On the Parameterised Complexity of String Morphism Problems

Henning Fernau, **Markus L. Schmid**,
Trier University

Yngve Villanger,
University of Bergen

Presented 12 December 2013 at FSTTCS

String Morphisms

$\Sigma := \{\mathtt{a}, \mathtt{b}\}$ is a finite alphabet.

String Morphisms

$\Sigma := \{\mathtt{a}, \mathtt{b}\}$ is a finite alphabet.

$w = \mathtt{a\,b\,b\,a} \in \Sigma^*$ is a string.

String Morphisms

$\Sigma := \{\mathtt{a}, \mathtt{b}\}$ is a finite alphabet.

$w = \mathtt{a\,b\,b\,a} \in \Sigma^*$ is a string.

$h : \Sigma \to \Sigma^*$ with

$$h(\mathtt{a}) = \mathtt{b\,b\,a}$$
$$h(\mathtt{b}) = \mathtt{a\,a}$$

is a morphism.

String Morphisms

$\Sigma := \{\mathtt{a}, \mathtt{b}\}$ is a finite alphabet.

$w = \mathtt{a\,b\,b\,a} \in \Sigma^*$ is a string.

$h : \Sigma \to \Sigma^*$ with

$$h(\mathtt{a}) = \mathtt{b\,b\,a}$$
$$h(\mathtt{b}) = \mathtt{a\,a}$$

is a morphism.

$h(w) = h(\mathtt{a\,b\,b\,a}) = h(\mathtt{a})\,h(\mathtt{b})\,h(\mathtt{b})\,h(\mathtt{a}) = \mathtt{b\,b\,a\,a\,a\,a\,a\,b\,b\,a}$.

String Morphisms

$\Sigma := \{\mathtt{a}, \mathtt{b}\}$ is a finite alphabet.

$w = \mathtt{a\,b\,b\,a} \in \Sigma^*$ is a string.

$h : \Sigma \to \Sigma^*$ with

$$h(\mathtt{a}) = \mathtt{b\,b\,a}$$
$$h(\mathtt{b}) = \mathtt{a\,a}$$

is a morphism.

$h(w) = h(\mathtt{a\,b\,b\,a}) = h(\mathtt{a})\,h(\mathtt{b})\,h(\mathtt{b})\,h(\mathtt{a}) = \mathtt{b\,b\,a\,a\,a\,a\,a\,b\,b\,a}$.

String Morphism Problem

Instance: Strings $u, w \in \Sigma^*$.

Question: Does there exist a morphism h with $h(u) = w$?

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Σ is the target alphabet of terminals

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Σ is the target alphabet of terminals

$u \in X^+$ is a source string

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Σ is the target alphabet of terminals

$u \in X^+$ is a source string

$w \in \Sigma^*$ is a target string

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Σ is the target alphabet of terminals

$u \in X^+$ is a source string

$w \in \Sigma^*$ is a target string

$h : X \to \Sigma^*$ is a morphism

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Σ is the target alphabet of terminals

$u \in X^+$ is a source string

$w \in \Sigma^*$ is a target string

$h : X \to \Sigma^*$ is a morphism

$h : (X \cup \Sigma) \to \Sigma^*$ with $h(a) = a$ for every $a \in \Sigma$ is a substitution

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Σ is the target alphabet of terminals

$u \in X^+$ is a source string

$w \in \Sigma^*$ is a target string

$h : X \to \Sigma^*$ is a morphism

$h : (X \cup \Sigma) \to \Sigma^*$ with $h(a) = a$ for every $a \in \Sigma$ is a substitution

h is non-erasing if $h(x) \neq \varepsilon$, $x \in X$

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Σ is the target alphabet of terminals

$u \in X^+$ is a source string

$w \in \Sigma^*$ is a target string

$h : X \to \Sigma^*$ is a morphism

$h : (X \cup \Sigma) \to \Sigma^*$ with $h(a) = a$ for every $a \in \Sigma$ is a substitution

h is non-erasing if $h(x) \neq \varepsilon$, $x \in X$

h is E-injective if $x \neq y$ and $\varepsilon \notin \{h(x), h(y)\}$ implies $h(x) \neq h(y)$

Some Notations and Definitions

$X = \{x_1, x_2, x_3, \ldots\}$ is the source alphabet of variables

Σ is the target alphabet of terminals

$u \in X^+$ is a source string

$w \in \Sigma^*$ is a target string

$h : X \to \Sigma^*$ is a morphism

$h : (X \cup \Sigma) \to \Sigma^*$ with $h(a) = a$ for every $a \in \Sigma$ is a substitution

h is non-erasing if $h(x) \neq \varepsilon$, $x \in X$

h is E-injective if $x \neq y$ and $\varepsilon \notin \{h(x), h(y)\}$ implies $h(x) \neq h(y)$

h is injective if it is non-erasing and E-injective

Some Examples

Example 1:

$$u = x_1\ x_1\ x_2\ x_3\ x_2$$

$$w = \mathtt{a\ b\ a\ b\ a\ b\ a\ b\ a\ b}$$

Some Examples

Example 1:

$$u = \texttt{a}\,\texttt{b}\,\texttt{a}\,\texttt{b}\,x_2\,x_3\,x_2$$

$$w = \texttt{a}\,\texttt{b}\,\texttt{a}\,\texttt{b}\,\texttt{a}\,\texttt{b}\,\texttt{a}\,\texttt{b}\,\texttt{a}\,\texttt{b}$$

Some Examples

Example 1:

$$u = \mathtt{a\,b\,a\,b\,a\,b}\,x_3\,\mathtt{a\,b}$$

$$w = \mathtt{a\,b\,a\,b\,a\,b\,a\,b\,a\,b}$$

Some Examples

Example 1:

$$u = \texttt{a b a b a b a b a b}$$
$$w = \texttt{a b a b a b a b a b}$$

Some Examples

Example 1:

$$u = \mathtt{a\,b\,a\,b\,a\,b\,a\,b\,a\,b}$$
$$w = \mathtt{a\,b\,a\,b\,a\,b\,a\,b\,a\,b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \mathtt{ababababab}$, h non-erasing, but not injective

Some Examples

Example 2:

$$u = x_1 \, \mathtt{a} \, x_2 \mathtt{b} \, x_2 x_1 \, x_2$$
$$w = \mathtt{b\,a\,c\,b\,a\,b\,b\,a\,c\,b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \mathtt{ababababab}$, h non-erasing, but not injective

