward more interesting/general classes, we need tools to reason about the non-minimality of solutions.

Upper Bounds

- Moving toward more interesting/general classes, we need tools to reason about the non-minimality of solutions.
- We establish a condition for parts of a solution to be 'removable' (thus implying non-minimality) based on a representation of solutions as **chains of positions**.

Upper Bounds

- Moving toward more interesting/general classes, we need tools to reason about the non-minimality of solutions.
- We establish a condition for parts of a solution to be 'removable' (thus implying non-minimality) based on a representation of solutions as **chains of positions**.
- While this representation can be generalised to all equations, we shall see that it yields particular benefits for QWEs.

Chains Representation of Solutions to QWEs

Positions:

- Let $U = V$ be a QWE E, and let h be a solution of E, so that $h(U) = h(V)$.

Chains Representation of Solutions to QWEs

Positions:

- Let $U = V$ be a QWE E, and let h be a solution of E, so that $h(U) = h(V)$.
- We number each occurrence of a letter/variable in the equation from left to right.

$$x\ x\ \mathtt{a}\ \mathtt{a}\ y = z\ y\ \mathtt{b}\ z \to x_{(1)}x_{(2)}\mathtt{a}_{(1)}\mathtt{a}_{(2)}y_{(1)} = z_{(1)}y_{(2)}\mathtt{b}_{(1)}z_{(2)}$$

Chains Representation of Solutions to QWEs

Positions:

- Let $U = V$ be a QWE E, and let h be a solution of E, so that $h(U) = h(V)$.
- We number each occurrence of a letter/variable in the equation from left to right.

$$x\ x\ \mathtt{a}\ \mathtt{a}\ y = z\ y\ \mathtt{b}\ z \rightarrow x_{(1)}x_{(2)}\mathtt{a}_{(1)}\mathtt{a}_{(2)}y_{(1)} = z_{(1)}y_{(2)}\mathtt{b}_{(1)}z_{(2)}$$

- The set of **positions** w.r.t. (E, h) is

$$\mathcal{P}_E^h = \{(x, i, d) \mid x \in X \cup A \wedge 1 \leq |UV|_x \leq i \wedge 1 \leq d \leq |h(x)|\}$$

Chains Representation of Solutions to QWEs

Positions:

- Intuitively, a position refers to a particular letter in the solution-word, specified by where it occurs relative to a particular occurrence of a variable or terminal.

Chains Representation of Solutions to QWEs

Positions:

- Intuitively, a position refers to a particular letter in the solution-word, specified by where it occurs relative to a particular occurrence of a variable or terminal.
- Hence there are $|h(U)| + |h(V)|$ total positions.

Chains Representation of Solutions to QWEs

Positions:

- Intuitively, a position refers to a particular letter in the solution-word, specified by where it occurs relative to a particular occurrence of a variable or terminal.
- Hence there are $|h(U)| + |h(V)|$ total positions.
- Since h is a solution, every position has the same letter as its ‘neighbour’ on the other side of the equation.

Chains Representation of Solutions to QWEs

Positions:

- Intuitively, a position refers to a particular letter in the solution-word, specified by where it occurs relative to a particular occurrence of a variable or terminal.
- Hence there are $|h(U)| + |h(V)|$ total positions.
- Since h is a solution, every position has the same letter as its 'neighbour' on the other side of the equation.
- For any variable x and $i_1, i_2, d \in \mathbb{N}$, we must also have that the positions (x, i_1, d) and (x, i_2, d) have the same value.

Example:

$$x \ \ y \ \ \mathtt{a} \ \ y = \mathtt{a} \ \ z \ \ \mathtt{b}\,\mathtt{b} \ \ x \ \ z$$

$$h(x) = \mathtt{aaab}, \ h(y) = \mathtt{baa}, \ h(z) = \mathtt{aa}$$

<table>
<tr><td>U</td><td colspan="4">x</td><td colspan="3">y</td><td>a</td><td colspan="3">y</td></tr>
<tr><td>$h(U)$</td><td>a</td><td>a</td><td>a</td><td>b</td><td>b</td><td>a</td><td>a</td><td>a</td><td>b</td><td>a</td><td>a</td></tr>
<tr><td>$h(V)$</td><td>a</td><td>a</td><td>a</td><td>b</td><td>b</td><td>a</td><td>a</td><td>a</td><td>b</td><td>a</td><td>a</td></tr>
<tr><td>V</td><td>a</td><td colspan="2">z</td><td>b</td><td>b</td><td colspan="4">x</td><td colspan="2">z</td></tr>
</table>

