Pattern Matching with Variables: Fast Algorithms and New Hardness Results

Henning Fernau[1] Florin Manea[2] Robert Mercaş[2,3] **Markus L. Schmid**[1]

[1]Trier University, Germany
[2]Kiel University, Germany
[3]King's College, London, UK

STACS 2015

Patterns with Variables

Finite alphabet of terminals	$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$
Set of variables	$X = \{x_1, x_2, x_3, \ldots\}$
Patterns	$\alpha \in (\Sigma \cup X)^+$
Words	$w \in \Sigma^+$
Substitution	$h : X \to \Sigma^+$ $\alpha = y_1 \ldots y_n$, $h(\alpha) = h(y_1) \ldots h(y_n)$, with $h(a) = a$, $a \in \Sigma$.

Pattern Matching with Variables

pattern α matches word w $\iff$ $\exists$ substitution $h : h(\alpha) = w$.

Pattern Matching with Variables

pattern α matches word w $\iff$ $\exists$ substitution $h : h(\alpha) = w$.

$$\alpha = x_1\, x_2\, x_1\, x_3\, x_2$$

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

Pattern Matching with Variables

pattern α matches word w $\iff$ $\exists$ substitution $h : h(\alpha) = w$.

$$\alpha = \mathtt{a}\,\mathtt{b}\,\mathtt{b}\,x_2\,\mathtt{a}\,\mathtt{b}\,\mathtt{b}\,x_3\,x_2$$

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

Pattern Matching with Variables

pattern α matches word w $\iff$ $\exists$ substitution $h : h(\alpha) = w$.

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

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

Pattern Matching with Variables

pattern α matches word w $\iff$ $\exists$ substitution $h : h(\alpha) = w$.

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

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

Pattern Matching with Variables

pattern α matches word w $\iff$ $\exists$ substitution $h : h(\alpha) = w$.

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

Pattern Matching with Variables

pattern α matches word w $\iff$ $\exists$ substitution $h : h(\alpha) = w$.

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

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

Pattern Matching with Variables

pattern α matches word w $\Longleftrightarrow$ $\exists$ substitution $h : h(\alpha) = w$.

$\alpha = \texttt{b a c b a c b c b a c b c}$

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

Motivation

- Learning theory (inductive inference, PAC learning),
- language theory (pattern languages),
- combinatorics on words (word equations, unavoidable patterns, ambiguity of morphisms, equality sets),
- pattern matching (parameterised matching, (generalised) function matching),
- matchtest for regular expressions with backreferences (text editors (grep, emacs), programming language (Perl, Java, Python)),
- database theory.

Complexity

Matching Problem (Match)

Given a pattern α, a word w. Does α match w (i. e., $\exists h : h(\alpha) = w$)?

- Match is (in general) NP-complete.

Complexity

Matching Problem (Match)

Given a pattern α, a word w. Does α match w (i. e., $\exists h : h(\alpha) = w$)?

- Match is (in general) NP-complete.

- Bad news: Match remains hard if numerical parameters are restricted (few exceptions):
  - Match $\in P$ if number of variables or word length bounded (trivial).
  - Match still hard if
    - alphabet size 2,
    - each variable has at most 2 occurrences,
    - $|h(x)| \leq 3$ for every x.

Complexity

Matching Problem (Match)

Given a pattern α, a word w. Does α match w (i. e., $\exists h : h(\alpha) = w$)?

- Match is (in general) NP-complete.

- Bad news: Match remains hard if numerical parameters are restricted (few exceptions):
  - Match $\in P$ if number of variables or word length bounded (trivial).
  - Match still hard if
    - alphabet size 2,
    - each variable has at most 2 occurrences,
    - $|h(x)| \leq 3$ for every x.

- Good news: Tractable if structure of patterns is restricted.

Notation

$\mathrm{var}(\alpha)$	Set of variables occurring in pattern α.
$\|\alpha\|_x$	Number of occurrences of variable x in pattern α.

Structural Restrictions of Patterns

- **Regular Patterns**:
  $|\alpha|_x = 1$, $x \in \mathrm{var}(\alpha)$.
  E. g., $\alpha = \mathtt{ab}x_1x_2\mathtt{b}x_3\mathtt{aaa}x_4\mathtt{b}$.