Some Examples

Example 2:

$$u = \mathtt{b\,a\,c\,b\,a}\, x_2 \mathtt{b}\, x_2 \mathtt{b\,a\,c\,b}\, x_2$$

$$w = \mathtt{b\,a\,c\,b\,a\,b\,b\,a\,c\,b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \mathtt{ababababab}$, h non-erasing, but not injective

Some Examples

Example 2:

$$u = \text{bacbabbacb}$$
$$w = \text{bacbabbacb}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \text{ababababab}$, h non-erasing, but not injective

Some Examples

Example 2:

$$u = \texttt{b a c b a b b a c b}$$
$$w = \texttt{b a c b a b b a c b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \texttt{abababababab}$, h non-erasing, but not injective
Ex. 2: $h(x_1 \texttt{a} x_2 \texttt{b} x_2 x_1 x_2) = \texttt{bacbabbacb}$, h E-injective, but erasing

Some Examples

Example 3:

$$u = x_1 \, \mathtt{a} \, x_2 \, \mathtt{b} \, x_2 \, x_1 \, x_2$$

$$w = \mathtt{a\,b\,a\,a\,b\,b\,a\,b\,a\,b\,a\,b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \mathtt{abababababab}$, h non-erasing, but not injective
Ex. 2: $h(x_1 \mathtt{a} x_2 \mathtt{b} x_2 x_1 x_2) = \mathtt{bacbabbacb}$, h E-injective, but erasing

Some Examples

Example 3:

$$u = \mathtt{a}\,\mathtt{b}\,\mathtt{a}\,x_2\,\mathtt{b}\,x_2\,\mathtt{a}\,\mathtt{b}\,x_2$$

$$w = \mathtt{a}\,\mathtt{b}\,\mathtt{a}\,\mathtt{a}\,\mathtt{b}\,\mathtt{b}\,\mathtt{a}\,\mathtt{b}\,\mathtt{a}\,\mathtt{b}\,\mathtt{a}\,\mathtt{b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \mathtt{ababababab}$, h non-erasing, but not injective
Ex. 2: $h(x_1 \mathtt{a} x_2 \mathtt{b} x_2 x_1 x_2) = \mathtt{bacbabbacb}$, h E-injective, but erasing

Some Examples

Example 3:

$$u = \texttt{a b a a b b a b a b a b}$$
$$w = \texttt{a b a a b b a b a b a b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \texttt{abababababab}$, h non-erasing, but not injective
Ex. 2: $h(x_1 \texttt{a} x_2 \texttt{b} x_2 x_1 x_2) = \texttt{bacbabbacb}$, h E-injective, but erasing

Some Examples

Example 3:

$$u = \texttt{a b a a b b a b a b a b}$$

$$w = \texttt{a b a a b b a b a b a b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) =$ ababababab, h non-erasing, but not injective
Ex. 2: $h(x_1 \texttt{a} x_2 \texttt{b} x_2 x_1 x_2) =$ bacbabbacb, h E-injective, but erasing
Ex. 3: $h(x_1 \texttt{a} x_2 \texttt{b} x_2 x_1 x_2) =$ abaabbababab, h non-erasing, but not injective

Some Examples

Example 3:

$$u = \mathtt{a\,b\,a\,a\,b\,b\,a\,b\,a\,b\,a\,b}$$
$$w = \mathtt{a\,b\,a\,a\,b\,b\,a\,b\,a\,b\,a\,b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \mathtt{ababababab}$, h non-erasing, but not injective
Ex. 2: $h(x_1 \mathtt{a} x_2 \mathtt{b} x_2 x_1 x_2) = \mathtt{bacbabbacb}$, h E-injective, but erasing
Ex. 3: $h(x_1 \mathtt{a} x_2 \mathtt{b} x_2 x_1 x_2) = \mathtt{abaabbababab}$, h non-erasing, but not injective

Ex. 2: $\nexists$ non-erasing h with $h(x_1 \mathtt{a} x_2 \mathtt{b} x_2 x_1 x_2) = \mathtt{bacbabbacb}$

Some Examples

Example 3:

$$u = \text{a b a a b b a b a b a b}$$
$$w = \text{a b a a b b a b a b a b}$$

Ex. 1: $h(x_1 x_1 x_2 x_3 x_2) = \text{ababababab}$, h non-erasing, but not injective
Ex. 2: $h(x_1 \text{a} x_2 \text{b} x_2 x_1 x_2) = \text{bacbabbacb}$, h E-injective, but erasing
Ex. 3: $h(x_1 \text{a} x_2 \text{b} x_2 x_1 x_2) = \text{abaabbababab}$, h non-erasing, but not injective

Ex. 2: $\nexists$ non-erasing h with $h(x_1 \text{a} x_2 \text{b} x_2 x_1 x_2) = \text{bacbabbacb}$
Ex. 3: $\nexists$ injective h with $h(x_1 \text{a} x_2 \text{b} x_2 x_1 x_2) = \text{abaabbababab}$

Different Versions of String Morphism Problems

StrMorph

Instance: Two strings $u \in X^*$ *and* $w \in \Sigma^*$.
Question: Does there exist a morphism h *with* $h(u) = w$*?*

Different Versions of String Morphism Problems

StrMorph

Instance: Two strings $u \in X^*$ *and* $w \in \Sigma^*$.
Question: Does there exist a morphism h *with* $h(u) = w$*?*

StrSubst

Instance: Two strings $u \in (X \cup \Sigma)^*$ *and* $w \in \Sigma^*$.
Question: Does there exist a substitution h *with* $h(u) = w$*?*

Different Versions of String Morphism Problems

StrMorph

Instance: Two strings $u \in X^*$ and $w \in \Sigma^*$.
Question: Does there exist a morphism h with $h(u) = w$?

StrSubst

Instance: Two strings $u \in (X \cup \Sigma)^*$ and $w \in \Sigma^*$.
Question: Does there exist a substitution h with $h(u) = w$?

For $K \in \{\text{StrMorph}, \text{StrSubst}\}$,
Ne-K denotes the non-erasing version of K,
Inj-K denotes the $\mathbb{E}$-injective version of K,
Ne-Inj-K denotes the non-erasing injective version of K.

