Pattern Matching with Variables: A Multivariate Complexity Analysis

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

Presented 19 June 2013 at CPM

Words with Coloured Holes

A word with (coloured) *holes*. . .

`a b □ c □ c b □ b □ c a □`

Words with Coloured Holes

A word with (coloured) *holes*...

$$\mathtt{a\,b}\,\square\,\mathtt{c}\,\square\,\mathtt{c\,b}\,\square\,\mathtt{b}\,\square\,\mathtt{c\,a}\,\square$$

...can be repaired...

$$\mathtt{a\,b\,a} \to \square$$
$$\mathtt{c\,b} \to \square$$

Words with Coloured Holes

A word with (coloured) *holes*...

$$\mathtt{a\,b}\;{\color{blue}\square}\;\mathtt{c}\;{\color{green}\square}\;\mathtt{c\,b}\;{\color{blue}\square}\;\mathtt{b}\;{\color{green}\square}\;\mathtt{c\,a}\;{\color{green}\square}$$

...can be repaired...

$$\mathtt{a\,b\,a} \to {\color{blue}\square}$$

$$\mathtt{c\,b} \to {\color{green}\square}$$

...by filling in new words:

$$\mathtt{a\,b}\,{\color{blue}\boxed{\mathtt{a\,b\,a}}}\,\mathtt{c}\,{\color{green}\boxed{\mathtt{c\,b}}}\,\mathtt{c\,b}\,{\color{blue}\boxed{\mathtt{a\,b\,a}}}\,\mathtt{b}\,{\color{green}\boxed{\mathtt{c\,b}}}\,\mathtt{c\,a}\,{\color{green}\boxed{\mathtt{c\,b}}}$$

A Special Kind of Pattern Matching

For given

α (a word with coloured holes),

w (a word without holes),

is it possible to fill the holes of α in such a way that we obtain w?

A Special Kind of Pattern Matching

For given

α (a word with coloured holes),

w (a word without holes),

is it possible to fill the holes of α in such a way that we obtain w?

Example 1:

$$\alpha = {\color{green}\square}\, \mathtt{a}\, \mathtt{a}\, {\color{blue}\square}\, {\color{green}\square}\, {\color{blue}\square}\, \mathtt{c}\, \mathtt{b}\, {\color{green}\square}$$

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

A Special Kind of Pattern Matching

For given

α (a word with coloured holes),

w (a word without holes),

is it possible to fill the holes of α in such a way that we obtain w?

Example 1:

$$\alpha = \mathtt{a\,c\,a\,a}\,\square\,\mathtt{a\,c}\,\square\,\mathtt{c\,b\,a\,c}$$

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

A Special Kind of Pattern Matching

For given

α (a word with coloured holes),

w (a word without holes),

is it possible to fill the holes of α in such a way that we obtain w?

Example 1:

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

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

A Special Kind of Pattern Matching

For given

α (a word with coloured holes),

w (a word without holes),

is it possible to fill the holes of α in such a way that we obtain w?

Example 2:

$$\alpha = {\color{green}\square}\, \mathtt{a}\, \mathtt{a}\, {\color{blue}\square}\, {\color{green}\square}\, {\color{blue}\square}\, \mathtt{c}\, \mathtt{b}\, {\color{green}\square}$$

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

A Special Kind of Pattern Matching

For given

α (a word with coloured holes),

w (a word without holes),

is it possible to fill the holes of α in such a way that we obtain w?

Example 2:

$$\alpha = \text{c c b a a} \square \text{c c b} \square \text{c b c c b}$$

$$v = \text{c c b a a c c b c b c c b}$$

A Special Kind of Pattern Matching

For given

α (a word with coloured holes),

w (a word without holes),

is it possible to fill the holes of α in such a way that we obtain w?

Example 2:

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

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

A Special Kind of Pattern Matching

For given

α (a word with coloured holes),

w (a word without holes),

is it possible to fill the holes of α in such a way that we obtain w?

