Matching Patterns with Variables Under Simon's Congruence

Pamela Fleischmann, Sungmin Kim, Tore Koß, Florin Manea, Dirk Nowotka, Stefan Siemer, **Max Wiedenhöft**

What you will See

Introduction & Problems

Complexity Results

Conclusion

Words – Needed Terminology

Σ **Alphabet of constants** (of constant size σ)

$\Sigma := \{ \mathbf{a}, \mathbf{b}, \mathbf{n} \}$ $(\sigma = \mathbf{3})$

Σ^* **All words over alphabet**

$\mathbf{banana} \in \{ a, b, n \}^*$

$|w|$ **Length of a word**

$|banana| = \mathbf{6}$

ε **Empty word** $(|\varepsilon| = 0)$

$banana\boldsymbol{\varepsilon} = banana$

Subsequences

(Scattered Factors)

Subsequences

(Scattered Factors)

bananacakes

Subsequences

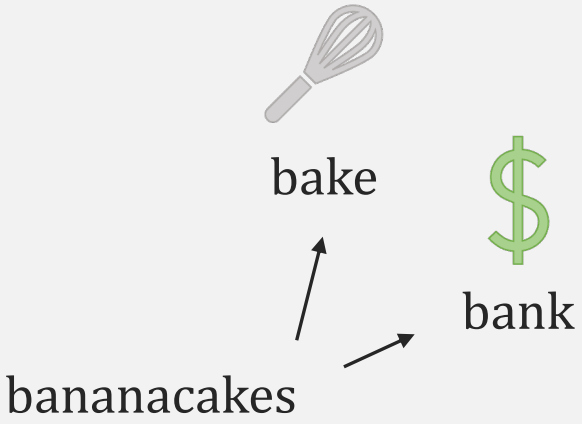
(Scattered Factors)

bake

bananacakes

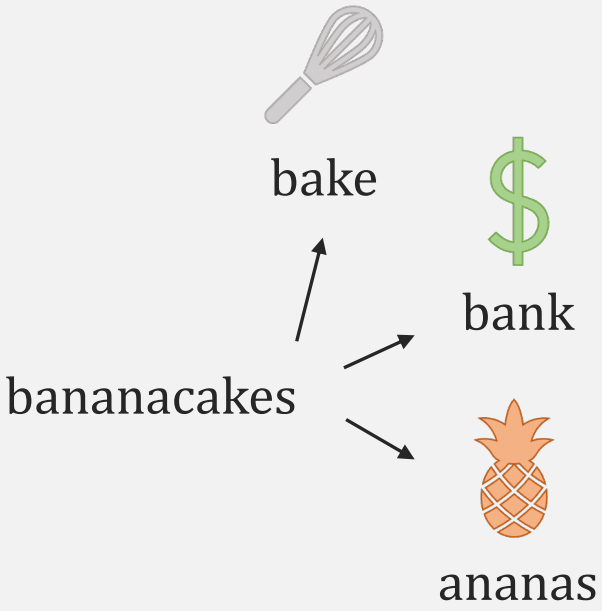
Subsequences

(Scattered Factors)



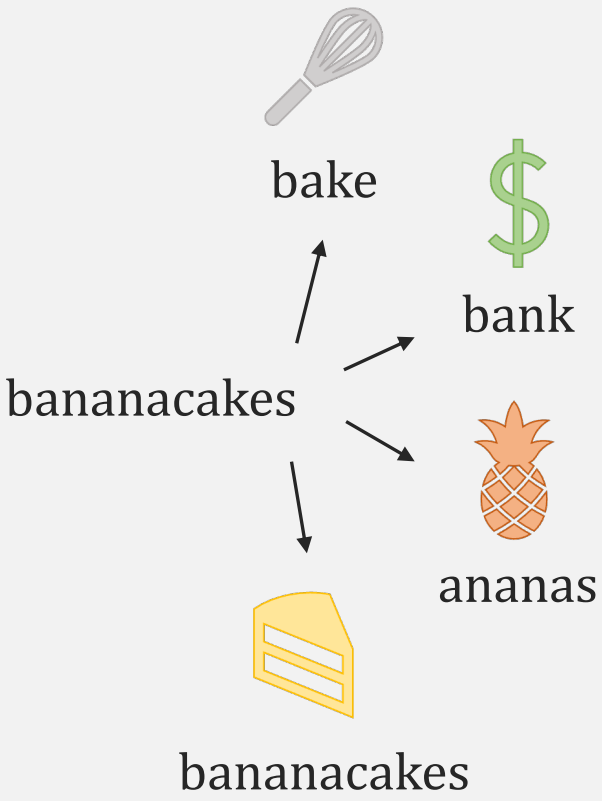
Subsequences

(Scattered Factors)