Different Versions of String Morphism Problems

StrMorph

Instance: Two strings $u \in X^*$ and $w \in \Sigma^*$.
Question: Does there exist a *morphism* h with $h(u) = w$?

StrSubst

Instance: Two strings $u \in (X \cup \Sigma)^*$ and $w \in \Sigma^*$.
Question: Does there exist a *substitution* h with $h(u) = w$?

For $K \in \{\text{StrMorph}, \text{StrSubst}\}$,
Ne-K denotes the non-erasing version of K,
Inj-K denotes the $\mathbb{E}$-injective version of K,
Ne-Inj-K denotes the non-erasing injective version of K.

$\text{SMP} := \{Z\text{-}\text{StrMorph}, Z\text{-}\text{StrSubst} \mid Z \in \{\varepsilon, \text{Ne}, \text{Inj}, \text{Ne-Inj}\}\}$.

Applications

- Theoretical: Inductive inference (of Angluin's Pattern languages, computational aspects of string morphisms, parameterised pattern matching).
- Practical: Matchtest for regular expressions with backreferences (as implemented in Perl, Java, Python, ...).

NP-Completeness

Theorem (Angluin 1980; Ehrenfeucht and Rozenberg 1979; Clifford, Harrow, Popa and Sach 2009; Fernau and S. 2013; . . .)

All versions of the string morphism problem are NP-complete.

Some More Notation

For any source string u (e. g., $u := x_1\mathtt{a}x_2x_1\mathtt{ba}x_2x_1x_3$),

Some More Notation

For any source string u (e. g., $u := x_1 \mathtt{a} x_2 x_1 \mathtt{ba} x_2 x_1 x_3$),

var(u) is the set of variables in u, e. g. $\mathsf{var}(u) = \{x_1, x_2, x_3\}$

Some More Notation

For any source string u (e. g., $u := x_1 \mathtt{a} x_2 x_1 \mathtt{ba} x_2 x_1 x_3$),

var(u) is the set of variables in u, e. g. $\text{var}(u) = \{x_1, x_2, x_3\}$

$|u|_x$ is the number of Occ. of x in u, e. g. $|u|_{x_1} = 3$

Different Versions

Types of string morphism problems:

- StrMorph, StrSubst.
- Ne-StrMorph, Ne-StrSubst.
- Inj-StrMorph, Inj-StrSubst.
- Ne-Inj-StrMorph, Ne-Inj-StrSubst.

Parameters of string morphism problems:

$|\mathrm{var}(u)|$ Number of variables in the source string.

Different Versions

Types of string morphism problems:

- StrMorph, StrSubst.
- Ne-StrMorph, Ne-StrSubst.
- Inj-StrMorph, Inj-StrSubst.
- Ne-Inj-StrMorph, Ne-Inj-StrSubst.

Parameters of string morphism problems:

$\lvert \mathrm{var}(u) \rvert$	Number of variables in the source string.
$\lvert w \rvert$	Length of the target string.

Different Versions

Types of string morphism problems:

- StrMorph, StrSubst.
- Ne-StrMorph, Ne-StrSubst.
- Inj-StrMorph, Inj-StrSubst.
- Ne-Inj-StrMorph, Ne-Inj-StrSubst.

Parameters of string morphism problems:

$\lvert \mathrm{var}(u) \rvert$	Number of variables in the source string.
$\lvert w \rvert$	Length of the target string.
$\lvert h(x) \rvert$	Max. length of substitution words.

Different Versions

Types of string morphism problems:

- StrMorph, StrSubst.
- Ne-StrMorph, Ne-StrSubst.
- Inj-StrMorph, Inj-StrSubst.
- Ne-Inj-StrMorph, Ne-Inj-StrSubst.

Parameters of string morphism problems:

$\lvert \mathrm{var}(u) \rvert$	Number of variables in the source string.
$\lvert w \rvert$	Length of the target string.
$\lvert h(x) \rvert$	Max. length of substitution words.
$\lvert u \rvert_x$	Max. occ. per variable.

Different Versions

Types of string morphism problems:

- StrMorph, StrSubst.
- Ne-StrMorph, Ne-StrSubst.
- Inj-StrMorph, Inj-StrSubst.
- Ne-Inj-StrMorph, Ne-Inj-StrSubst.

Parameters of string morphism problems:

$\lvert \mathrm{var}(u) \rvert$	Number of variables in the source string.
$\lvert w \rvert$	Length of the target string.
$\lvert h(x) \rvert$	Max. length of substitution words.
$\lvert u \rvert_x$	Max. occ. per variable.
$\lvert \Sigma \rvert$	Size of target alphabet.

Different Versions

Types of string morphism problems:

- StrMorph, StrSubst.
- Ne-StrMorph, Ne-StrSubst.
- Inj-StrMorph, Inj-StrSubst.
- Ne-Inj-StrMorph, Ne-Inj-StrSubst.

Parameters of string morphism problems:

$\lvert \mathsf{var}(u) \rvert$	Number of variables in the source string.
$\lvert w \rvert$	Length of the target string.
$\lvert h(x) \rvert$	Max. length of substitution words.
$\lvert u \rvert_x$	Max. occ. per variable.
$\lvert \Sigma \rvert$	Size of target alphabet.

2^3 types, 2^5 combinations of parameters $\rightarrow$
256 parametrised versions of string morphism problems.

Parameterised Complexity Theory

Parameterised problems Instances are of form (x, k), where $k \in \mathbb{N}$ is the parameter.

Parameterised Complexity Theory

Parameterised problems Instances are of form (x, k), where $k \in \mathbb{N}$ is the parameter.

FPT Parameterised problems solvable in time

$$f(k) \times |x|^{O(1)},$$

where k is parameter and f is computable.

Parameterised Complexity Theory

Parameterised problems Instances are of form (x, k), where $k \in \mathbb{N}$ is the parameter.

FPT Parameterised problems solvable in time

$$f(k) \times |x|^{O(1)},$$

where k is parameter and f is computable.

Parameterised reduction Reduction from (P, k) to (P', k') that runs in "FPT"-time (with respect to k) and k' only depends on k.

Parameterised Complexity Theory

Parameterised problems Instances are of form (x, k), where $k \in \mathbb{N}$ is the parameter.

FPT Parameterised problems solvable in time

$$f(k) \times |x|^{O(1)},$$

where k is parameter and f is computable.

Parameterised reduction Reduction from (P, k) to (P', k') that runs in "FPT"-time (with respect to k) and k' only depends on k.