Example 3:

$$\alpha = \square\, \mathtt{a}\, \mathtt{a}\, \square\, \square\, \square\, \mathtt{c}\, \mathtt{b}\, \square$$

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

Some Notations and Definitions

Σ is a terminal alphabet, $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$

Some Notations and Definitions

Σ is a terminal alphabet, $\Sigma = \{\texttt{a}, \texttt{b}, \texttt{c}\}$

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

Some Notations and Definitions

Σ is a terminal alphabet, $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$

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

$w \in \Sigma^*$ is a word `abaacba`

Some Notations and Definitions

Σ is a terminal alphabet, $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$

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

$w \in \Sigma^*$ is a word $\mathtt{abaacba}$

$\alpha \in (\Sigma \cup X)^+$ is a pattern $\alpha := x_1\mathtt{a}x_2x_1\mathtt{ba}x_2x_1x_3$

Some Notations and Definitions

Σ	is a terminal alphabet,	$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
X	is the set of variables,	$X = \{x_1, x_2, x_3, \ldots\}$
$w \in \Sigma^*$	is a word	$\mathtt{abaacba}$
$\alpha \in (\Sigma \cup X)^+$	is a pattern	$\alpha := x_1 \mathtt{a} x_2 x_1 \mathtt{ba} x_2 x_1 x_3$
$X \to \Sigma^+$	is a substitution	$h(x_1) := \mathtt{ab},\ h(x_2) := \mathtt{bcc}$

Pattern Matching with Variables

VPatMatch
Instance: A pattern $\alpha \in (\Sigma \cup X)^*$*, a word* $w \in \Sigma^*$*.*
Question: Does there exist a substitution h *with* $h(\alpha) = w$*?*

Pattern Matching with Variables

VPatMatch
Instance: A pattern $\alpha \in (\Sigma \cup X)^*$*, a word* $w \in \Sigma^*$*.*
Question: Does there exist a substitution h *with* $h(\alpha) = w$*?*

Two variants:

E-VPatMatch	Substitution may map variables to the empty word ε.
NE-VPatMatch	Substitution can only map to non-empty words.

A Very Brief History

Three branches:

- Learning theory and Language theory (1980 - today):
  - Membership problem of Angluin's pattern languages.
  - First NE-case, later E-case.
  - Word equations, where one side is "variable-free".

A Very Brief History

Three branches:

- Learning theory and Language theory (1980 - today):
  - Membership problem of Angluin's pattern languages.
  - First NE-case, later E-case.
  - Word equations, where one side is "variable-free".
- Pattern matching community (1996 - today):
  - Baker's parameterised matching (finding repetitions in program code).
  - A. Amir, Y. Aumann, R. Cole, M. Lewenstein: function matching.
  - A. Amir, I. Nor: generalized function matching.
  - R. Clifford, A. W. Harrow, A. Popa, B. Sach: generalised matching.
  - Only NE-case.

A Very Brief History

Three branches:

- Learning theory and Language theory (1980 - today):
  - Membership problem of Angluin's pattern languages.
  - First NE-case, later E-case.
  - Word equations, where one side is "variable-free".
- Pattern matching community (1996 - today):
  - Baker's parameterised matching (finding repetitions in program code).
  - A. Amir, Y. Aumann, R. Cole, M. Lewenstein: function matching.
  - A. Amir, I. Nor: generalized function matching.
  - R. Clifford, A. W. Harrow, A. Popa, B. Sach: generalised matching.
  - Only NE-case.
- The "real" world (?? - today):
  - Matchtest for regular expressions with backreferences.
  - Nowadays a standard tool in text editors (grep, emacs, . . .) and programming language (Perl, Java, Python, . . .).

NP-Completeness

Theorem (Angluin 1980, Ehrenfeucht and Rozenberg 1979)

If $|\Sigma| \geq 2$, *then* E- *and* NE-$\mathrm{VP}\textsc{at}\mathrm{M}\textsc{atch}$ *are NP-complete.*

NP-Completeness

Theorem (Angluin 1980, Ehrenfeucht and Rozenberg 1979)

If $|\Sigma| \geq 2$*, then* E*- and* $\mathrm{NE\text{-}VP{\scriptsize AT}M{\scriptsize ATCH}}$ *are NP-complete.*

3CNF formula (without negated variables)

$$\psi = (v_1 \vee v_2 \vee v_3) \wedge (v_2 \vee v_4 \vee v_5) \wedge (v_3 \vee v_1 \vee v_3) \wedge (v_4 \vee v_1 \vee v_2)$$

NP-Completeness

Theorem (Angluin 1980, Ehrenfeucht and Rozenberg 1979)

If $|\Sigma| \geq 2$, then E- and NE-VPatMatch are NP-complete.