Example:

$$x_{(1)} \; y_{(1)} \; \mathtt{a}_{(1)} \; y_{(2)} = \mathtt{a}_{(2)} \; z_{(1)} \; \mathtt{b}_{(1)} \, \mathtt{b}_{(2)} \; x_{(2)} \; z_{(2)}$$

$$h(x) = \mathtt{aaab}, \; h(y) = \mathtt{baa}, \; h(z) = \mathtt{aa}$$

U	$x_{(1)}$				$y_{(1)}$			$\mathtt{a}_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$\mathtt{a}_{(2)}$	$z_{(1)}$		$\mathtt{b}_{(1)}$	$\mathtt{b}_{(2)}$	$x_{(2)}$				$z_{(2)}$	

$(x, 1, 3)$

Chains Representation of Solutions to QWEs

Example:

$$x_{(1)} \;\; y_{(1)} \;\; \mathtt{a}_{(1)} \;\; y_{(2)} = \mathtt{a}_{(2)} \;\; z_{(1)} \;\; \mathtt{b}_{(1)}\, \mathtt{b}_{(2)} \;\; x_{(2)} \;\; z_{(2)}$$

$$h(x) = \mathtt{aaab},\; h(y) = \mathtt{baa},\; h(z) = \mathtt{aa}$$

U	$x_{(1)}$				$y_{(1)}$			$\mathtt{a}_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$\mathtt{a}_{(2)}$	$z_{(1)}$		$\mathtt{b}_{(1)}$	$\mathtt{b}_{(2)}$	$x_{(2)}$				$z_{(2)}$	

$(z, 2, 1)$

Chains Representation of Solutions to QWEs

Positions (Neighbour Relation):

- Every position has a unique **neighbour** corresponding to the same position on the other side of the equation.

Chains Representation of Solutions to QWEs

Example:

$$x_{(1)}\ y_{(1)}\ \mathtt{a}_{(1)}\ y_{(2)} = \mathtt{a}_{(2)}\ z_{(1)}\ \mathtt{b}_{(1)}\mathtt{b}_{(2)}\ x_{(2)}\ z_{(2)}$$

$$h(x) = \mathtt{aaab},\ h(y) = \mathtt{baa},\ h(z) = \mathtt{aa}$$

U	$x_{(1)}$				$y_{(1)}$			$\mathtt{a}_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$\mathtt{a}_{(2)}$	$z_{(1)}$		$\mathtt{b}_{(1)}$	$\mathtt{b}_{(2)}$	$x_{(2)}$				$z_{(2)}$	

$(x, 1, 3)$ and $(z, 1, 2)$ are **neighbours**

Chains Representation of Solutions to QWEs

Positions (Sibling Relation):

- Every position associated with a variable occurring twice has a **sibling** corresponding to the other occurrence of that variable.
- More formally, two positions (x, i, d) and (y, j, e) are siblings if $x = y$, $d = e$ and $i \neq j$.

Chains Representation of Solutions to QWEs

Example:

$$x_{(1)}\ \ y_{(1)}\ \ \mathtt{a}_{(1)}\ \ y_{(2)} = \mathtt{a}_{(2)}\ \ z_{(1)}\ \ \mathtt{b}_{(1)}\,\mathtt{b}_{(2)}\ \ x_{(2)}\ \ z_{(2)}$$

$$h(x) = \mathtt{aaab},\ \ h(y) = \mathtt{baa},\ \ h(z) = \mathtt{aa}$$

U	$x_{(1)}$				$y_{(1)}$			$\mathtt{a}_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$\mathtt{a}_{(2)}$	$z_{(1)}$		$\mathtt{b}_{(1)}$	$\mathtt{b}_{(2)}$	$x_{(2)}$				$z_{(2)}$	

$(x, 1, 3)$ and $(x, 2, 3)$ are **Siblings**

Chains Representation of Solutions to QWEs

Construction of Chains: We partition the solution h into chains of positions $p_1 \to p_2 \to \ldots \to p_k$ as follows:

Chains Representation of Solutions to QWEs

Construction of Chains: We partition the solution h into chains of positions $p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_k$ as follows:

- Take a position $p \in \mathcal{P}_E^h$ corresponding to a terminal symbol, or a variable which occurs only once.

Chains Representation of Solutions to QWEs

Construction of Chains: We partition the solution h into chains of positions $p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_k$ as follows:

- Take a position $p \in \mathcal{P}_E^h$ corresponding to a terminal symbol, or a variable which occurs only once.
- $p_1 = p$ and p_2 is the (unique) neighbour of p_1.