W[1] Parameterised problems as hard as k-Clique.

Parameterised Complexity Theory

Parameterised problems	Instances are of form (x, k), where $k \in \mathbb{N}$ is the parameter.
FPT	Parameterised problems solvable in time $$f(k) \times \lvert x \rvert^{O(1)},$$ where k is parameter and f is computable.
Parameterised reduction	Reduction from (P, k) to (P', k') that runs in "FPT"-time (with respect to k) and k' only depends on k.
W[1]	Parameterised problems as hard as k-Clique.
W[1]-hard problems	Parameterised problems hard for $W[1]$. ("Fixed parameter intractable problems").

Summary

Problems	$\lvert \mathrm{var}(u) \rvert$	$\lvert w \rvert$	$\lvert u \rvert_{\mathrm{var}}$	$\lvert h \rvert$	$\lvert \Sigma \rvert$	Complexity
SMP	p	p	–	–	–	FPT
{Ne, Ne - Inj}-{StrMorph, StrSubst}	–	p	–	–	–	FPT
SMP	p	–	–	p	–	FPT
SMP	p	–	p	–	6	$W[1]$-hard
{ε, Inj}-{StrMorph, StrSubst}	–	p	3	1	p	$W[1]$-hard

Summary

Problems	$\lvert\text{var}(u)\rvert$	$\lvert w\rvert$	$\lvert u\rvert_{\text{var}}$	$\lvert h\rvert$	$\lvert\Sigma\rvert$	Complexity
SMP	p	p	–	–	–	FPT
$\{$Ne, Ne-Inj$\}$-$\{$StrMorph, StrSubst$\}$	–	p	–	–	–	FPT
SMP	p	–	–	p	–	FPT
SMP	p	–	p	–	6	$W[1]$-hard
$\{\varepsilon$, Inj$\}$-$\{$StrMorph, StrSubst$\}$	–	p	3	1	p	$W[1]$-hard

Summary

Problems	$\lvert var(u) \rvert$	$\lvert w \rvert$	$\lvert u \rvert_{var}$	$\lvert h \rvert$	$\lvert \Sigma \rvert$	Complexity
SMP	p	p	–	–	–	FPT
$\{$Ne, Ne - Inj$\}$-$\{$StrMorph, StrSubst$\}$	–	p	–	–	–	FPT
SMP	p	–	–	p	–	FPT
SMP	p	–	p	–	6	$W[1]$-hard
$\{\varepsilon$, Inj$\}$-$\{$StrMorph, StrSubst$\}$	–	p	3	1	p	$W[1]$-hard

Summary

Problems	$\lvert var(u) \rvert$	$\lvert w \rvert$	$\lvert u \rvert_{var}$	$\lvert h \rvert$	$\lvert \Sigma \rvert$	Complexity
SMP	p	p	–	–	–	FPT
{Ne, Ne - Inj}-{StrMorph, StrSubst}	–	p	–	–	–	FPT
SMP	p	–	–	p	–	FPT
SMP	p	–	p	–	6	$W[1]$-hard
$\{\varepsilon$, Inj$\}$-{StrMorph, StrSubst}	–	p	3	1	p	$W[1]$-hard

Summary

Problems	$\lvert var(u) \rvert$	$\lvert w \rvert$	$\lvert u \rvert_{var}$	$\lvert h \rvert$	$\lvert \Sigma \rvert$	Complexity
SMP	p	p	–	–	–	FPT
{Ne, Ne - Inj}-{StrMorph, StrSubst}	–	p	–	–	–	FPT
SMP	p	–	–	p	–	FPT
SMP	p	–	p	–	6	$W[1]$-hard
{ε, Inj}-{StrMorph, StrSubst}	–	p	3	1	p	$W[1]$-hard

Summary

Problems	$\lvert\mathrm{var}(u)\rvert$	$\lvert w\rvert$	$\lvert u\rvert_{\mathrm{var}}$	$\lvert h\rvert$	$\lvert\Sigma\rvert$	Complexity
SMP	p	p	–	–	–	FPT
{Ne, Ne - Inj}-{StrMorph, StrSubst}	–	p	–	–	–	FPT
SMP	p	–	–	p	–	FPT
SMP	p	–	p	–	6	$W[1]$-hard
$\{\varepsilon$, Inj$\}$-{StrMorph, StrSubst}	–	p	3	1	p	$W[1]$-hard

Fixed Parameter Intractability

Theorem (Stephan, Yoshinaka, Zeugmann)

The problem Ne-StrSubst, *parameterised by*

- *number of variables* ($|\mathrm{var}(u)|$),
- *cardinality of target alphabet* ($|\Sigma|$) *and*
- *maximum occurrences per variable* ($|u|_{\mathrm{var}}$),

is $W[1]$*-hard.*

Fixed Parameter Intractability

Theorem (Stephan, Yoshinaka, Zeugmann)

The problem Ne-StrSubst*, parameterised by*

- *number of variables* ($|\mathsf{var}(u)|$),
- *cardinality of target alphabet* ($|\Sigma|$) *and*
- *maximum occurrences per variable* ($|u|_{\mathsf{var}}$),

is $W[1]$*-hard.*

Theorem

For every $K \in$ SMP*, the problem* K*, parameterised by*

- *number of variables* ($|\mathsf{var}(u)|$),
- *cardinality of target alphabet* ($|\Sigma|$) *and*
- *maximum occurrences per variable* ($|u|_{\mathsf{var}}$),

is $W[1]$*-hard.*

Fixed Parameter Intractability

Theorem

The problems StrMorph, Inj-StrMorph, StrSubst *and* Inj-StrSubst, *parameterised by*

- *length of the target string* ($|w|$),
- *cardinality of target alphabet* ($|\Sigma|$),
- *maximum occurrences per variable* ($|u|_{\mathsf{var}}$) *and*
- *maximum length of substitution words* ($|h(x)|$),

