Local Patterns

k-Locality and the Matching Problem

Joel Day, Pamela Fleischmann, Florin Manea, and Dirk Nowotka

{jda,fpa,flm,dn}@informatik.uni-kiel.de

February 20, 2018

Kiel University

Dependable Systems Group

Motivation

The *Marking-Words* Game

CONFERENCE

- 2 Players alternately mark **all** occurrences of a letter

The *Marking-Words* Game

CONFERENCE

- 2 Players alternately mark **all** occurrences of a letter
- each new block $\Rightarrow$ -1 point

The *Marking-Words* Game

CONFERENCE

- 2 Players alternately mark **all** occurrences of a letter
- each new block ⇒ -1 point
- combining two blocks ⇒ +1 point

The *Marking-Words* Game

- 2 Players alternately mark **all** occurrences of a letter
- each new block ⇒ -1 point
- combining two blocks ⇒ +1 point

The *Marking-Words* Game

- 2 Players alternately mark **all** occurrences of a letter
- each new block ⇒ -1 point
- combining two blocks ⇒ +1 point

The *Marking-Words* Game

- 2 Players alternately mark **all** occurrences of a letter
- each new block ⇒ -1 point
- combining two blocks ⇒ +1 point

The *Marking-Words* Game

- 2 Players alternately mark **all** occurrences of a letter
- each new block ⇒ -1 point
- combining two blocks ⇒ +1 point

The *Marking-Words* Game

- 2 Players alternately mark **all** occurrences of a letter
- each new block ⇒ -1 point
- combining two blocks ⇒ +1 point

The *Marking-Words* Game

- 2 Players alternately mark **all** occurrences of a letter
- each new block ⇒ -1 point
- combining two blocks ⇒ +1 point

The *Marking-Words* Game

- 2 Players alternately mark **all** occurrences of a letter
- each new block ⇒ -1 point
- combining two blocks ⇒ +1 point

What is the minimal number of blocks needed?

k-Locality

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word

Example

aabccdaabbeee

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word

Example

$$\underbrace{\text{aa}}\ \underbrace{\text{b}}\ \underbrace{\text{cc}}\ \underbrace{\text{d}}\ \underbrace{\text{aa}}\ \underbrace{\text{bb}}\ \underbrace{\text{eee}}$$

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word

Example

$$\mathtt{aabccdaabbeee} \longrightarrow [\mathtt{a}]^{\mathfrak{b}}[\mathtt{b}]^{\mathfrak{b}}[\mathtt{c}]^{\mathfrak{b}}[\mathtt{d}]^{\mathfrak{b}}[\mathtt{a}]^{\mathfrak{b}}[\mathtt{b}]^{\mathfrak{b}}[\mathtt{e}]^{\mathfrak{b}}$$

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word
- **marking**: mapping from letter a to copy $\overline{a}$

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word
- **marking**: mapping from letter a to copy $\overline{a}$
- **marking sequence**: enumeration of letters

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word
- **marking**: mapping from letter a to copy $\overline{a}$
- **marking sequence**: enumeration of letters

Example

$$\mathsf{banana} + (\mathsf{a}, \mathsf{b}, \mathsf{n}) \longrightarrow \mathsf{b}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \text{ (3 blocks)}$$

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word
- **marking**: mapping from letter a to copy $\overline{a}$
- **marking sequence**: enumeration of letters

Example

$$\mathsf{banana} + (\mathsf{a}, \mathsf{b}, \mathsf{n}) \longrightarrow \mathsf{b}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \text{ (3 blocks)}$$

$$\longrightarrow \overline{\mathsf{ba}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \text{ (3 blocks)}$$

Blocks and k-Locality

Definition (Block, Marking, Marking Sequence)

- **block**: continuous part in the word
- **marking**: mapping from letter a to copy $\overline{a}$
- **marking sequence**: enumeration of letters