Chains Representation of Solutions to QWEs

Construction of Chains: We partition the solution h into chains of positions $p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_k$ as follows:

- Take a position $p \in \mathcal{P}_E^h$ corresponding to a terminal symbol, or a variable which occurs only once.
- $p_1 = p$ and p_2 is the (unique) neighbour of p_1.
- for $i \geq 2$, if p_i corresponds to a terminal symbol or variable occurring only once, the chain terminates, and

Chains Representation of Solutions to QWEs

Construction of Chains: We partition the solution h into chains of positions $p_1 \to p_2 \to \ldots \to p_k$ as follows:

- Take a position $p \in \mathcal{P}_E^h$ corresponding to a terminal symbol, or a variable which occurs only once.
- $p_1 = p$ and p_2 is the (unique) neighbour of p_1.
- for $i \geq 2$, if p_i corresponds to a terminal symbol or variable occurring only once, the chain terminates, and
- for $i \geq 2$, if p_i corresponds to a variable occurring twice, p_{i+1} is the neighbour of the sibling of p_i.

Construction of Chains (Example):

U	$x_{(1)}$				$y_{(1)}$			$\mathtt{a}_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$\mathtt{a}_{(2)}$	$z_{(1)}$		$\mathtt{b}_{(1)}$	$\mathtt{b}_{(2)}$	$x_{(2)}$				$z_{(2)}$	

$(\mathrm{a}, 2, 1)$

Construction of Chains (Example):

U	$x_{(1)}$				$y_{(1)}$			$a_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$a_{(2)}$	$z_{(1)}$		$b_{(1)}$	$b_{(2)}$	$x_{(2)}$				$z_{(2)}$	

$$(\mathrm{a}, 2, 1) \rightarrow (x, 1, 1) \rightarrow$$

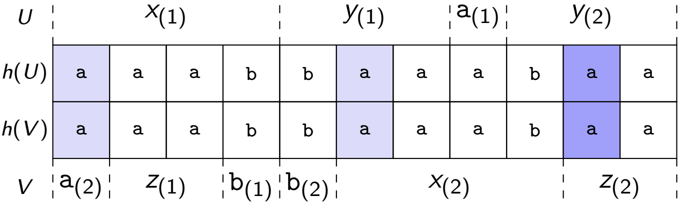
Construction of Chains (Example):

U	$x_{(1)}$				$y_{(1)}$			$\mathtt{a}_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$\mathtt{a}_{(2)}$	$z_{(1)}$		$\mathtt{b}_{(1)}$	$\mathtt{b}_{(2)}$	$x_{(2)}$				$z_{(2)}$	

$$(\mathtt{a}, 2, 1) \to (x, 1, 1) \to (y, 1, 2) \to$$

Constructing the Chains

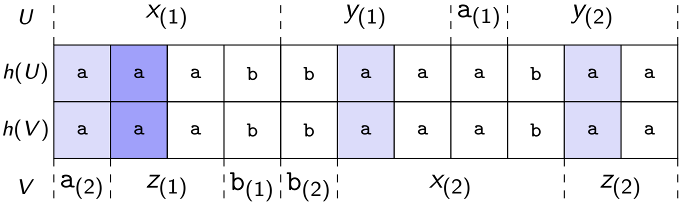
Example:



$$(\mathrm{a},2,1) \to (x,1,1) \to (y,1,2) \to (z,2,1) \to$$

Constructing the Chains

Example:



$$(\mathtt{a},2,1) \to (x,1,1) \to (y,1,2) \to (z,2,1) \to (x,1,2) \to$$

Constructing the Chains

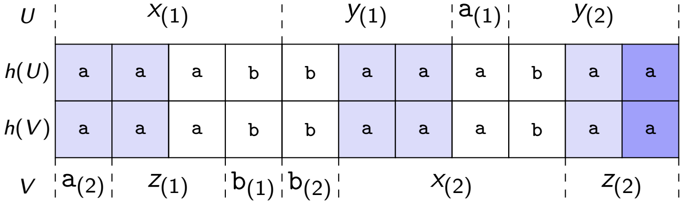
Example:

U	$x_{(1)}$				$y_{(1)}$			$\mathtt{a}_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$\mathtt{a}_{(2)}$	$z_{(1)}$		$\mathtt{b}_{(1)}$	$\mathtt{b}_{(2)}$	$x_{(2)}$				$z_{(2)}$	

$$(\mathtt{a}, 2, 1) \to (x, 1, 1) \to (y, 1, 2) \to (z, 2, 1) \to (x, 1, 2) \to (y, 1, 3) \to$$