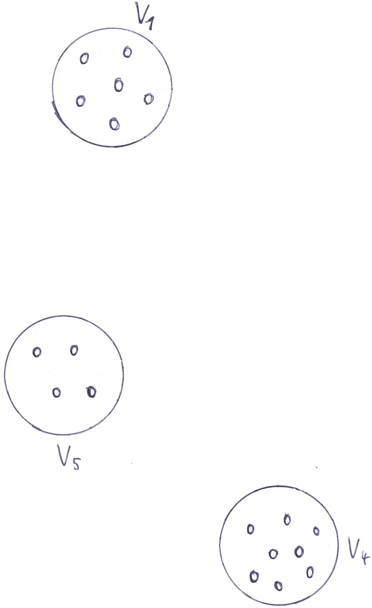
are $W[1]$*-hard.*

k-Multicoloured-Clique

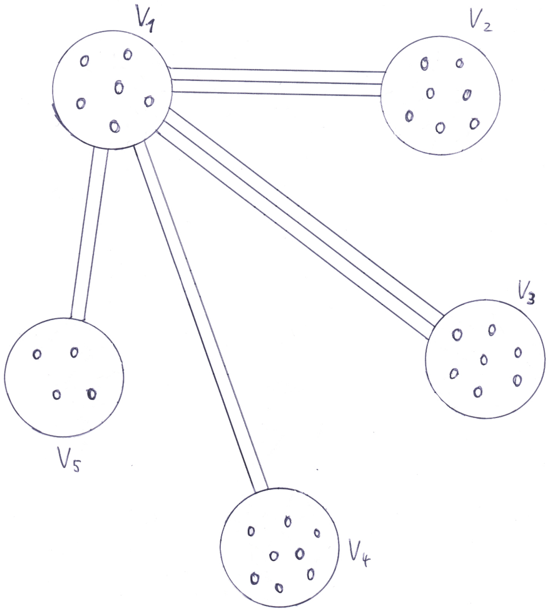
k-Multicoloured-Clique

Instance	A graph $\mathcal{G} := (V, E)$ and a partition $V_1, V_2, \ldots, V_k$ of V, such that every V_i is an independent set.
Parameter	k.
Question	Does $\mathcal{G}$ has a clique of size k?

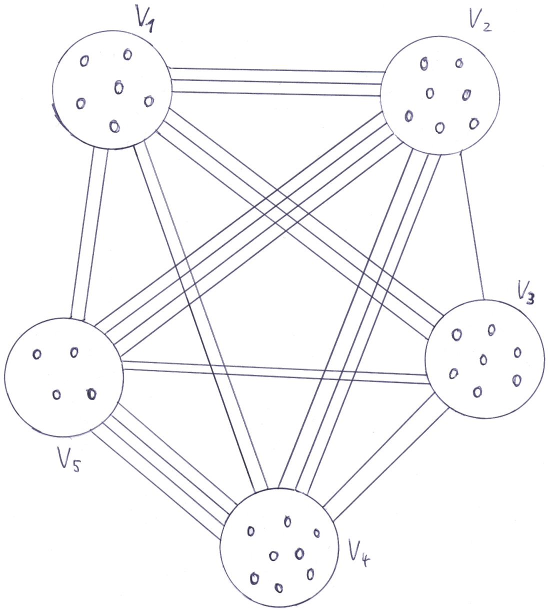
k-Multicoloured-Clique



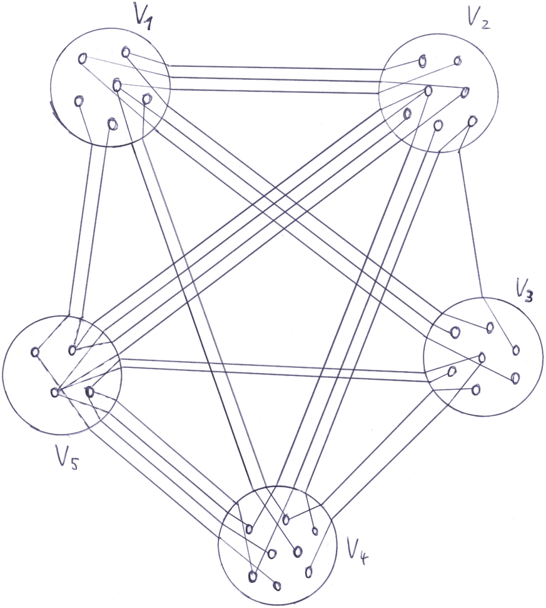
k-Multicoloured-Clique



k-Multicoloured-Clique



k-Multicoloured-Clique



k-Multicoloured-Clique

k-Multicoloured-Clique

Instance	A graph $\mathcal{G} := (V, E)$ and a partition $V_1, V_2, \ldots, V_k$ of V, such that every V_i is an independent set.
Parameter	k.
Question	Does $\mathcal{G}$ has a clique of size k?

k-Multicoloured-Clique

k-Multicoloured-Clique

Instance A graph $\mathcal{G} := (V, E)$ and a partition $V_1, V_2, \ldots, V_k$ of V, such that every V_i is an independent set.

Parameter k.

Question Does $\mathcal{G}$ has a clique of size k?

Theorem (Fellows, Hermelin, Rosamond, and Vialette)

k-Multicoloured-Clique *is* $W[1]$*-hard.*

Sketch of the Reduction

$(\mathcal{G}, V_1, V_2, \ldots, V_k)$ is a k-Multicoloured-Clique instance.
$\mathcal{G} = (V, E)$, $V_i := \{v_{i,1}, v_{i,2}, \ldots, v_{i,t_i}\}$.

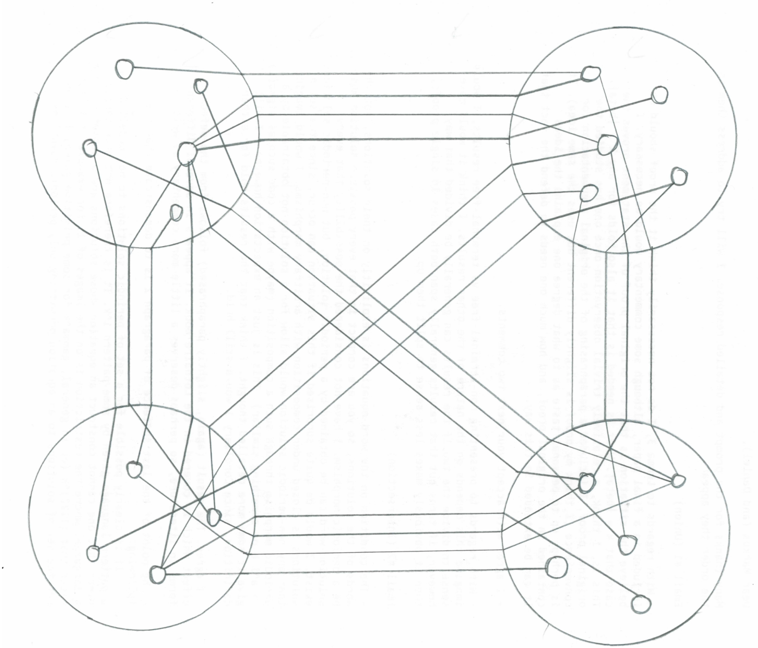
Sketch of the Reduction

$(\mathcal{G}, V_1, V_2, \ldots, V_k)$ is a k-Multicoloured-Clique instance.
$\mathcal{G} = (V, E)$, $V_i := \{v_{i,1}, v_{i,2}, \ldots, v_{i,t_i}\}$.