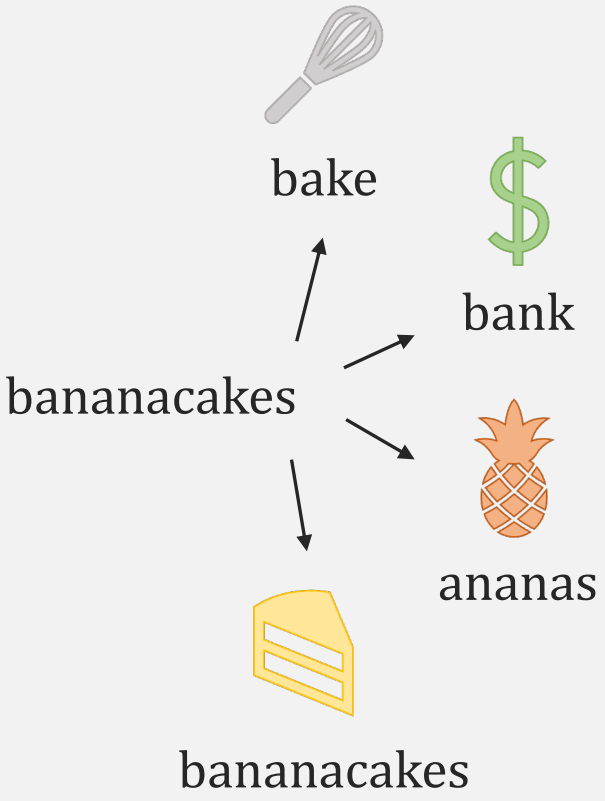
Subsequences

(Scattered Factors)



Subsequences

(Scattered Factors)



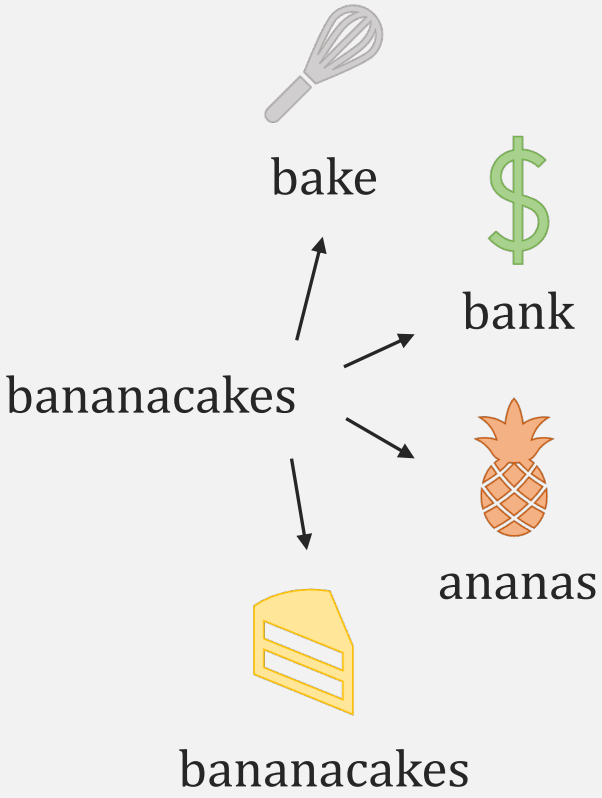
Notation

$\mathbb{S}(w)$ Set of all subsequences of w

$\mathbb{S}_k(w)$ Set of all subsequences of w up to length k

Subsequences

(Scattered Factors)