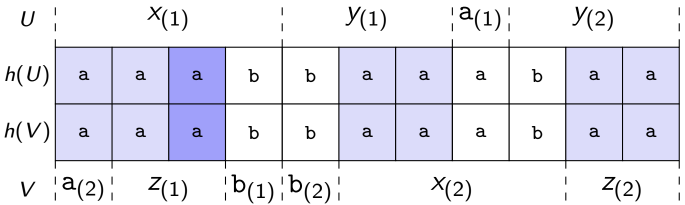
Constructing the Chains

Example:



$$(\mathtt{a},2,1) \to (x,1,1) \to (y,1,2) \to (z,2,1) \to (x,1,2) \to (y,1,3) \to (z,2,2) \to$$

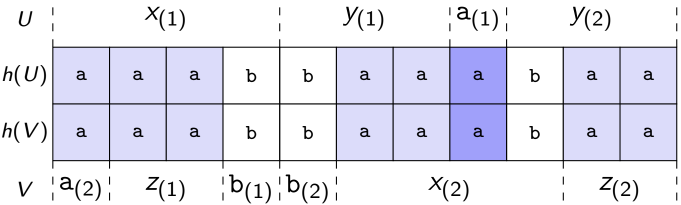
Example:



$$(\mathrm{a}, 2, 1) \rightarrow (x, 1, 1) \rightarrow (y, 1, 2) \rightarrow (z, 2, 1) \rightarrow (x, 1, 2) \rightarrow (y, 1, 3) \rightarrow (z, 2, 2) \rightarrow$$
$$\rightarrow (x, 1, 3)$$

Constructing the Chains

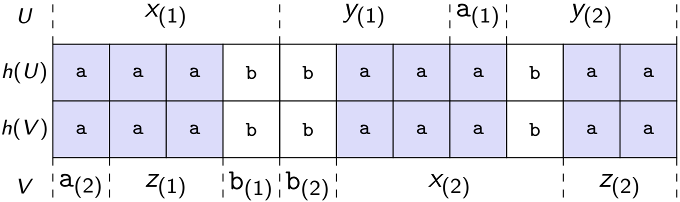
Example:



$$(\mathrm{a}, 2, 1) \to (x, 1, 1) \to (y, 1, 2) \to (z, 2, 1) \to (x, 1, 2) \to (y, 1, 3) \to (z, 2, 2) \to$$
$$\to (x, 1, 3) \to (\mathrm{a}, 1, 1)$$

Constructing the Chains

Example:



$$(a, 2, 1) \to (x, 1, 1) \to (y, 1, 2) \to (z, 2, 1) \to (x, 1, 2) \to (y, 1, 3) \to (z, 2, 2) \to$$
$$\to (x, 1, 3) \to (a, 1, 1)$$

Constructing the Chains

Example:

U	$x_{(1)}$				$y_{(1)}$			$\mathtt{a}_{(1)}$	$y_{(2)}$		
$h(U)$	a	a	a	b	b	a	a	a	b	a	a
$h(V)$	a	a	a	b	b	a	a	a	b	a	a
V	$\mathtt{a}_{(2)}$	$z_{(1)}$		$\mathtt{b}_{(1)}$	$\mathtt{b}_{(2)}$	$x_{(2)}$				$z_{(2)}$	

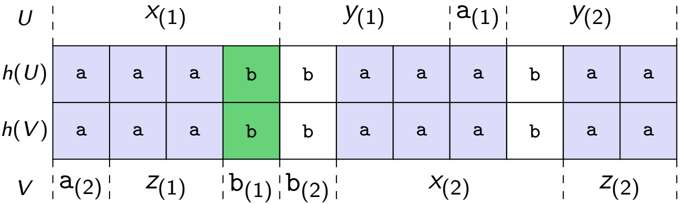
$$(\mathtt{a},2,1) \to (x,1,1) \to (y,1,2) \to (z,2,1) \to (x,1,2) \to (y,1,3) \to (z,2,2) \to$$

$$\to (x,1,3) \to (\mathtt{a},1,1)$$

$$(\mathtt{b},1,1) \to$$

Constructing the Chains

Example:



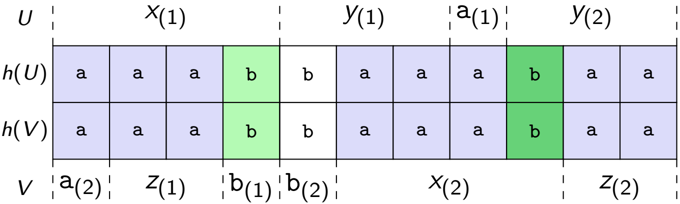
$$(\mathtt{a},2,1) \to (x,1,1) \to (y,1,2) \to (z,2,1) \to (x,1,2) \to (y,1,3) \to (z,2,2) \to$$