Target alphabet: $\Sigma := \{\mathtt{a}_{\{i,j\}} \mid 1 \leq i \leq j \leq k, i \neq j\} \cup \{\$\}$,
source alphabet: $X := \{x_e \mid e \in E\}$.

Important: $|\Sigma| = \mathrm{O}(k^2)$

Sketch of the Reduction - Main Gadget



Sketch of the Reduction - Main Gadget

Let $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$, and let $e_1, e_2, \ldots, e_l$ be exactly the edges connecting a vertex from V_i with a vertex from V_j.

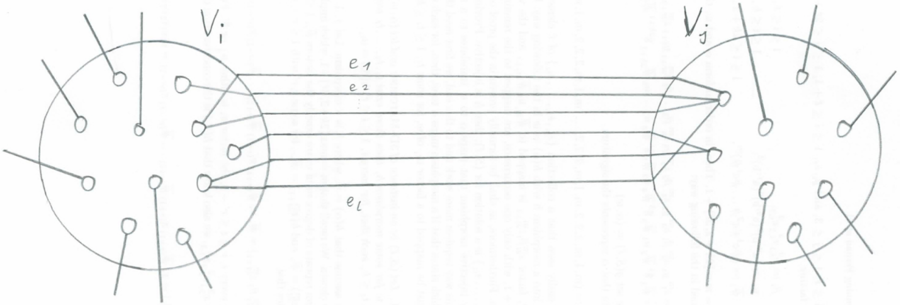
Sketch of the Reduction - Main Gadget

Let $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$, and let $e_1, e_2, \ldots, e_l$ be exactly the edges connecting a vertex from V_i with a vertex from V_j.

$$\overline{u}_{i,j} := \$\, x_{e_1}\, x_{e_2} \ldots x_{e_l}\, \$\,,$$

$$\overline{w}_{i,j} := \$\, \mathsf{a}_{\{i,j\}}\, \$\,,$$

Sketch of the Reduction - Main Gadget



Sketch of the Reduction - Main Gadget

Let $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$, and let $e_1, e_2, \ldots, e_l$ be exactly the edges connecting a vertex from V_i with a vertex from V_j.

$$\overline{u}_{i,j} := \$\, x_{e_1}\, x_{e_2} \ldots x_{e_l}\, \$\,,$$
$$\overline{w}_{i,j} := \$\, \mathsf{a}_{\{i,j\}}\, \$\,,$$

Sketch of the Reduction - Main Gadget

Let $i, j \in \{1, 2, \ldots, k\}$, $i \neq j$, and let $e_1, e_2, \ldots, e_l$ be exactly the edges connecting a vertex from V_i with a vertex from V_j.

$$\overline{u}_{i,j} := \$\, x_{e_1}\, x_{e_2} \ldots x_{e_l}\, \$\,,$$
$$\overline{w}_{i,j} := \$\, \mathsf{a}_{\{i,j\}}\, \$\,,$$

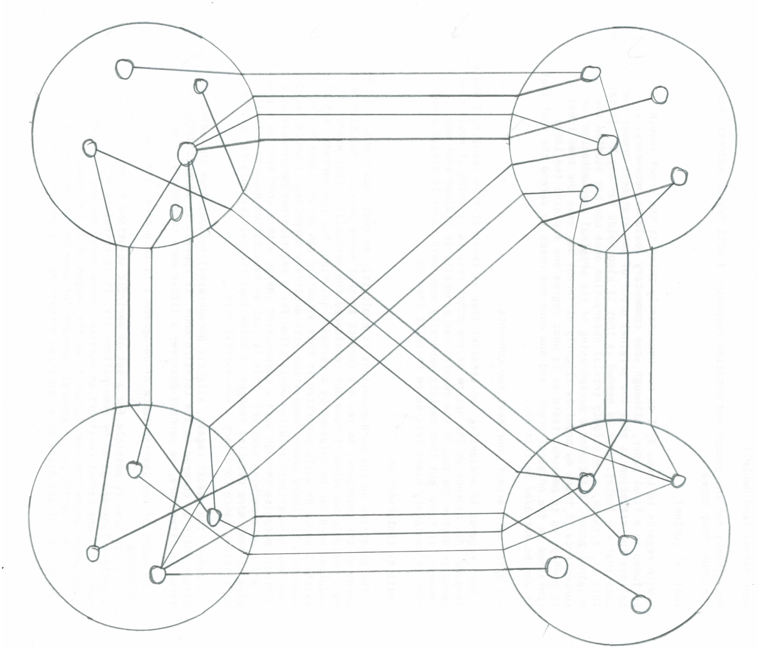
We combine all these gadgets to one big gadget:

$$\overline{u} := \overline{u}_{1,2}\, \overline{u}_{1,3} \ldots \overline{u}_{1,k}\, \overline{u}_{2,3}\, \overline{u}_{2,4} \ldots \overline{u}_{2,k}\ \ldots\ \overline{u}_{k-1,k}\,,$$
$$\overline{w} := \overline{w}_{1,2}\, \overline{w}_{1,3} \ldots \overline{w}_{1,k}\, \overline{w}_{2,3}\, \overline{w}_{2,4} \ldots \overline{w}_{2,k}\ \ldots\ \overline{w}_{k-1,k}\,.$$