Structural Restrictions of Patterns

- **Regular Patterns**:
  $|\alpha|_x = 1$, $x \in \mathrm{var}(\alpha)$.
  E. g., $\alpha = \mathtt{ab}x_1x_2\mathtt{b}x_3\mathtt{aaa}x_4\mathtt{b}$.
- **Non-Cross Patterns**:
  $\alpha = \ldots x \ldots y \ldots x \ldots$ is not possible.
  E. g., $\alpha = x_1\mathtt{aba}x_1\mathtt{a}x_1x_2x_2\mathtt{ba}x_2x_3x_3\mathtt{bb}x_3\mathtt{a}x_3$

Structural Restrictions of Patterns

- ***k*-Repeated-Variable Patterns**:
  $|\{x \in \mathrm{var}(\alpha) \mid |\alpha|_x \geq 2\}| \leq k$.
  E. g., $\alpha = x_1 \mathtt{ab} x_2 \mathtt{a} x_2 \mathtt{a} x_3 \mathtt{ba} x_2 \mathtt{bb} x_4 x_2 x_5$ is a 1-repeated-variable pattern.

Structural Restrictions of Patterns

- ***k*-Repeated-Variable Patterns**:
  $|\{x \in \text{var}(\alpha) \mid |\alpha|_x \geq 2\}| \leq k$.
  E. g., $\alpha = x_1\mathtt{ab}x_2\mathtt{a}x_2\mathtt{a}x_3\mathtt{ba}x_2\mathtt{bb}x_4x_2x_5$ is a 1-repeated-variable pattern.

- **Pattern with Bounded Scope Coincidence Degree**:
  Scope (of x): shortest factor containing all occ. of x,
  Scope coincidence degree: maximum number of coinciding scopes.

$$\alpha_1 = x_1 \; x_2 \; x_1 \; x_3 \; x_2 \; x_3 \; x_1 \; x_2 \; x_3 \quad \text{scd}(\alpha_1) = 3$$

$$\alpha_2 = x_1 \; x_2 \; x_1 \; x_1 \; x_2 \; x_3 \; x_2 \; x_3 \; x_3 \quad \text{scd}(\alpha_2) = 2$$

Structural Restrictions of Patterns - Complexity

Known results: MATCH is in P for

- regular patterns $\mathcal{O}(|\alpha| + |w|)$,
- non-cross patterns $\mathcal{O}(|\alpha||w|^4)$,
- patterns with scd $\leq k$ $\mathcal{O}(|\alpha||w|^{2(k+3)}(k+2)^2)$.

Structural Restrictions of Patterns - Complexity

Known results: MATCH is in P for

- regular patterns $\mathcal{O}(|\alpha| + |w|)$,
- non-cross patterns $\mathcal{O}(|\alpha||w|^4)$,
- patterns with scd $\leq k$ $\mathcal{O}(|\alpha||w|^{2(k+3)}(k+2)^2)$.

Our contribution:

- Find (efficient) algorithms for these cases.
- Can we extend our algorithms to the injective case (i. e., different variables are replaced by different words)?

k-Repeated Variable Patterns

Lemma

MATCH *for* 1*-repeated-variable patterns is solvable in* $\mathcal{O}(|w|^2)$.

Theorem

MATCH *for* k*-repeated-variable patterns is solvable in* $\mathcal{O}\left(\frac{|w|^{2k}}{((k-1)!)^2}\right)$.

Non-Cross Patterns

Dynamic programming approach!

α non-cross $\Rightarrow$

$\alpha = w_0\alpha_1 w_1\alpha_2 \ldots \alpha_\ell w_\ell$. $\quad \mathrm{var}(\alpha_i) = \{x_i\}$, $w_i \in \Sigma^*$

Non-Cross Patterns

Dynamic programming approach!

α non-cross $\Rightarrow$

$\alpha = w_0\alpha_1 w_1 \alpha_2 \ldots \alpha_\ell w_\ell$. $\qquad$ $\mathrm{var}(\alpha_i) = \{x_i\}$, $w_i \in \Sigma^*$

Compute all sub-problems:

Does $w_0\alpha_1 w_1 \ldots w_{i-1}\alpha_i$ match $w[1..j]$? $\qquad$ $1 \leq i \leq \ell$, $1 \leq j \leq |w|$