$$\to (x,1,3) \to (\mathtt{a},1,1)$$

$$(\mathtt{b},1,1) \to (x,1,4) \to$$

Constructing the Chains

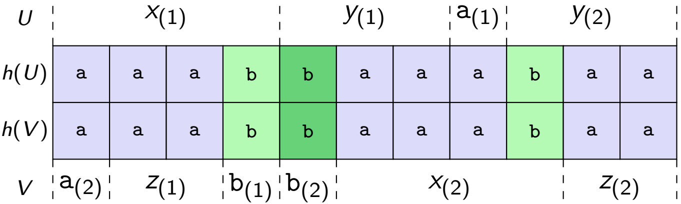
Example:



$$(\mathtt{a}, 2, 1) \to (x, 1, 1) \to (y, 1, 2) \to (z, 2, 1) \to (x, 1, 2) \to (y, 1, 3) \to (z, 2, 2) \to$$
$$\to (x, 1, 3) \to (\mathtt{a}, 1, 1)$$
$$(\mathtt{b}, 1, 1) \to (x, 1, 4) \to (y, 2, 1) \to$$

Constructing the Chains

Example:



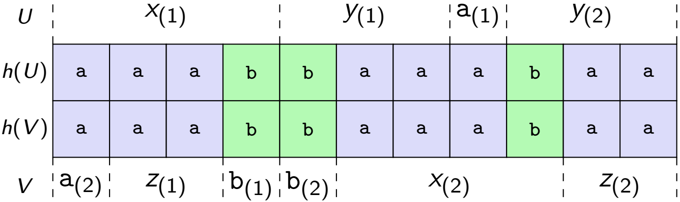
$$(\mathtt{a},2,1) \to (x,1,1) \to (y,1,2) \to (z,2,1) \to (x,1,2) \to (y,1,3) \to (z,2,2) \to$$

$$\to (x,1,3) \to (\mathtt{a},1,1)$$

$$(\mathtt{b},1,1) \to (x,1,4) \to (y,2,1) \to (\mathtt{b},2,1)$$

Constructing the Chains

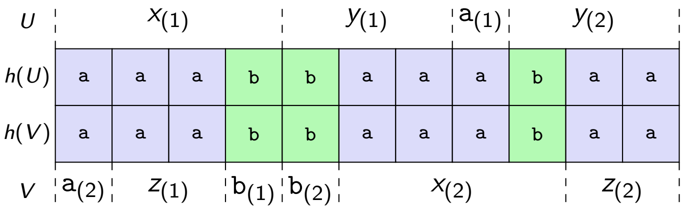
Example:



$$(\mathtt{a},2,1) \to (x,1,1) \to (y,1,2) \to (z,2,1) \to (x,1,2) \to (y,1,3) \to (z,2,2) \to$$
$$\to (x,1,3) \to (\mathtt{a},1,1)$$
$$(\mathtt{b},1,1) \to (x,1,4) \to (y,2,1) \to (\mathtt{b},2,1)$$

Constructing the Chains

Example:



$$(\mathtt{a},2,1) \to (x,1,1) \to (y,1,2) \to (z,2,1) \to (x,1,2) \to (y,1,3) \to (z,2,2) \to$$
$$\to (x,1,3) \to (\mathtt{a},1,1)$$
$$(\mathtt{b},1,1) \to (x,1,4) \to (y,2,1) \to (\mathtt{b},2,1)$$

Constructing the Chains

- If we simply view the chains as equivalence classes, we get the method of **filling the positions**.

Constructing the Chains

- If we simply view the chains as equivalence classes, we get the method of **filling the positions**.
- However, we want to make explicit use of the **order** in which the positions are connected.

Constructing the Chains

- If we simply view the chains as equivalence classes, we get the method of **filling the positions**.
- However, we want to make explicit use of the **order** in which the positions are connected.
- For a minimal solution, the number of chains will be linear in the length of the equation, and the sum of the lengths of the chains will be linear in the length of the solution.

Constructing the Chains

- If we simply view the chains as equivalence classes, we get the method of **filling the positions**.
- However, we want to make explicit use of the **order** in which the positions are connected.
- For a minimal solution, the number of chains will be linear in the length of the equation, and the sum of the lengths of the chains will be linear in the length of the solution.

Lemma

Let h be a minimal solution to some QWE $U = V$. Let $\mathcal{C}$ be the longest chain of h w.r.t $U = V$. Then $|h(U)| \leq |\mathcal{C}||UV|$.

From Chains to Words

Chain-Words:

- Let Γ be an alphabet of size $2|A \cup X|$ and let $\varphi : \mathcal{P}_E^h \to \Gamma$ be an a mapping such that $\varphi((x, i, d)) = \varphi((y, j, e))$ if and only if $x = y$ and $i = j$.