3CNF formula (without negated variables)

$$\psi = (v_1 \vee v_2 \vee v_3) \wedge (v_2 \vee v_4 \vee v_5) \wedge (v_3 \vee v_1 \vee v_3) \wedge (v_4 \vee v_1 \vee v_2)$$

E-VPatMatch instance:

$$\alpha_\psi = x_1 x_2 x_3 \, \mathtt{b} \, x_2 x_4 x_5 \, \mathtt{b} \, x_3 x_1 x_3 \, \mathtt{b} \, x_4 x_1 x_2$$

$$w_\psi = \mathtt{a\,b\,a\,b\,a\,b\,a}.$$

NP-Completeness

Theorem (Angluin 1980, Ehrenfeucht and Rozenberg 1979)

If $|\Sigma| \geq 2$*, then* E- *and* NE-$\mathrm{VPatMatch}$ *are NP-complete.*

3CNF formula (without negated variables)

$$\psi = (v_1 \vee v_2 \vee v_3) \wedge (v_2 \vee v_4 \vee v_5) \wedge (v_3 \vee v_1 \vee v_3) \wedge (v_4 \vee v_1 \vee v_2)$$

E-VPatMatch instance:

$$\alpha_\psi = x_1\, x_2\, x_3\ \mathtt{b}\ x_2\, x_4\, x_5\ \mathtt{b}\ x_3\, x_1\, x_3\ \mathtt{b}\ x_4\, x_1\, x_2$$
$$w_\psi = \mathtt{a}\,\mathtt{b}\,\mathtt{a}\,\mathtt{b}\,\mathtt{a}\,\mathtt{b}\,\mathtt{a}.$$

$\exists\, h : h(\alpha_\psi) = w_\psi$ iff ψ is “1-in-3-satisfiable”.

Special Cases

OK, so VPatMatch is a hard problem, but what if

Special Cases

OK, so VPatMatch is a hard problem, but what if

- we are only interested in texts of size at most 50,

Special Cases

OK, so VPatMatch is a hard problem, but what if

- we are only interested in texts of size at most 50,
- we are only interested in injective substitutions,

Special Cases

OK, so VPatMatch is a hard problem, but what if

- we are only interested in texts of size at most 50,
- we are only interested in injective substitutions,
- in our patterns every variable occurs at most twice,

Special Cases

OK, so VPatMatch is a hard problem, but what if

- we are only interested in texts of size at most 50,
- we are only interested in injective substitutions,
- in our patterns every variable occurs at most twice,
- we are only interested in patterns without any terminal symbols and we only consider substitutions of the form $h : X \to \{\mathtt{a}, \mathtt{b}, \varepsilon\}$?
  (i. e., for some u over some alphabet Γ and some $w \in \{\mathtt{a}, \mathtt{b}\}^*$, can we obtain w by replacing every $x \in \Gamma$ in u by either a or b or deleting it?)

Some More Notation

For any pattern α (e. g., $\alpha := x_1 \mathtt{a} x_2 x_1 \mathtt{ba} x_2 x_1 x_3$),

Some More Notation

For any pattern α (e. g., $\alpha := x_1 \mathtt{a} x_2 x_1 \mathtt{ba} x_2 x_1 x_3$),

$\mathsf{var}(\alpha)$ is the set of variables in α $\qquad$ $\mathsf{var}(\alpha) = \{x_1, x_2, x_3\}$

Some More Notation

For any pattern α (e. g., $\alpha := x_1 \mathtt{a} x_2 x_1 \mathtt{ba} x_2 x_1 x_3$),

$\mathrm{var}(\alpha)$ is the set of variables in α $\qquad \mathrm{var}(\alpha) = \{x_1, x_2, x_3\}$

$|\alpha|_x$ is the number of Occ. of x in α $\qquad |\alpha|_{x_1} = 3$

Different Versions

Types of VPatMatch:

- Substitutions can be erasing or must be non-erasing.

Different Versions

Types of VPATMATCH:

- Substitutions can be erasing or must be non-erasing.
- Substitutions can be non-injective or must be injective.

Different Versions

Types of VPatMatch:

- Substitutions can be erasing or must be non-erasing.
- Substitutions can be non-injective or must be injective.
- Patterns can contain terminal symbols or must be terminal-free.

Different Versions

Types of VPatMatch:

- Substitutions can be erasing or must be non-erasing.
- Substitutions can be non-injective or must be injective.
- Patterns can contain terminal symbols or must be terminal-free.

Parameters of VPatMatch:

$|\mathsf{var}(\alpha)|$ Number of variables.

Different Versions

Types of VPatMatch:

- Substitutions can be erasing or must be non-erasing.
- Substitutions can be non-injective or must be injective.
- Patterns can contain terminal symbols or must be terminal-free.