Non-Cross Patterns

Case 1: $\alpha_i = x_i$

$$w_0\alpha_1 w_1 \ldots w_{i-1} \;\; \alpha_i$$

$$\downarrow$$

$$w[1..j]$$

Non-Cross Patterns

Case 1: $\alpha_i = x_i$

$$w_0\alpha_1 w_1 \ldots w_{i-1}\ x_i$$

$$\downarrow$$

$$w[1..j]$$

Non-Cross Patterns

Case 1: $\alpha_i = x_i$

$$w_0\alpha_1 w_1 \ldots w_{i-1}\ \ x_i$$

$$\downarrow$$

$$w[1..j]$$

$$\Longleftrightarrow$$

$$w_0\alpha_1 w_1 \ldots w_{i-1}$$

$$\downarrow$$

$$w[1..j']$$

Non-Cross Patterns

Case 1: $\alpha_i = x_i$

$$w_0\alpha_1 w_1 \ldots w_{i-1}\ x_i$$

$$\downarrow$$

$$w[1..j]$$

$$\Longleftrightarrow$$

$$w_0\alpha_1 w_1 \ldots w_{i-1} \qquad\qquad x_i$$

$$\downarrow \qquad\qquad \downarrow$$

$$w[1..j'] \qquad\qquad w[j'+1..j]$$

Non-Cross Patterns

Case $2a$: $\alpha_i = (x_i)^k$ (x_i is mapped to primitive word t)

$$w_0 \alpha_1 w_1 \ldots w_{i-1} \;\; \alpha_i$$

$$\downarrow$$

$$w[1..j]$$

Non-Cross Patterns

Case $2a$: $\alpha_i = (x_i)^k$ (x_i is mapped to primitive word t)

$$w_0\alpha_1 w_1 \ldots w_{i-1}\ x_i x_i \ldots x_i$$

$$\downarrow$$

$$w[1..j]$$

Non-Cross Patterns

Case $2a$: $\alpha_i = (x_i)^k$ (x_i is mapped to primitive word t)

$$w_0\alpha_1 w_1 \dots w_{i-1}\ \ x_i x_i \dots x_i$$
$$\downarrow$$
$$w[1..j]$$

$$\Longleftrightarrow$$

$\exists$ primitive word t with t^k suffix of $w[1..j]$ and

$$w_0\alpha_1 w_1 \dots w_{i-1}$$
$$\downarrow$$
$$w[1..j-(k|t|)]$$

Non-Cross Patterns

Case $2a$: $\alpha_i = (x_i)^k$ (x_i is mapped to primitive word t)

$$w_0\alpha_1 w_1 \dots w_{i-1} \; x_i x_i \dots x_i$$
$$\downarrow$$
$$w[1..j]$$

$$\Longleftrightarrow$$

$\exists$ primitive word t with t^k suffix of $w[1..j]$ and

$$w_0\alpha_1 w_1 \dots w_{i-1} \qquad x_i x_i \dots x_i$$
$$\downarrow \qquad \downarrow$$
$$w[1..j-(k|t|)] \qquad tt\dots t$$

Non-Cross Patterns

Case $2a$: Find all primitive t such that $w[1..j]$ has t^2 as a suffix!

Lemma (Crochemore, 1981)

Primitive u_1, u_2, u_3, $|u_1| < |u_2| < |u_3|$, $w = w_i u_i u_i$, $1 \leq i \leq 3 \Rightarrow 2|u_1| < |u_3|$.

$\Rightarrow$ w has at most $2 \log |w|$ primitively rooted squares as suffix.

Non-Cross Patterns

Case $2a$: Find all primitive t such that $w[1..j]$ has t^2 as a suffix!

Lemma (Crochemore, 1981)

Primitive u_1, u_2, u_3, $|u_1| < |u_2| < |u_3|$, $w = w_i u_i u_i$, $1 \leq i \leq 3 \Rightarrow 2|u_1| < |u_3|$.

$\Rightarrow$ w has at most $2 \log |w|$ primitively rooted squares as suffix.

Lemma

We can compute in $\mathcal{O}(n \log n)$ *time all the sets* $P_i = \{u \mid u \text{ primitive}, u^2 \text{ suffix of } w[1..i]\}$, $1 \leq i \leq |w|$.

$\Rightarrow$ **Case** $2a$ can be done efficiently.