From Chains to Words

Chain-Words:

- Let Γ be an alphabet of size $2|A \cup X|$ and let $\varphi : \mathcal{P}_E^h \to \Gamma$ be an a mapping such that $\varphi((x, i, d)) = \varphi((y, j, e))$ if and only if $x = y$ and $i = j$.
- For each chain $p_1 \to p_2 \to \ldots \to p_n$, we construct a word $w = \varphi(p_1)\varphi(p_2)\varphi(p_3)\ldots\varphi(p_{n-1})\varphi(p_n)$.

From Chains to Words

Chain-Words:

- Let Γ be an alphabet of size $2|A \cup X|$ and let $\varphi : \mathcal{P}_E^h \to \Gamma$ be an a mapping such that $\varphi((x, i, d)) = \varphi((y, j, e))$ if and only if $x = y$ and $i = j$.
- For each chain $p_1 \to p_2 \to \ldots \to p_n$, we construct a word $w = \varphi(p_1)\varphi(p_2)\varphi(p_3)\ldots\varphi(p_{n-1})\varphi(p_n)$.
- We say that w is a **chain-word** of h w.r.t. E.

From Chains to Words

Chain-Words:

- Let Γ be an alphabet of size $2|A \cup X|$ and let $\varphi : \mathcal{P}_E^h \rightarrow \Gamma$ be an a mapping such that $\varphi((x,i,d)) = \varphi((y,j,e))$ if and only if $x = y$ and $i = j$.
- For each chain $p_1 \rightarrow p_2 \rightarrow \ldots \rightarrow p_n$, we construct a word $w = \varphi(p_1)\varphi(p_2)\varphi(p_3)\ldots\varphi(p_{n-1})\varphi(p_n)$.
- We say that w is a **chain-word** of h w.r.t. E.

Squares:

- A word u is a **square** if it is a direct repetition (it has the form $u = vv$ for some non-empty word v).

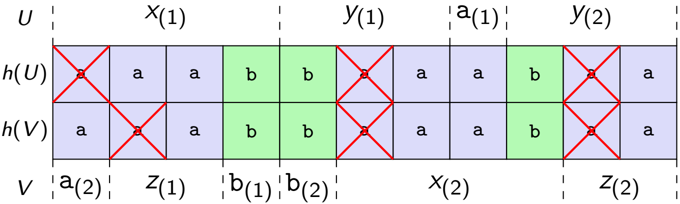
The Squares Lemma

Lemma (Squares Lemma)

Let E be a QWE, h be a solution to E and let w be a chain word of h w.r.t. E. If w contains a square, then h is not minimal.

The Squares Lemma

Example:



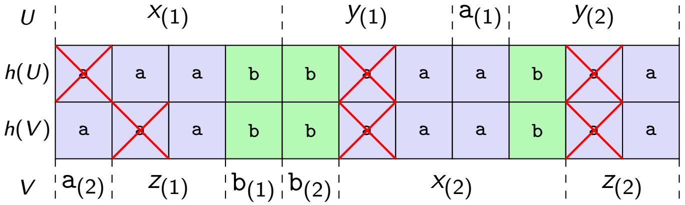
$$(\mathtt{a}, 2, \not{1}) \to (\mathbf{x}, \mathbf{1}, \not{1}) \to (\mathbf{y}, \mathbf{1}, \not{2}) \to (\mathbf{z}, \mathbf{2}, \not{1}) \to (\mathbf{x}, \mathbf{1}, \not{2}) \to (\mathbf{y}, \mathbf{1}, \not{3}) \to (\mathbf{z}, \mathbf{2}, \not{2}) \to$$

$$\to (x, 1, \not{3}) \to (\mathtt{a}, 1, \not{1})$$

$$(\mathtt{b}, 1, \not{1}) \to (x, 1, \not{4}) \to (y, 2, \not{1}) \to (\mathtt{b}, 2, \not{1})$$

The Squares Lemma

Example:



$h'(x) = \mathtt{aab}, h'(y) = \mathtt{ba}, h'(z) = \mathtt{a}$ is also a solution!

h is not minimal.

The Squares Lemma

- The squares lemma provides a general tool for showing complexity upper bounds for (classes of) QWEs – assume that a 'long' solution exists and show that one of the induced chain-words must contain a square.

The Squares Lemma

- The squares lemma provides a general tool for showing complexity upper bounds for (classes of) QWEs – assume that a 'long' solution exists and show that one of the induced chain-words must contain a square.
- We can also generalise it to work with additional constraints on solutions such as regular constraints, and involutions.