Notation

$\mathbb{S}(w)$ Set of all subsequences of w

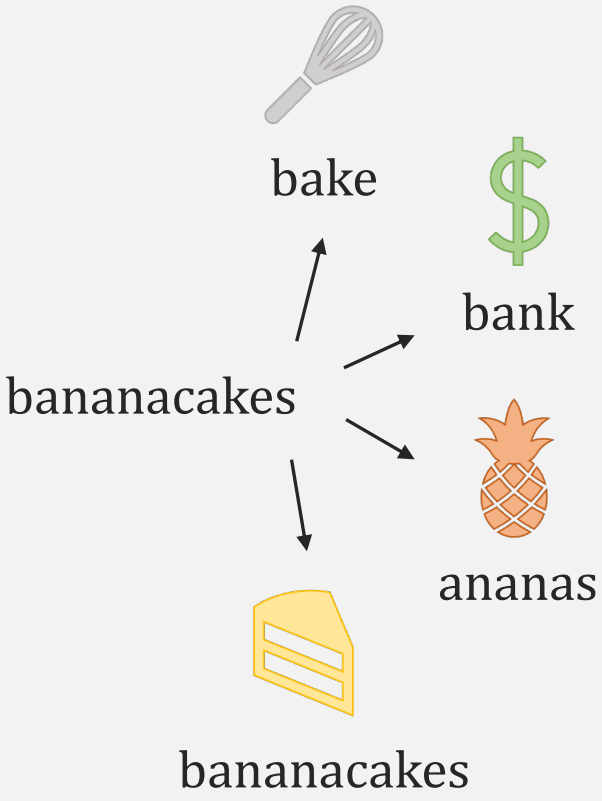
$\mathbb{S}_k(w)$ Set of all subsequences of w up to length k

k-Universality

w is k-universal ($\iota(w) = k$) iff $\mathbb{S}_k(w) = \Sigma^{\leq k}$

Subsequences

(Scattered Factors)



Notation

$\mathbb{S}(w)$ Set of all subsequences of w

$\mathbb{S}_k(w)$ Set of all subsequences of w up to length k

k-Universality

w is k-universal ($\iota(w) = k$) iff $\mathbb{S}_k(w) = \Sigma^{\leq k}$

Simon's Congruence

w_1 and w_2 are k-Simon congruent ($w_1 \sim_k w_2$) iff:

$$\mathbb{S}_k(w_1) = \mathbb{S}_k(w_2)$$

Patterns with Variables

Patterns with Variables

Infinite set of variables
(disjoint to Σ)

$$X = \{x_1, x_2, \ldots\}$$

Patterns with Variables

X

Infinite set of variables
(disjoint to Σ)

$X = \{x_1, x_2, \ldots\}$

α

Patterns
(over alphabet und variables)

$\mathrm{b}x_1x_1\mathrm{a}x_2 \in (\Sigma \cup X)^*$

Patterns with Variables

X

Infinite set of variables
(disjoint to Σ)

$X = \{x_1, x_2, \ldots\}$

α

Patterns
(over alphabet und variables)

$\mathrm{b}x_1x_1\mathrm{a}x_2 \in (\Sigma \cup X)^*$

h

Substitutions
(map patterns to words)

$h : (\Sigma \cup X)^* \to \Sigma^*$

$\mathrm{b}\ x_1\ x_1\ \mathrm{a}\ x_2$ → $\mathrm{b\ an\ an\ a\ cake}$

$\mathrm{b\ an\ an\ a}\ \varepsilon$

Patterns with Variables

X

Infinite set of variables
(disjoint to Σ)

$X = \{x_1, x_2, \ldots\}$

α

Patterns
(over alphabet und variables)

$\mathrm{b}x_1x_1\mathrm{a}x_2 \in (\Sigma \cup X)^*$

h

Substitutions
(map patterns to words)

$h : (\Sigma \cup X)^* \to \Sigma^*$

$\mathrm{b}\ x_1\ x_1\ \mathrm{a}\ x_2$ → $\mathrm{b\ an\ an\ a\ cake}$

$\mathrm{b\ an\ an\ a}\ \varepsilon$

Fundamental Problem: Match

Given

Pattern α

Word w

Problem

Does there exist a substitution h with

$$h(\alpha) = w$$

Patterns with Variables

X

Infinite set of variables
(disjoint to Σ)

$X = \{x_1, x_2, \ldots\}$

α

Patterns
(over alphabet und variables)

$\mathrm{b}x_1x_1\mathrm{a}x_2 \in (\Sigma \cup X)^*$

h

Substitutions
(map patterns to words)

$h : (\Sigma \cup X)^* \to \Sigma^*$

$\mathrm{b}\ x_1\ x_1\ \mathrm{a}\ x_2$ → $\mathrm{b\ an\ an\ a\ cake}$

$\mathrm{b\ an\ an\ a}\ \varepsilon$

Fundamental Problem: Match

Given

Pattern α

Word w

Problem

Does there exist a substitution h with

$\boldsymbol{h(\alpha) = w}$

!

Match is NP-complete

Three Main Problems

Three Main Problems

1

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

Three Main Problems

1

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

2

MatchSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$$h(\alpha) \sim_k w$$

Three Main Problems

1

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$\boldsymbol{\iota(h(\alpha)) = k}$

2

MatchSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$\boldsymbol{h(\alpha) \sim_k w}$

3

WESimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$\boldsymbol{h(\alpha) \sim_k h(\beta)}$

Let’s see how hard these Problems are!