Example

$$\begin{aligned}\mathsf{banana} + (\mathsf{a}, \mathsf{b}, \mathsf{n}) &\longrightarrow \mathsf{b}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \text{ (3 blocks)}\\ &\longrightarrow \overline{\mathsf{ba}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \text{ (3 blocks)}\\ &\longrightarrow \overline{\mathsf{banana}} \text{ (1 block)}\end{aligned}$$

k-locality

Definition (k-local)

- word ***k*-local**: there exists marking sequence, such that at any time no more than k blocks are marked

k-locality

Definition (k-local)

- word **k-local**: there exists marking sequence, such that at any time no more than k blocks are marked

Example

$\mathsf{banana} + (\mathsf{a}, \mathsf{b}/\mathsf{n}, \mathsf{n}/\mathsf{b}) \longrightarrow \mathsf{b\overline{a}n\overline{a}n\overline{a}} \longrightarrow \mathsf{\overline{ba}n\overline{a}n\overline{a}} \longrightarrow \mathsf{\overline{banana}}$ 3-local

Definition (k-local)

- word k**-local**: there exists marking sequence, such that at any time no more than k blocks are marked

Example

$\mathsf{banana} + (\mathsf{a}, \mathsf{b}/\mathsf{n}, \mathsf{n}/\mathsf{b}) \longrightarrow \mathsf{b}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \longrightarrow \overline{\mathsf{ba}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \longrightarrow \overline{\mathsf{banana}}$ 3-local

$\mathsf{banana} + (\mathsf{b}, \mathsf{a}/\mathsf{n}, \mathsf{n}/\mathsf{a}) \longrightarrow \overline{\mathsf{b}}\mathsf{anana} \longrightarrow \overline{\mathsf{ba}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \longrightarrow \overline{\mathsf{banana}}$ 3-local

k-locality

Definition (*k*-local)

- word ***k*-local**: there exists marking sequence, such that at any time no more than k blocks are marked

Example

$$\mathsf{banana} + (\mathsf{a}, \mathsf{b}/\mathsf{n}, \mathsf{n}/\mathsf{b}) \longrightarrow \mathsf{b}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \longrightarrow \overline{\mathsf{ba}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \longrightarrow \overline{\mathsf{banana}} \quad \text{3-local}$$

$$\mathsf{banana} + (\mathsf{b}, \mathsf{a}/\mathsf{n}, \mathsf{n}/\mathsf{a}) \longrightarrow \overline{\mathsf{b}}\mathsf{anana} \longrightarrow \overline{\mathsf{ba}}\mathsf{n}\overline{\mathsf{a}}\mathsf{n}\overline{\mathsf{a}} \longrightarrow \overline{\mathsf{banana}} \quad \text{3-local}$$

$$\mathsf{banana} + (\mathsf{n}, \mathsf{b}, \mathsf{a}) \longrightarrow \mathsf{ba}\overline{\mathsf{n}}\mathsf{a}\overline{\mathsf{n}}\mathsf{a} \longrightarrow \overline{\mathsf{b}}\mathsf{a}\overline{\mathsf{n}}\mathsf{a}\overline{\mathsf{n}}\mathsf{a} \longrightarrow \overline{\mathsf{banana}} \quad \text{3-local}$$

k-locality

Definition (k-local)

- word k**-local**: there exists marking sequence, such that at any time no more than k blocks are marked

Example

$$\mathtt{banana} + (\mathtt{a}, \mathtt{b}/\mathtt{n}, \mathtt{n}/\mathtt{b}) \longrightarrow \mathtt{b}\overline{\mathtt{a}}\mathtt{n}\overline{\mathtt{a}}\mathtt{n}\overline{\mathtt{a}} \longrightarrow \overline{\mathtt{ba}}\mathtt{n}\overline{\mathtt{a}}\mathtt{n}\overline{\mathtt{a}} \longrightarrow \overline{\mathtt{banana}} \quad \text{3-local}$$

$$\mathtt{banana} + (\mathtt{b}, \mathtt{a}/\mathtt{n}, \mathtt{n}/\mathtt{a}) \longrightarrow \overline{\mathtt{b}}\mathtt{anana} \longrightarrow \overline{\mathtt{ba}}\mathtt{n}\overline{\mathtt{a}}\mathtt{n}\overline{\mathtt{a}} \longrightarrow \overline{\mathtt{banana}} \quad \text{3-local}$$