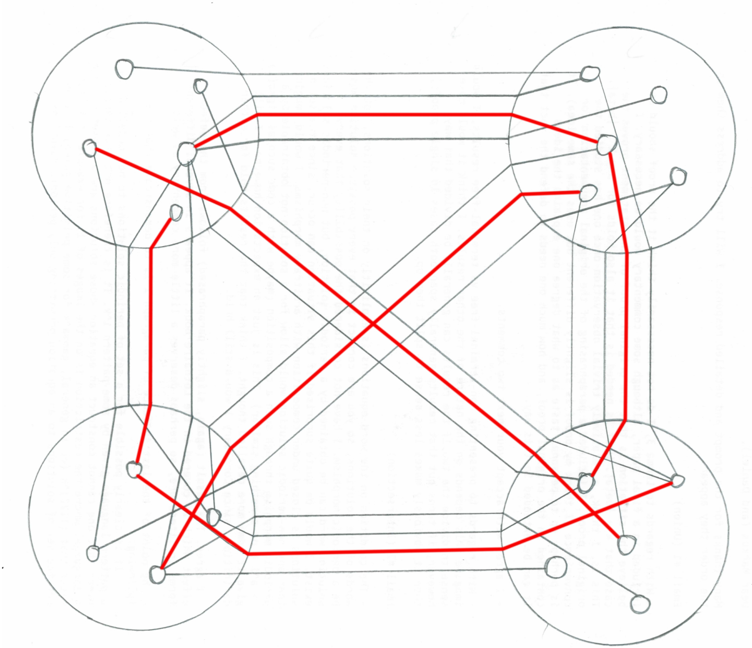
Important:

- $|\overline{w}| = O(k^2)$,
- every variable has 1 occurrences.

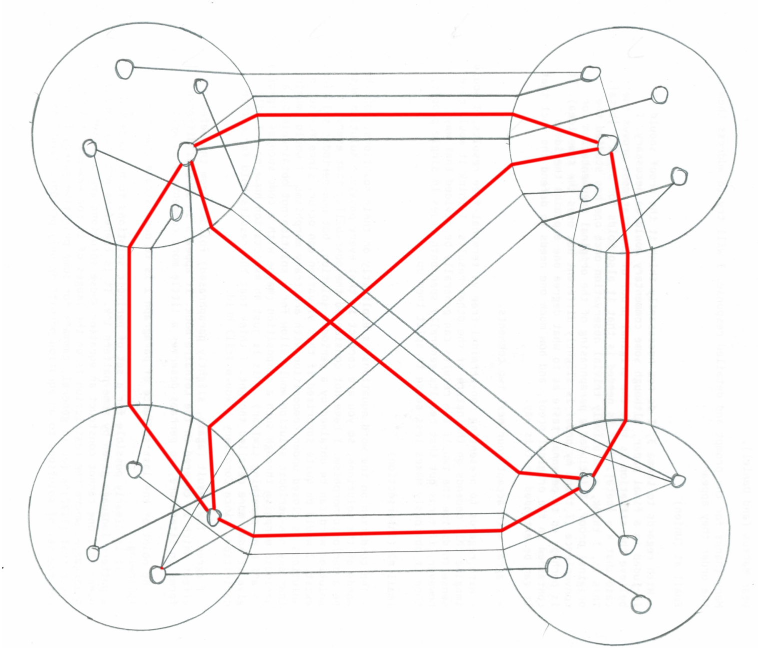
Sketch of the Reduction - Main Gadget



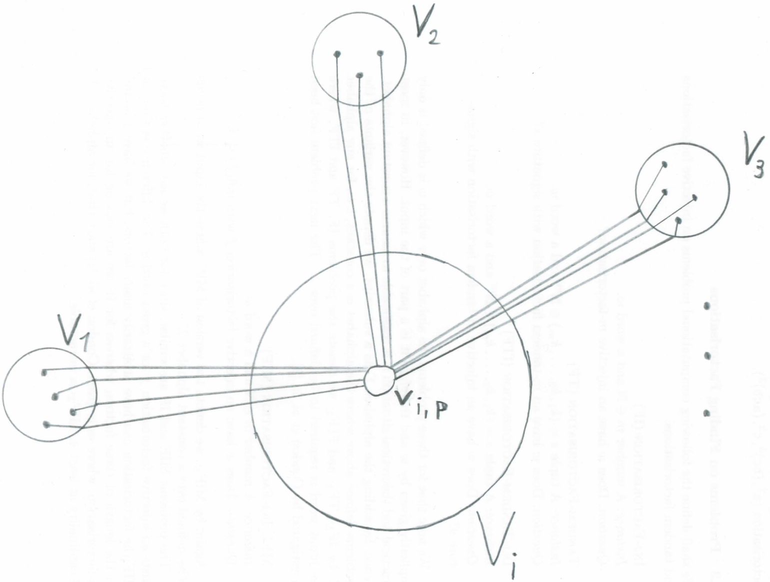
Sketch of the Reduction - Main Gadget



Sketch of the Reduction - Main Gadget



Sketch of the Reduction - Enforcer Gadget



Sketch of the Reduction - Enforcer Gadget

$e_{j,1}, e_{j,2}, \ldots, e_{j,q_j}$ are the edges between $v_{i,p}$ and a vertex in V_j.

Sketch of the Reduction - Enforcer Gadget

$e_{j,1}, e_{j,2}, \ldots, e_{j,q_j}$ are the edges between $v_{i,p}$ and a vertex in V_j.

$$\widehat{u}_{i,p} := x_{e_{1,1}} x_{e_{1,2}} \ldots x_{e_{1,q_1}} x_{e_{2,1}} x_{e_{2,2}} \ldots x_{e_{2,q_2}} \ldots x_{e_{k,1}} x_{e_{k,2}} \ldots x_{e_{k,q_k}} ,$$

Sketch of the Reduction - Enforcer Gadget

$e_{j,1}, e_{j,2}, \ldots, e_{j,q_j}$ are the edges between $v_{i,p}$ and a vertex in V_j.

$$\widehat{u}_{i,p} := x_{e_{1,1}} x_{e_{1,2}} \ldots x_{e_{1,q_1}} x_{e_{2,1}} x_{e_{2,2}} \ldots x_{e_{2,q_2}} \ldots x_{e_{k,1}} x_{e_{k,2}} \ldots x_{e_{k,q_k}} ,$$

$$\widetilde{w}_i := \mathtt{a}_{\{i,1\}} \, \mathtt{a}_{\{i,2\}} \ldots \mathtt{a}_{\{i,k\}} .$$

Sketch of the Reduction - Enforcer Gadget

$e_{j,1}, e_{j,2}, \ldots, e_{j,q_j}$ are the edges between $v_{i,p}$ and a vertex in V_j.