Introduction & Problems

Complexity Results

Conclusion

1

MatchUniv

MatchUniv

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

MatchUniv

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

Example

$\Sigma = \{a, b, c\}$

$k = 3$

$\alpha = abc\ abc\ \boldsymbol{x_1}$

1

MatchUniv

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

Example

$\Sigma = \{a, b, c\}$

$k = 3$

$\alpha = \text{abc abc } \boldsymbol{x_1} \longrightarrow \text{abc abc } \mathbf{abc}$

MatchUniv

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

Example

$\Sigma = \{a, b, c\}$

$k = 3$

$\alpha = \text{abc abc } \boldsymbol{x_1} \longrightarrow \text{abc abc } \mathbf{abc}$

$k = 1$?

1

MatchUniv

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

Example

$\Sigma = \{a, b, c\}$

$k = 3$

$\alpha = \text{abc abc } \boldsymbol{x_1} \longrightarrow \text{abc abc } \mathbf{abc}$

$k = 1$?

1

MatchUniv

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

Example

$\Sigma = \{a, b, c\}$

$k = 3$

$\alpha = \text{abc abc } x_1 \longrightarrow \text{abc abc } \mathbf{abc}$

$k = 1$?

$\Sigma = \{a, b, c, \mathbf{d}\}$?

$\alpha = \text{abc abc } x_1$

1

MatchUniv

MatchUniv

Given

Pattern α

Number k

Problem

Can we find substitution h with

$$\iota(h(\alpha)) = k$$

Example

$\Sigma = \{a, b, c\}$

$k = 3$

$\alpha = \text{abc abc } x_1 \longrightarrow \text{abc abc } \mathbf{abc}$

$k = 1$? 

$\Sigma = \{a, b, c, \mathbf{d}\}$?

$\alpha = \text{abc abc } x_1 \longrightarrow \text{abc abc } \mathbf{d}$

MatchUniv is NP-Hard

MatchUniv is NP-Hard

Reduce from 3CNFSAT!

MatchUniv is NP-Hard

Reduce from 3CNFSAT!

$(x_1 \vee \boldsymbol{x_2} \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \boldsymbol{\bar{x}_3})$

True with e.g. $\boldsymbol{x_2 = 1}$ and $\boldsymbol{x_3 = 0}$

1 MatchUniv is NP-Hard

Reduce from 3CNFSAT!

$(x_1 \vee \boldsymbol{x_2} \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \boldsymbol{\bar{x}_3})$

True with e.g. $\boldsymbol{x_2 = 1}$ and $\boldsymbol{x_3 = 0}$

Reduction Layout

$\Rightarrow$

Map to MatchUniv instance
(with specific k)

$z_i\ u_i$

Two pattern variables z_i, u_i
(per clause variable $\mathrm{x_i}$)

1 MatchUniv is NP-Hard

Reduce from 3CNFSAT!

$(x_1 \vee \boldsymbol{x_2} \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \boldsymbol{\bar{x}_3})$

True with e.g. $\boldsymbol{x_2 = 1}$ and $\boldsymbol{x_3 = 0}$

Reduction Layout

$\Rightarrow$

Map to MatchUniv instance (with specific k)

$z_i\ u_i$

Two pattern variables z_i, u_i (per clause variable x_i)

1010..

Binarisation Gadget

1 | 0

Boolean Gadget

$1 \leftrightarrow 0$

Complementation Gadget

$\wedge\vee$

Clause Gadget

MatchUniv is in NP

1

MatchUniv is in NP

Idea

Guess substitution h and check whether $\iota\big(h(\alpha)\big) = k$

1

MatchUniv is in NP

Idea

Guess substitution h and check whether $\iota\big(h(\alpha)\big) = k$

Problem

Potential exponential substitution size!

1

MatchUniv is in NP

Idea

Guess substitution h and check whether $\iota\big(h(\alpha)\big) = k$

Problem

Potential exponential substitution size!

Solution

Find smaller representation for words with certain universality

1

MatchUniv is in NP

Idea

Guess substitution h and check whether $\iota\big(h(\alpha)\big) = k$

Problem

Potential exponential substitution size!