Non-Cross Patterns

Case 2*b*: $\alpha_i = (x_i)^k$ (x_i is mapped to some word $t = v^{h+1}$)

$$w_0 \alpha_1 w_1 \ldots w_{i-1} \; x_i x_i \ldots x_i$$

$$\downarrow$$

$$w[1..j]$$

Non-Cross Patterns

Case 2b: $\alpha_i = (x_i)^k$ (x_i is mapped to some word $t = v^{h+1}$)

$$w_0\alpha_1 w_1 \ldots w_{i-1}\; x_i x_i \ldots x_i$$
$$\downarrow$$
$$w[1..j]$$

$$\Longleftrightarrow$$

$\exists$ primitive word v with v^k suffix of $w[1..j]$ and

$$w_0\alpha_1 w_1 \ldots w_{i-1} x_i x_i \ldots x_i \qquad \text{with } h(x_i) = v^h$$
$$\downarrow$$
$$w[1..j - k|v|)]$$

Non-Cross Patterns

Case 3: $\alpha_i = x_i^{\ell_0} u_1 x_i^{\ell_1} u_2 \ldots x_i^{\ell_{p-1}} u_p x_i^{\ell_p}$ $\qquad u_k \in \Sigma^+$

$$w_0 \alpha_1 w_1 \ldots w_{i-1} \;\; \alpha_i$$

$$\downarrow$$

$$w[1..j]$$

Non-Cross Patterns

Case 3: $\alpha_i = x_i^{\ell_0} u_1 x_i^{\ell_1} u_2 \ldots x_i^{\ell_{p-1}} u_p x_i^{\ell_p}$ $\qquad u_k \in \Sigma^+$

$$w_0 \alpha_1 w_1 \ldots w_{i-1} \; x_i^{\ell_0} u_1 x_i^{\ell_1} u_2 \ldots x_i^{\ell_{p-1}} u_p x_i^{\ell_p}$$

$$\downarrow$$

$$w[1..j]$$

Non-Cross Patterns

Case 3: $\alpha_i = x_i^{\ell_0} u_1 x_i^{\ell_1} u_2 \ldots x_i^{\ell_{p-1}} u_p x_i^{\ell_p}$ $\qquad u_k \in \Sigma^+$

$$w_0 \alpha_1 w_1 \ldots w_{i-1} \; x_i^{\ell_0} u_1 x_i^{\ell_1} u_2 \ldots x_i^{\ell_{p-1}} u_p x_i^{\ell_p}$$

$$\downarrow$$

$$w[1..j]$$

- $\ell_p \geq 2$: proceed similar to **Case** 2 (more involved, details omitted).
- $\ell_p = 1$: find all primitive $u_p t$ such that $t u_p t$ is a suffix of $w[1..j]$.

Non-Cross Patterns

Generalisation of Crochemore's result:

Lemma

For a fixed v, w has $\mathcal{O}(\log |w|)$ factors uvu with uv primitive as suffixes.

Lemma

For fixed v, w, we can compute in $\mathcal{O}(n \log n)$ time all the sets $R_i^v = \{u \mid uv \text{ primitive}, uvu \text{ suffix of } w[1..i]\}$, $1 \leq i \leq |w|$.

$\Rightarrow$ **Case** 3 can be done efficiently.

Non-Cross Patterns

Theorem

MATCH *for non-cross patterns is solvable in* $\mathcal{O}(|w|m \log |w|)$, *where* m *is the number of one-variable blocks of the pattern.*

Theorem

MATCH *for patterns with scope coincidence degree of at most* k *is solvable in* $\mathcal{O}\left(\frac{|w|^{2k}m}{((k-1)!)^2}\right)$, *where* m *is the number of one-variable blocks of the pattern.*

Injective Match

InjMatch: Like Match, but we are looking for an injective substitution h, i. e., $x \neq y \Rightarrow h(x) \neq h(y)$.

Can we use our (or other) Match-algorithms also for InjMatch?

InjMatch remains NP-complete for patterns for which Match is (trivially) in P.

Injective MATCH

Theorem

INJMATCH *is NP-complete even for patterns* $x_1 x_2 \ldots x_n$, $n \geq 1$.

We prove NP-completeness of the equivalent problem

UNFACT

Instance: A word w and an integer $k \geq 1$.
Question: $w = u_1 u_2 \ldots u_{k'}$ with $k' \geq k$ and $u_i \neq u_j$, $1 \leq i < j \leq k$?

Corollary

INJMATCH *is NP-complete for regular, non-cross, k-repeated-variable, bounded scd patterns.*

Hardness of InjMatch - Proof Idea

3D-Match

Instance: An integer $\ell \in \mathbb{N}$ and a set
$S \subseteq \{(p, q, r) \mid 1 \leq p < \ell + 1 \leq q < 2\ell + 1 \leq r \leq 3\ell\}$.
Question: Does there exist a subset S' of S with cardinality ℓ such that, for each two elements $(p, q, r), (p', q', r') \in S'$, $p \neq p'$, $q \neq q'$ and $r \neq r'$?