$$\mathtt{banana} + (\mathtt{n}, \mathtt{b}, \mathtt{a}) \longrightarrow \mathtt{ba}\overline{\mathtt{n}}\mathtt{a}\overline{\mathtt{n}}\mathtt{a} \longrightarrow \overline{\mathtt{b}}\mathtt{a}\overline{\mathtt{n}}\mathtt{a}\overline{\mathtt{n}}\mathtt{a} \longrightarrow \overline{\mathtt{banana}} \quad \text{3-local}$$

$$\mathtt{banana} + (\mathtt{n}, \mathtt{a}, \mathtt{b}) \longrightarrow \mathtt{ba}\overline{\mathtt{n}}\mathtt{a}\overline{\mathtt{n}}\mathtt{a} \longrightarrow \mathtt{b}\overline{\mathtt{anana}} \longrightarrow \overline{\mathtt{banana}} \quad \text{2-local}$$

k-locality

Definition (k-local)

- word k-**local**: there exists marking sequence, such that at any time no more than k blocks are marked

Problem (Decision Problem)

Given a word w and $k \in \mathbb{N}$, decide whether w is k-local.

k-locality

Definition (k-local)

- word ***k*-local**: there exists marking sequence, such that at any time no more than k blocks are marked

Problem (Decision Problem)

Given a word w and $k \in \mathbb{N}$, decide whether w is k-local.

Problem (Minimisation Problem)

Find the minimal $k \in \mathbb{N}$ for a given $w \in \Sigma^*$, such that w is k-local.

More Examples

- **minimisation** is 3-local

More Examples

- **minimisation** is 3-local
- **floccinaucinihilipilification** is 6-local

More Examples

- **minimisation** is 3-local
- **floccinaucinihilipilification** is 6-local
- **pneumonoultramicroscopicsilicovolcanoconiosis** is 8-local

More Examples

- **minimisation** is 3-local
- **floccinaucinihilipilification** is 6-local
- **pneumonoultramicroscopicsilicovolcanoconiosis** is 8-local
- **Lentokonesuihkuturbiinimoottoriapumekaanikko-aliupseerioppilas** is 10-local

More Examples

- **minimisation** is 3-local
- **floccinaucinihilipilification** is 6-local
- **pneumonoultramicroscopicsilicovolcanoconiosis** is 8-local
- **Lentokonesuihkuturbiinimoottoriapumekaanikko-aliupseerioppilas** is 10-local

Does it have to be long?

Basic Observation

- worst case = highest minimal k: $(ab)^n$ is strictly $\frac{|w|}{2}$-local

Basic Observation

- worst case = highest minimal k: $(ab)^n$ is strictly $\frac{|w|}{2}$-local
- it doesn't have to be artifical or long

Onko teidän jo **tutustuttu** reittin?

Basic Observation

- worst case = highest minimal k: $(ab)^n$ is strictly $\frac{|w|}{2}$-local
- it doesn't have to be artifical

$$\text{Onko teidät jo } \underbrace{\textbf{tutustuttu}}_{\text{4-local}} \text{ reittiin?}$$

Basic Observation

- worst case = highest minimal k: $(ab)^n$ is strictly $\frac{|w|}{2}$-local
- it doesn't have to be artifical

$$\text{Onko teidät jo } \underbrace{\textbf{tutustuttu}}_{\text{4-local}} \text{ reittiin?}$$

- k-local $\Rightarrow$ $(k+i)$-local for $i \in \mathbb{N}$

Basic Observation

- worst case = highest minimal k: $(ab)^n$ is strictly $\frac{|w|}{2}$-local
- it doesn't have to be artifical

$$\text{Onko teidät jo } \underbrace{\textbf{tutustuttu}}_{\text{4-local}} \text{ reittiin?}$$

- k-local $\Rightarrow$ $(k+i)$-local for $i \in \mathbb{N}$
- factor of k-local word is k-local

Complexity

Theorem

The decision problem, whether a word is k-local is NPC in general.