Parameters of VPatMatch:

$\lvert \mathsf{var}(\alpha) \rvert$	Number of variables.
$\lvert w \rvert$	word length.

Different Versions

Types of VPATMATCH:

- Substitutions can be erasing or must be non-erasing.
- Substitutions can be non-injective or must be injective.
- Patterns can contain terminal symbols or must be terminal-free.

Parameters of VPATMATCH:

$\lvert \mathsf{var}(\alpha) \rvert$	Number of variables.
$\lvert w \rvert$	word length.
$\lvert h(x) \rvert$	Max. length of substitution words.

Different Versions

Types of VPATMATCH:

- Substitutions can be erasing or must be non-erasing.
- Substitutions can be non-injective or must be injective.
- Patterns can contain terminal symbols or must be terminal-free.

Parameters of VPATMATCH:

$\lvert \mathsf{var}(\alpha) \rvert$	Number of variables.
$\lvert w \rvert$	word length.
$\lvert h(x) \rvert$	Max. length of substitution words.
$\lvert \alpha \rvert_x$	Max. occ. per variable.

Different Versions

Types of VPatMatch:

- Substitutions can be erasing or must be non-erasing.
- Substitutions can be non-injective or must be injective.
- Patterns can contain terminal symbols or must be terminal-free.

Parameters of VPatMatch:

$\lvert \mathsf{var}(\alpha) \rvert$	Number of variables.
$\lvert w \rvert$	word length.
$\lvert h(x) \rvert$	Max. length of substitution words.
$\lvert \alpha \rvert_x$	Max. occ. per variable.
$\lvert \Sigma \rvert$	Alphabet size.

Different Versions

Types of VPATMATCH:

- Substitutions can be erasing or must be non-erasing.
- Substitutions can be non-injective or must be injective.
- Patterns can contain terminal symbols or must be terminal-free.

Parameters of VPATMATCH:

$\lvert \mathsf{var}(\alpha) \rvert$	Number of variables.
$\lvert w \rvert$	word length.
$\lvert h(x) \rvert$	Max. length of substitution words.
$\lvert \alpha \rvert_x$	Max. occ. per variable.
$\lvert \Sigma \rvert$	Alphabet size.

2^3 types, 2^5 combinations of parameters $\rightarrow$ 256 versions of VPATMATCH.

Research Questions

256 Questions of the following form:

Main Research Question

For any type X of $\textsc{VPatMatch}$ and for any subset P of parameters, can we bound the parameters in P by constants, such that type X of $\textsc{VPatMatch}$ is still NP-complete?

First Observations

Theorem (Geilke, Zilles, 2011)

If

$$\begin{aligned} |\mathrm{var}(\alpha)| &\leq c \textit{ or} \\ |w| &\leq c, \end{aligned}$$

for some constant c, then all variants of $\textsc{VPatMatch}$ *are in P.*

First Observations

Theorem (Geilke, Zilles, 2011)

If

$$|\mathrm{var}(\alpha)| \leq c \text{ or}$$
$$|w| \leq c,$$

for some constant c, then all variants of VPatMatch *are in P.*

So we focus on the parameters $|h(x)|$, $|\alpha|_x$ and $|\Sigma|$.

First Observations

Theorem (Geilke, Zilles, 2011)

If

$$\begin{aligned} |\mathsf{var}(\alpha)| &\leq c \textit{ or} \\ |w| &\leq c, \end{aligned}$$

for some constant c, then all variants of $\textsc{VPatMatch}$ *are in P.*

So we focus on the parameters $|h(x)|$, $|\alpha|_x$ and $|\Sigma|$.

Observation

If

$$\begin{aligned} |\alpha|_x &= 1 \text{ or} \\ |\Sigma| &= 1, \end{aligned}$$

then all variants of $\textsc{VPatMatch}$ are in P.