The Squares Lemma

- The squares lemma provides a general tool for showing complexity upper bounds for (classes of) QWEs – assume that a ‘long’ solution exists and show that one of the induced chain-words must contain a square.
- We can also generalise it to work with additional constraints on solutions such as regular constraints, and involutions.
- The existence of a square in one of the chain-words corresponds to a set of factors in the solution which may be ’pumped’.

The Squares Lemma

- The squares lemma provides a general tool for showing complexity upper bounds for (classes of) QWEs – assume that a ‘long’ solution exists and show that one of the induced chain-words must contain a square.
- We can also generalise it to work with additional constraints on solutions such as regular constraints, and involutions.
- The existence of a square in one of the chain-words corresponds to a set of factors in the solution which may be 'pumped'.
- Unfortunately, proving that long solutions/chain-words must contain squares seems very difficult.

Upper Bounds

ROWEs

Theorem

The Satisfiability Problem for regular-ordered word equations is NP*-complete.*

Upper Bounds

ROWEs

Theorem

The Satisfiability Problem for regular-ordered word equations is NP*-complete.*

- A quick inspection shows that for ROWEs, the chains will either go from right to left, or left to right (but will never change direction).

Upper Bounds

ROWEs

Theorem

The Satisfiability Problem for regular-ordered word equations is NP*-complete.*

- A quick inspection shows that for ROWEs, the chains will either go from right to left, or left to right (but will never change direction).
- Thus if a chain visits the same variable more than once, it must be consecutively. This would induce a "square", so by our lemma, the solution would not be minimal.

Upper Bounds

ROWEs

Theorem

The Satisfiability Problem for regular-ordered word equations is NP*-complete.*

- A quick inspection shows that for ROWEs, the chains will either go from right to left, or left to right (but will never change direction).
- Thus if a chain visits the same variable more than once, it must be consecutively. This would induce a "square", so by our lemma, the solution would not be minimal.
- In a minimal solution, each chain has length linear in the length of the equation.

Thus any minimal solution is quadratic in the length of the equation.

Upper Bounds

Variable Sparse QWEs (VSQWEs)

- We say that a QWE $U = V$ is **variable-sparse** if

$$|\{x \in X \mid |UV|_x = 2\}| \leq \log |UV|$$

Upper Bounds

Variable Sparse QWEs (VSQWEs)

- We say that a QWE $U = V$ is **variable-sparse** if

$$|\{x \in X \mid |UV|_x = 2\}| \leq \log |UV|$$

Theorem

The Satisfiabiliity Problem for VSQWEs is in NP.

Regular-Reversed Word Equations (RRWEs)

- We say that a QWE $U = V$ is **regular-reversed** if it has the form:

$$u_0 x_1 u_1 x_2 u_2 \dots x_n u_n = v_n x_n v_{n-1} x_{n-1} \dots v_1 x_1 v_0.$$

where $u_i, v_i \in A^*$ and the x_is are (distinct) variables.

Upper Bounds

Regular-Reversed Word Equations (RRWEs)

- We say that a QWE $U = V$ is **regular-reversed** if it has the form:

$$u_0 x_1 u_1 x_2 u_2 \dots x_n u_n = v_n x_n v_{n-1} x_{n-1} \dots v_1 x_1 v_0.$$

where $u_i, v_i \in A^*$ and the x_is are (distinct) variables.

Theorem

The Satisfiability Problem for RRWEs is in NP.

Upper Bounds

Regular-Reversed Word Equations (RRWEs)

- We say that a QWE $U = V$ is **regular-reversed** if it has the form:

$$u_0 x_1 u_1 x_2 u_2 \dots x_n u_n = v_n x_n v_{n-1} x_{n-1} \dots v_1 x_1 v_0.$$

where $u_i, v_i \in A^*$ and the x_is are (distinct) variables.

Theorem

The Satisfiabiliity Problem for RRWEs is in NP*.*

- The proof in this case requires a much more involved analysis, but relies mostly on the squares lemma.

Towards All QWEs

Open Problem

Does there exist an exponential(ish) function f such that, for any QWE $U = V$, if h is a solution and $|h(U)| > f(|UV|)$, then at least one of the chain-words of h w.r.t E contains a square?

Towards All QWEs

Open Problem

Does there exist an exponential(ish) function f such that, for any QWE $U = V$, if h is a solution and $|h(U)| > f(|UV|)$, then at least one of the chain-words of h w.r.t E contains a square?

- A positive answer would imply that the Satisfiability Problem for QWEs is in NP.

Towards All QWEs

Open Problem

Does there exist an exponential(ish) function f such that, for any QWE $U = V$, if h is a solution and $|h(U)| > f(|UV|)$, then at least one of the chain-words of h w.r.t E contains a square?