Theorem

The decision problem, whether a word is k-local is NPC in general.

- Reduction from Clique

Complexity

Theorem

The decision problem, whether a word is k-local is NPC in general.

Theorem

There exists an algorithm deciding k-locality in time $\mathcal{O}(k|w|^{2k})$

Complexity

Theorem

The decision problem, whether a word is k-local is NPC in general.

Theorem

There exists an algorithm deciding k-locality in time $\mathcal{O}(k|w|^{2k})$

- dynamic programming, memoize marked factors

Complexity

Theorem

The decision problem, whether a word is k-local is NPC in general.

Theorem

There exists an algorithm deciding k-locality in time $\mathcal{O}(k|w|^{2k})$

- dynamic programming, memoize marked factors
- algorithm is constructive ($\Rightarrow$ marking sequence)

1-locality

Intuition for 1-local words:

1-locality

Intuition for 1-local words:

aaa

1-locality

Intuition for 1-local words:

aaa ~~aaaa~~

1-locality

Intuition for 1-local words:

aaa

1-locality

Intuition for 1-local words:

bbaaab

1-locality

Intuition for 1-local words:

bbaaabcccc

1-locality

Intuition for 1-local words:

dddbbaaabccccdd

1-locality

Intuition for 1-local words:

$$\ldots \text{dddbbaaabccccdd} \ldots$$

1-locality

Intuition for 1-local words:

$$\ldots \text{dddbbaaabccccdd} \ldots$$

Let's rewrite this:

$$\ldots [d]^\flat [b]^\flat [a]^\flat [b]^\flat [c]^\flat [d]^\flat \ldots$$

1-locality

Intuition for 1-local words:

$$\ldots \text{dddbbaaabccccdd} \ldots$$

Let's rewrite this:

$$\ldots [d]^\flat \underbrace{[c]^\flat}_{\text{empty}} [b]^\flat [a]^\flat [b]^\flat [c]^\flat [d]^\flat \ldots$$

1-locality

Intuition for 1-local words:

$$\ldots \text{dddbbaaabccccdd} \ldots$$

Let's rewrite this:

$$\ldots [d]^\flat \underbrace{[c]^\flat}_{\text{empty}} [b]^\flat [a]^\flat [b]^\flat [c]^\flat [d]^\flat \ldots$$

$\Rightarrow$ 1-local have palindromic structure

Decision Problem for $k = 1$

Theorem

Given a word $w \in \Sigma^m$ we can decide in time $\mathcal{O}(m)$ whether w is 1-local.

Decision Problem for $k = 1$

Theorem

Given a word $w \in \Sigma^m$ we can decide in time $\mathcal{O}(m)$ whether w is 1-local.

- algorithm is again constructive

2-locality

Intuition for 2-local words:

2-locality

Intuition for 2-local words:

$$[a]^{\mathrm{b}} \ldots [a]^{\mathrm{b}}$$

2-locality

Intuition for 2-local words:

$$[b]^{\mathfrak{b}}[a]^{\mathfrak{b}}[b]^{\mathfrak{b}}\dots[b]^{\mathfrak{b}}[a]^{\mathfrak{b}}[b]^{\mathfrak{b}}$$

2-locality

Intuition for 2-local words:

$$\color{red}{[c]^\flat}[b]^\flat[a]^\flat[b]^\flat\color{red}{[c]^\flat}[b]^\flat[a]^\flat[b]^\flat\color{red}{[c]^\flat}\dots\color{red}{[c]^\flat}$$

2-locality

Intuition for 2-local words:

$$\color{red}{[c]^{\mathfrak{b}}}[b]^{\mathfrak{b}}[a]^{\mathfrak{b}}[b]^{\mathfrak{b}}\color{red}{[c]^{\mathfrak{b}}}[b]^{\mathfrak{b}}[a]^{\mathfrak{b}}[b]^{\mathfrak{b}}\color{red}{[c]^{\mathfrak{b}}}\dots\color{red}{[c]^{\mathfrak{b}}}$$