The Non-injective Case

Theorem

Erasing, non-injective VPATMATCH *is NP-complete,*

The Non-injective Case

Theorem

Erasing, non-injective $\textsc{VPatMatch}$ *is NP-complete,*

- *even if*

$$\begin{aligned} |h(x)| &\leq 1\,, \\ |\alpha|_x &\leq 2\,, \\ |\Sigma| &\leq 2\,. \end{aligned}$$

The Non-injective Case

Theorem

Erasing, non-injective $\textsc{VPatMatch}$ *is NP-complete,*

- *even if*

$$\begin{aligned} |h(x)| &\leq 1\,, \\ |\alpha|_x &\leq 2\,, \\ |\Sigma| &\leq 2\,. \end{aligned}$$

- *even if terminal-free and*

$$\begin{aligned} |h(x)| &\leq 1\,, \\ |\alpha|_x &\leq 8, \\ |\Sigma| &\leq 2\,. \end{aligned}$$

The Non-injective Case

Theorem

Erasing, non-injective VPATMATCH *is NP-complete,*

- *even if*

$$\begin{aligned} |h(x)| &\leq 1\,, \\ |\alpha|_x &\leq 2\,, \\ |\Sigma| &\leq 2\,. \end{aligned}$$

- *even if terminal-free and*

$$\begin{aligned} |h(x)| &\leq 1\,, \\ |\alpha|_x &\leq \not{8}\,2, \\ |\Sigma| &\leq 2\,. \end{aligned}$$

The Non-injective Case

Theorem

(Non-)Erasing, non-injective VPATMATCH *is NP-complete,*

- *even if*

$$|h(x)| \leq 1\,,\quad |\alpha|_x \leq 2\,,\quad |\Sigma| \leq 2\,.$$

- *even if terminal-free and*

$$|h(x)| \leq 1\,,\quad |\alpha|_x \leq \not{8}\,2,\quad |\Sigma| \leq 2\,.$$

The Non-injective Case

Theorem

(Non-)Erasing, non-injective VPatMatch *is NP-complete,*

- *even if*

$$
\begin{aligned}
|h(x)| &\le 1\,(3),\\
|\alpha|_x &\le 2\,(2),\\
|\Sigma| &\le 2\,(2).
\end{aligned}
$$

- *even if terminal-free and*

$$
\begin{aligned}
|h(x)| &\le 1\,(3),\\
|\alpha|_x &\le \cancel{8}\,2\,(3),\\
|\Sigma| &\le 2\,(4).
\end{aligned}
$$

The Injective Case 1/2

Theorem

Let $c_1, c_2 \in \mathbb{N}$. All injective variants of $\textsc{VPatMatch}$, *restricted to*

$$\begin{aligned} |h(x)| &\leq c_1, \\ |\Sigma| &\leq c_2, \end{aligned}$$

are in P.

The Injective Case 2/2

For all other injective variants, we have NP-completeness, but the constants are a bit larger.

The Injective Case 2/2

For all other injective variants, we have NP-completeness, but the constants are a bit larger.

Theorem

Injective, erasing or non-erasing, terminal-free or non-terminal-free VPatMatch *is NP-complete,*

- *even if*

$$\begin{aligned} |h(x)| &\leq 19, \\ |\alpha|_x &\leq 4, \end{aligned}$$

- *even if*

$$\begin{aligned} |\alpha|_x &\leq 9, \\ |\Sigma| &\leq 5. \end{aligned}$$

Further Research 1/2

Main Research Question

For any variant X of VPatMatch and for any subset P of parameters, can we bound the parameters in P by constants, such that variant X of VPatMatch is still NP-complete?

Further Research 1/2

Main Research Question

For any variant X of VPatMatch and for any subset P of parameters, can we bound the parameters in P by constants, such that variant X of VPatMatch is still NP-complete?

Dichotomy Result

For any variant X of VPatMatch, for any subset P of parameters and for any set C of specific bounds for the parameters in P, is the variant X of VPatMatch still NP-complete if the parameters of P are bounded by the constants in C?

Further Research 1/2

Main Research Question

For any variant X of $\textsc{VPatMatch}$ and for any subset P of parameters, can we bound the parameters in P by constants, such that variant X of $\textsc{VPatMatch}$ is still NP-complete?

Dichotomy Result for Erasing and Non-injective Case

Let $c_1, c_2, c_3 \in \mathbb{N}$. Erasing, non-injective $\textsc{VPatMatch}$, restricted to

$$\begin{aligned} |h(x)| &\leq c_1, \\ |\alpha|_x &\leq c_2, \\ |\Sigma| &\leq c_3, \end{aligned}$$

is NP-Complete **if and only if** $c_1 \geq 1$, $c_2 \geq 2$, $c_3 \geq 2$.

Further Research 2/2

Parameterized Complexity

Consider the parameters ($|\mathrm{var}|$, $|\Sigma|$, $\ldots$) as parameters in terms of *parameterized complexity theory* and investigate the parameterized complexity of the corresponding parameterized problems.

Thank you very much for your attention.