- A positive answer would imply that the Satisfiability Problem for QWEs is in NP.
- So the question is, what does the set of all chain words of QWEs look like?

Towards All QWEs

Open Problem

Does there exist an exponential(ish) function f such that, for any QWE $U = V$, if h is a solution and $|h(U)| > f(|UV|)$, then at least one of the chain-words of h w.r.t E contains a square?

- A positive answer would imply that the Satisfiability Problem for QWEs is in NP.
- So the question is, what does the set of all chain words of QWEs look like?
- We have a characterisation for **regular** equations (each variable occurs at most once per side).

Towards All QWEs

Theorem

Let w be a word and let Γ be the alphabet of letters occurring in w. There exists a regular word equation E with solution h such that w is a chain-word of h w.r.t. E if and only if there exist letters $\$, \# \notin \Gamma$ and linear orders $<_1$, $<_2$ on the sets $\Gamma \cup \{\#\}$ and $\Gamma \cup \{\$\}$ respectively such that for every $u \in \Gamma^*$ and $A, B, C, D \in \Gamma \cup \{\$, \#\}$ with $A \neq B$ and $C \neq D$, if AuC and BuD are both factors of $\#w\$$, then either that $A <_2 B$ and $C <_1 D$ or that $B <_2 A$ and $D <_1 C$.

Towards All QWEs

- We expect that generalising this to all QWEs is not too hard.

Towards All QWEs

- We expect that generalising this to all QWEs is not too hard.
- As a consequence, we get some further nice restrictions on how possible chain-words might look.

Towards All QWEs

- We expect that generalising this to all QWEs is not too hard.
- As a consequence, we get some further nice restrictions on how possible chain-words might look.

Corollary

Let E be a regular word equation and let h be a solution to E. Let w be a chain-word of h w.r.t. E. Let A, B, C, D be letters from w such that $A \neq B$ and $C \neq D$ Then for any word u, at least one of AuC, BuC, AuD, BuD is not a factor of w.

Towards All QWEs

- We expect that generalising this to all QWEs is not too hard.
- As a consequence, we get some further nice restrictions on how possible chain-words might look.

Corollary

Let E be a regular word equation and let h be a solution to E. Let w be a chain-word of h w.r.t. E. Let A, B, C, D be letters from w such that $A \neq B$ and $C \neq D$ Then for any word u, at least one of AuC, BuC, AuD, BuD is not a factor of w.

Corollary

Let E be a regular word equation and let h be a solution to E. Let w be a chain-word of h w.r.t. E. Let n be the number of variables in E. Then w contains at most $2n - 1$ distinct factors of length 2.

Towards All QWEs

Lemma

Let w be a chain-word of some solution h w.r.t. some regular word equation E. Suppose that w contains a factor of the form $x_1x_2x_3x_4x_2x_1x_3$ such that x_3 is not a prefix of x_1 or x_2. Then some chain-word w' of h w.r.t. E contains a square, and h is not minimal.

Towards All QWEs

Lemma

Let w be a chain-word of some solution h w.r.t. some regular word equation E. Suppose that w contains a factor of the form $x_1x_2x_3x_4x_2x_1x_3$ such that x_3 is not a prefix of x_1 or x_2. Then some chain-word w' of h w.r.t. E contains a square, and h is not minimal.

- Unlike squares, all words which are long enough will encounter a factor of the form $x_1x_2x_3x_4x_2x_1x_3$.

Towards All QWEs

Lemma

Let w be a chain-word of some solution h w.r.t. some regular word equation E. Suppose that w contains a factor of the form $x_1x_2x_3x_4x_2x_1x_3$ such that x_3 is not a prefix of x_1 or x_2. Then some chain-word w' of h w.r.t. E contains a square, and h is not minimal.

- Unlike squares, all words which are long enough will encounter a factor of the form $x_1x_2x_3x_4x_2x_1x_3$.
- Unfortunately, we do not know that the same holds if in addition we ask that x_3 is not a prefix of x_1 or x_2.

Towards All QWEs

Lemma

Let w be a chain-word of some solution h w.r.t. some regular word equation E. Suppose that w contains a factor of the form $x_1x_2x_3x_4x_2x_1x_3$ such that x_3 is not a prefix of x_1 or x_2. Then some chain-word w' of h w.r.t. E contains a square, and h is not minimal.

- Unlike squares, all words which are long enough will encounter a factor of the form $x_1x_2x_3x_4x_2x_1x_3$.
- Unfortunately, we do not know that the same holds if in addition we ask that x_3 is not a prefix of x_1 or x_2.
- It is possible to produce other patterns with prefix/suffix restrictions for which the lemma holds.

Questions?

Thank you!