$\Rightarrow$ structure is more complicated

Pattern Matching

Basic Definitions

Definition

- $\Sigma = \{a, b, \ldots\}$ finite terminal alphabet

Basic Definitions

Definition

- $\Sigma = \{a, b, \ldots\}$ finite terminal alphabet
- $X = \{x_1, x_2, \ldots\}$ infinite set of variables

Basic Definitions

Definition

- $\Sigma = \{a, b, \dots\}$ finite terminal alphabet
- $X = \{x_1, x_2, \dots\}$ infinite set of variables
- **pattern:** word over $(X \cup \Sigma)$

Basic Definitions

Definition

- $\Sigma = \{a, b, \dots\}$ finite terminal alphabet
- $X = \{x_1, x_2, \dots\}$ infinite set of variables
- **pattern:** word over $(X \cup \Sigma)$

Example

$$x_4\, loc\, x_2\, nau\, x_2\, n\, x_3\, h\, x_1\, p\, x_1\, x_4\, x_3\, cat\, x_3\, on$$

Cont.

Definition

- **substitution:** $h : (X \cup \Sigma)^* \longrightarrow \Sigma^*$ with $h(a) = a$ for all $a \in \Sigma$

Example

$$x_4\, loc\, x_2\, nau\, x_2\, n\, x_3\, h\, x_1\, p\, x_1\, x_4\, x_3\, cat\, x_3\, on$$

Cont.

Definition

- **substitution:** $h : (X \cup \Sigma)^* \longrightarrow \Sigma^*$ with $h(a) = a$ for all $a \in \Sigma$

Example

$h(x_1) = ili$

$$x_4\, loc\, x_2\, nau\, x_2\, n\, x_3\, h\, ili\, p\, ili\, x_4\, x_3\, cat\, x_3\, on$$

Cont.

Definition

- **substitution:** $h : (X \cup \Sigma)^* \longrightarrow \Sigma^*$ with $h(a) = a$ for all $a \in \Sigma$

Example

$h(x_1) = ili, h(x_2) = ci$

$$x_4\, loc\, ci\, nau\, ci\, n\, x_3\, h\, ili\, p\, ili\, x_4\, x_3\, cat\, x_3\, on$$

Cont.

Definition

- **substitution:** $h : (X \cup \Sigma)^* \longrightarrow \Sigma^*$ with $h(a) = a$ for all $a \in \Sigma$

Example

$h(x_1) = ili$, $h(x_2) = ci$, $h(x_3) = i$

$$x_4\, loc\, ci\, nau\, ci\, n\, i\, h\, ili\, p\, ili\, x_4\, i\, cat\, i\, on$$

Cont.

Definition

- **substitution:** $h : (X \cup \Sigma)^* \longrightarrow \Sigma^*$ with $h(a) = a$ for all $a \in \Sigma$

Example

$h(x_1) = ili, h(x_2) = ci, h(x_3) = i, h(x_4) = f$

$$f\, loc\, ci\, nau\, ci\, n\, i\, h\, ili\, p\, ili\, f\, i\, cat\, i\, on$$

Matching Problem

Problem

Decide for a word $w \in \Sigma^*$ *and a pattern* $\alpha \in (X \cup \Sigma)^+$ *whether there exists a substitution* h *with* $h(\alpha) = w$.

Matching Problem

Problem

Decide for a word $w \in \Sigma^*$ *and a pattern* $\alpha \in (X \cup \Sigma)^+$ *whether there exists a substitution* h *with* $h(\alpha) = w$.

Theorem (Angluin)

Match is NP-complete.