Hardness of InjMatch - Proof Idea

3D-Match instance (S, ℓ): $S = \{s_1, s_2, \ldots, s_k\}$
Transform every $s_i = (p_i, q_i, r_i)$, $1 \leq i \leq k$, into

$$v_i = \quad \star_i \quad p_i \quad \mathtt{a} \quad \mathtt{b}_{i,1} \quad \mathtt{b}_{i,2} \quad q_i \quad \mathtt{a} \quad \mathtt{b}_{i,3} \quad \mathtt{b}_{i,4} \quad r_i \quad \mathtt{a} \quad \diamond_i$$

$\star_i$, $\diamond_i$, $\mathtt{b}_{i,j}$ have only one occurrence!

Hardness of InjMatch - Proof Idea

3D-Match instance (S, ℓ): $S = \{s_1, s_2, \ldots, s_k\}$
Transform every $s_i = (p_i, q_i, r_i)$, $1 \leq i \leq k$, into

$$v_i = \quad \star_i \quad p_i \quad \mathtt{a} \quad \mathtt{b}_{i,1} \quad \mathtt{b}_{i,2} \quad q_i \quad \mathtt{a} \quad \mathtt{b}_{i,3} \quad \mathtt{b}_{i,4} \quad r_i \quad \mathtt{a} \quad \diamond_i$$

$\star_i$, $\diamond_i$, $\mathtt{b}_{i,j}$ have only one occurrence!

Let $S' \subseteq S$.

$$(p_i, q_i, r_i) \notin S' \quad \Leftrightarrow \star_i p_i \quad \mathtt{a}\mathtt{b}_{i,1} \quad \mathtt{b}_{i,2} q_i \quad \mathtt{a}\mathtt{b}_{i,3} \quad \mathtt{b}_{i,4} r_i \quad \mathtt{a}\diamond_i$$

$$(p_i, q_i, r_i) \in S' \quad \Leftrightarrow \quad \star_i \quad p_i\mathtt{a} \quad \mathtt{b}_{i,1}\mathtt{b}_{i,2} \quad q_i\mathtt{a} \quad \mathtt{b}_{i,3}\mathtt{b}_{i,4} \quad r_i\mathtt{a} \quad \diamond_i$$

Hardness of InjMatch - Proof Idea

3D-Match instance (S, ℓ): $S = \{s_1, s_2, \ldots, s_k\}$
Transform every $s_i = (p_i, q_i, r_i)$, $1 \leq i \leq k$, into

$$v_i = \star_i \quad p_i \quad \mathtt{a} \quad \mathtt{b}_{i,1} \quad \mathtt{b}_{i,2} \quad q_i \quad \mathtt{a} \quad \mathtt{b}_{i,3} \quad \mathtt{b}_{i,4} \quad r_i \quad \mathtt{a} \quad \diamond_i$$

$\star_i$, $\diamond_i$, $\mathtt{b}_{i,j}$ have only one occurrence!

Let $S' \subseteq S$.

$$(p_i, q_i, r_i) \notin S' \quad \Leftrightarrow \star_i p_i \quad \mathtt{a}\mathtt{b}_{i,1} \quad \mathtt{b}_{i,2} q_i \quad \mathtt{a}\mathtt{b}_{i,3} \quad \mathtt{b}_{i,4} r_i \quad \mathtt{a}\diamond_i$$

$$(p_i, q_i, r_i) \in S' \quad \Leftrightarrow \quad \star_i \quad p_i\mathtt{a} \quad \mathtt{b}_{i,1}\mathtt{b}_{i,2} \quad q_i\mathtt{a} \quad \mathtt{b}_{i,3}\mathtt{b}_{i,4} \quad r_i\mathtt{a} \quad \diamond_i$$

$v = u_1 u_2 \ldots u_n$ with $n = 7\ell + 6(k - \ell)$ and $u_i \neq u_j$, $1 \leq i < j \leq n$
$\Longleftrightarrow$
S' is a solution of (S, ℓ).

Alphabet Size

Our Reduction needs an unbounded alphabet!

Hardness of InjMatch for fixed alphabets is open, but...

Theorem

InjMatch *(with constant alphabet) is NP-complete for ~~regular~~, non-cross, ~~k-repeated-variable~~, bounded scd patterns.*

Thank you very much for your attention.