$$\begin{aligned} \widehat{u}_{i,p} &:= \quad x_{e_{1,1}} x_{e_{1,2}} \ldots x_{e_{1,q_1}} \quad x_{e_{2,1}} x_{e_{2,2}} \ldots x_{e_{2,q_2}} \quad \ldots \quad x_{e_{k,1}} x_{e_{k,2}} \ldots x_{e_{k,q_k}}\,, \\ \widetilde{w}_i &:= \quad \mathtt{a}_{\{i,1\}} \quad \mathtt{a}_{\{i,2\}} \quad \ldots \quad \mathtt{a}_{\{i,k\}}\,. \end{aligned}$$

Sketch of the Reduction - Enforcer Gadget

$e_{j,1}, e_{j,2}, \ldots, e_{j,q_j}$ are the edges between $v_{i,p}$ and a vertex in V_j.

$$\begin{aligned} \widehat{u}_{i,p} &:= && x_{e_{1,1}} x_{e_{1,2}} \ldots x_{e_{1,q_1}} && x_{e_{2,1}} x_{e_{2,2}} \ldots x_{e_{2,q_2}} && \ldots && x_{e_{k,1}} x_{e_{k,2}} \ldots x_{e_{k,q_k}}, \\ \widetilde{w}_i &:= && \mathtt{a}_{\{i,1\}} && \mathtt{a}_{\{i,2\}} && \ldots && \mathtt{a}_{\{i,k\}}. \end{aligned}$$

$$\begin{aligned} \widetilde{u}_i &:= \widehat{u}_{i,1}\, \widehat{u}_{i,2} \ldots \widehat{u}_{i,p} \ldots \widehat{u}_{i,t_i}, \\ \widetilde{w}_i &:= \mathtt{a}_{\{i,1\}}\, \mathtt{a}_{\{i,2\}} \ldots \mathtt{a}_{\{i,i-1\}}\, \mathtt{a}_{\{i,i+1\}}\, \mathtt{a}_{\{i,i+2\}} \ldots \mathtt{a}_{\{i,k\}}. \end{aligned}$$

Sketch of the Reduction - Enforcer Gadget

$e_{j,1}, e_{j,2}, \ldots, e_{j,q_j}$ are the edges between $v_{i,p}$ and a vertex in V_j.

$$\begin{aligned} \widehat{u}_{i,p} &:= \quad x_{e_{1,1}} x_{e_{1,2}} \ldots x_{e_{1,q_1}} \quad x_{e_{2,1}} x_{e_{2,2}} \ldots x_{e_{2,q_2}} \quad \ldots \quad x_{e_{k,1}} x_{e_{k,2}} \ldots x_{e_{k,q_k}}, \\ \widetilde{w}_i &:= \quad \mathtt{a}_{\{i,1\}} \quad \mathtt{a}_{\{i,2\}} \quad \ldots \quad \mathtt{a}_{\{i,k\}}. \end{aligned}$$

$$\begin{aligned} \widetilde{u}_i &:= (\widehat{u}_{i,1})^2 (\widehat{u}_{i,2})^2 \ldots (\widehat{u}_{i,p})^2 \ldots (\widehat{u}_{i,t_i})^2, \\ \widetilde{w}_i &:= (\mathtt{a}_{\{i,1\}} \mathtt{a}_{\{i,2\}} \ldots \mathtt{a}_{\{i,i-1\}} \mathtt{a}_{\{i,i+1\}} \mathtt{a}_{\{i,i+2\}} \ldots \mathtt{a}_{\{i,k\}})^2 . \end{aligned}$$

Sketch of the Reduction - Enforcer Gadget

$e_{j,1}, e_{j,2}, \dots, e_{j,q_j}$ are the edges between $v_{i,p}$ and a vertex in V_j.

$$\begin{array}{rllll} \widehat{u}_{i,p} := & x_{e_{1,1}} x_{e_{1,2}} \dots x_{e_{1,q_1}} & x_{e_{2,1}} x_{e_{2,2}} \dots x_{e_{2,q_2}} & \dots & x_{e_{k,1}} x_{e_{k,2}} \dots x_{e_{k,q_k}}\,, \\ \widetilde{w}_i := & \mathtt{a}_{\{i,1\}} & \mathtt{a}_{\{i,2\}} & \dots & \mathtt{a}_{\{i,k\}}\,. \end{array}$$

$$\widetilde{u}_i := (\widehat{u}_{i,1})^2\,(\widehat{u}_{i,2})^2 \dots (\widehat{u}_{i,p})^2 \dots (\widehat{u}_{i,t_i})^2,$$

$$\widetilde{w}_i := (\mathtt{a}_{\{i,1\}}\,\mathtt{a}_{\{i,2\}} \dots \mathtt{a}_{\{i,i-1\}}\,\mathtt{a}_{\{i,i+1\}}\,\mathtt{a}_{\{i,i+2\}} \dots \mathtt{a}_{\{i,k\}})^2\,.$$

Enforcer Gadget:

$$\widetilde{u} := \$\,\widetilde{u}_1\,\$\,\widetilde{u}_2\,\$ \dots \$\,\widetilde{u}_k\,\$\,,$$

$$\widetilde{w} := \$\,\widetilde{w}_1\,\$\,\widetilde{w}_2\,\$ \dots \$\,\widetilde{w}_k\,\$\,.$$

Important:

- $|\widetilde{w}| = O(k^2)$,
- every variable has 2 occurrences.

Results not covered in this talk

Theorem ($W[1]$- and $W[P]$-Membership)

All $K \in \mathsf{SMP}$

- *parameterised by $|\mathsf{var}(u)|$ and $|u|_{\mathsf{var}}$ are in $W[1]$,*
- *parameterised by $|w|$ are in $W[1]$,*
- *parameterised by $|\mathsf{var}(u)|$ are in $W[P]$.*

Theorem (ETH Lower Bound)

For every $K \in \{\mathsf{StrMorph}, \mathsf{StrSubst}\}$ and $k_1, k_2 \in \mathbb{N}$, $k_2 \geq 2$, problem K, where $|h|$, $|\Sigma|$ are bounded by constants k_1, k_2, resp., cannot be solved in time $(|u||w|)^{O(1)} \times 2^{o(|\mathsf{var}(u)|)}$, unless ETH fails.

Thank you very much for your attention.