Bad and Good News

Bad news

Bad and Good News

Bad news

- still NPC, if every variable only constant number of times (Fernau, Schmid 2013)

Bad and Good News

Bad news

- still NPC, if every variable only constant number of times (Fernau, Schmid 2013)
- natural parametrized variants $w[1]$-hard (Fernau et al. 2015)

Good news (polynomial time)

Bad and Good News

Bad news

- still NPC, if every variable only constant number of times (Fernau, Schmid 2013)
- natural parametrized variants $w[1]$-hard (Fernau et al. 2015)

Good news (polynomial time)

- non-cross

Bad and Good News

Bad news

- still NPC, if every variable only constant number of times (Fernau, Schmid 2013)
- natural parametrized variants $w[1]$-hard (Fernau et al. 2015)

Good news (polynomial time)

- non-cross
- bounded scope coincidence degree

Bad and Good News

Bad news

- still NPC, if every variable only constant number of times (Fernau, Schmid 2013)
- natural parametrized variants $w[1]$-hard (Fernau et al. 2015)

Good news (polynomial time)

- non-cross
- bounded scope coincidence degree
- mildly entwined patterns

Bad and Good News

Bad news

- still NPC, if every variable only constant number of times (Fernau, Schmid 2013)
- natural parametrized variants $w[1]$-hard (Fernau et al. 2015)

Good news (polynomial time)

- non-cross
- bounded scope coincidence degree
- mildly entwined patterns
- bounded tree-width (most general (Reidenbach, Schmid 2014))

Bad and Good News

Bad news

- still NPC, if every variable only constant number of times (Fernau, Schmid 2013)
- natural parametrized variants $w[1]$-hard (Fernau et al. 2015)

Good news (polynomial time)

- non-cross
- bounded scope coincidence degree
- mildly entwined patterns
- bounded tree-width (most general (Reidenbach, Schmid 2014))

On which side is k-locality?

k-local Patterns

Mark variables instead of letters.

Mark variables instead of letters.

Definition

- **skeleton of a word:** word without letters

Mark variables instead of letters.

Definition

- **skeleton of a word:** word without letters

Example

$$x_4\, loc\, x_2\, nau\, x_2\, n\, x_3\, h\, x_1\, p\, x_1\, x_4\, x_3\, cat\, x_3\, on \to$$
$$x_4\, x_2\, x_2\, x_3\, x_1\, x_1\, x_4\, x_3\, x_3$$

Mark variables instead of letters.

Definition

- **skeleton of a word:** word without letters
- **pattern k-local:** skeleton is k-local

Mark variables instead of letters.

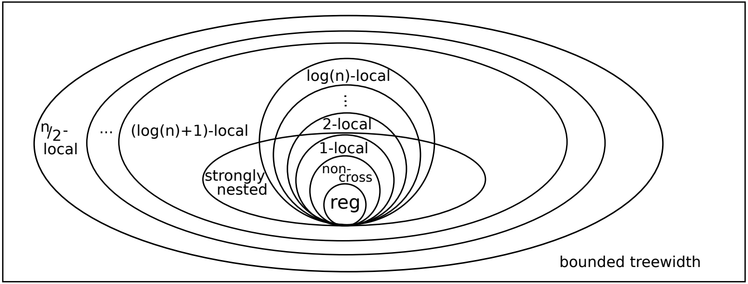
Definition

- **skeleton of a word:** word without letters
- **pattern k-local:** skeleton is k-local

Example

$$x_4\, x_2\, x_2\, x_3\, x_1\, x_1\, x_4\, x_3\, x_3 + (x_1, x_2, x_4, x_3) \Rightarrow \text{2-local}$$

Overview



$\mathrm{Pat}_{\Sigma}^{\leq n}$

incomparable to bounded

- scope coincidence degree
- number of variables
- recurrences of variables

Matching k-local Patterns

Theorem

There exists an algorithm that decides for given $k \in \mathbb{N}$, k-local pattern α and word w whether w matches α in time $\mathcal{O}(k|\alpha||w|^{\max\{3k-1,2k+1\}})$.

Matching k-local Patterns

Theorem

There exists an algorithm that decides for given $k \in \mathbb{N}$, k-local pattern α and word w whether w matches α in time $\mathcal{O}(k|\alpha||w|^{\max\{3k-1,2k+1\}})$.

- k-locality $\Rightarrow$ bounded treewidth $\Rightarrow$ $4k+4$ (Reidenbach, Schmid 2014)

Matching k-local Patterns

Theorem

There exists an algorithm that decides for given $k \in \mathbb{N}$, k-local pattern α and word w whether w matches α in time $\mathcal{O}(k|\alpha||w|^{\max\{3k-1,2k+1\}})$.

- k-locality $\Rightarrow$ bounded treewidth $\Rightarrow 4k+4$ (Reidenbach, Schmid 2014)

Theorem ($k = 1$)

There exists an algorithm that decides for given 1-local pattern α and a word w whether w matches α in time $\mathcal{O}(|\alpha||w|^2 \log(|w|))$

Cont.

- $k = 1$: combinatorics on words + data structures

Cont.

- $k = 1$: combinatorics on words + data structures
- general: compute marking sequence for α, match each variable to possible parts of w which are adjacent (local) to already marked parts

Cont.

- $k = 1$: combinatorics on words + data structures
- general: compute marking sequence for α, match each variable to possible parts of w which are adjacent (local) to already marked parts
- $\Rightarrow$ (k small $\Rightarrow$ matching efficient)

Excursus: Strongly Nested Patterns

Definition (Strongly Nestedness)

- $\alpha \in X^*$ **strongly nested**: $\alpha = \alpha_1[x]^\flat \alpha_2[x]^\flat \alpha_3$
  - $\alpha_1, \alpha_2, \alpha_3 \in (X \backslash \{x\})^*$
  - $\mathrm{Var}(\alpha_1\alpha_3) \cap \mathrm{Var}(\alpha_2) = \emptyset$

Excursus: Strongly Nested Patterns

Definition (Strongly Nestedness)

- $\alpha \in X^*$ **strongly nested**: $\alpha = \alpha_1 [x]^\flat \alpha_2 [x]^\flat \alpha_3$
  - $\alpha_1, \alpha_2, \alpha_3 \in (X \backslash \{x\})^*$
  - $\operatorname{Var}(\alpha_1 \alpha_3) \cap \operatorname{Var}(\alpha_2) = \emptyset$

Example

$$x_8 x_7 \underbrace{x_6 \underbrace{x_1 x_2 x_1}\ \underbrace{x_3 x_4 x_5 x_3}\ x_6}$$

$$\underbrace{x_8 x_7 x_6 x_1 x_2 x_1 x_3 x_4 x_5 x_3 x_6 x_8 x_9}$$

Excursus: Strongly Nested Patterns

Definition (Strongly Nestedness)

- $\alpha \in X^*$ **strongly nested**: $\alpha = \alpha_1[x]^\flat\alpha_2[x]^\flat\alpha_3$
  - $\alpha_1, \alpha_2, \alpha_3 \in (X \setminus \{x\})^*$
  - $\mathrm{Var}(\alpha_1\alpha_3) \cap \mathrm{Var}(\alpha_2) = \emptyset$

- Strongly nested patterns are $\Theta(\log(|\alpha|))$-local.

Strongly Nested Patterns

Theorem

Deciding whether a pattern is strongly nested is doable in linear time.

Strongly Nested Patterns

Theorem

Deciding whether a pattern is strongly nested is doable in linear time.

Theorem

There exists an algorithm that, given a strongly nested pattern α and a word w decides whether w matches α in time $\mathcal{O}(|\alpha||w|^3)$

Strongly Nested Patterns

Theorem

Deciding whether a pattern is strongly nested is doable in linear time.

Theorem

There exists an algorithm that, given a strongly nested pattern α and a word w decides whether w matches α in time $\mathcal{O}(|\alpha||w|^3)$

- improvement from mildly entwined patterns ($\mathcal{O}(|\alpha||w|^6$ (Reidenbach, Schmid 2014))
- algorithm uses the *tree-structure* of the patterns

Questions?

thanks

danke kiitos

děkuji grazie

dziękuję merci datorită

gracias tak спаси́бо

Questions?

thanks

danke kiitos

děkuji grazie

dziękuję merci datorită

gracias tak спаси́бо

all 1-local