Solution

Find smaller representation for words with certain universality

⇒ **Universality Signatures**

1 Subsequence Universality Signatures

(Schnoebelen & Veron 2023)

Subsequence Universality Signatures

(Schnoebelen & Veron 2023)

<u>Compression of words using specific tuple</u>

preserves universality of words

Subsequence Universality Signatures

(Schnoebelen & Veron 2023)

Compression of words using specific tuple

preserves universality of words

Concatenation Property (Schnoebelen & Veron 2023)

given universality signatures $s(w_1)$ and $s(w_2)$
$\Rightarrow s(w_1 w_2)$ computable in polynomial time

1 Subsequence Universality Signatures

(Schnoebelen & Veron 2023)

Compression of words using specific tuple

preserves universality of words

Concatenation Property (Schnoebelen & Veron 2023)

given universality signatures $s(w_1)$ and $s(w_2)$
$\Rightarrow s(w_1 w_2)$ computable in polynomial time

Constant Signature Validity / Existence Check

given signature tuple
$\Rightarrow$ constant bound on word length to check if signature tuple exists
(if alphabet is of constant size)

NP-Containment by Verification

NP-Containment by Verification

Given

pattern α and number k

1 NP-Containment by Verification

Given

pattern α and number k

Verification Approach

1: Guess signature tuple $\boldsymbol{s}(\boldsymbol{h}(\boldsymbol{x}))$ of potential substitution $\boldsymbol{h}$ for all $\boldsymbol{x} \in \mathbf{var}(\boldsymbol{\alpha})$ $O(|\mathrm{var}(\alpha)|)$

1 NP-Containment by Verification

Given

pattern α and number k

Verification Approach

1: Guess signature tuple $\boldsymbol{s}(\boldsymbol{h}(\boldsymbol{x}))$ of potential substitution $\boldsymbol{h}$ for all $\boldsymbol{x} \in \mathbf{var}(\boldsymbol{\alpha})$ $O(|\mathrm{var}(\alpha)|)$

2: Check if the signatures are valid $O(|\mathrm{var}(\alpha)|)$

1 NP-Containment by Verification

Given

pattern α and number k

Verification Approach

1: Guess signature tuple $\boldsymbol{s}(\boldsymbol{h}(\boldsymbol{x}))$ of potential substitution $\boldsymbol{h}$ for all $\boldsymbol{x} \in \mathbf{var}(\boldsymbol{\alpha})$ $O(|\mathrm{var}(\alpha)|)$

2: Check if the signatures are valid $O(|\mathrm{var}(\alpha)|)$

3: Check if the signatures lead to a substitution of universality k $O(\mathrm{poly}(|h(\alpha)|))$

1 NP-Containment by Verification

Given

pattern α and number k

Verification Approach

1: Guess signature tuple $\boldsymbol{s}(\boldsymbol{h}(\boldsymbol{x}))$ of potential substitution $\boldsymbol{h}$ for all $\boldsymbol{x} \in \mathbf{var}(\boldsymbol{\alpha})$ $O(|\mathrm{var}(\alpha)|)$

2: Check if the signatures are valid $O(|\mathrm{var}(\alpha)|)$

3: Check if the signatures lead to a substitution of universality k $O(\mathrm{poly}(|h(\alpha)|))$

MatchUniv is NP-complete!

1 NP-Containment by Verification

Given

pattern α and number k

Verification Approach

1: Guess signature tuple $\boldsymbol{s}(\boldsymbol{h}(\boldsymbol{x}))$ of potential substitution $\boldsymbol{h}$ for all $\boldsymbol{x} \in \mathbf{var}(\boldsymbol{\alpha})$ $O(|\mathrm{var}(\alpha)|)$

2: Check if the signatures are valid $O(|\mathrm{var}(\alpha)|)$

3: Check if the signatures lead to a substitution of universality k $O(\mathrm{poly}(|h(\alpha)|))$

MatchUniv is NP-complete!

Additional Results (under constant alphabet size)

- MatchUniv(α, k) in P if there exists unique variable in α
- MatchUniv(α, k) in P if the number of variables is bound by a constant

2

MatchSimon

2

MatchSimon

MatchSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$$h(\alpha) \sim_k w$$

2

MatchSimon

MatchSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$h(\alpha) \sim_k w$

MatchSimon$(\alpha, w, |w|)$ is equivalent to Match(α, w)

⇒ **MatchSimon is NP-hard**

2

MatchSimon

MatchSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$$h(\alpha) \sim_k w$$

MatchSimon$(\alpha, w, |w|)$ is equivalent to Match(α, w)

⇒ **MatchSimon is NP-hard**

Not strict! Solution trivial in many cases!

($h(\alpha) \sim_{k+1} w$ allowed!)

2

MatchSimon

MatchSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$h(\alpha) \sim_k w$

MatchSimon$(\alpha, w, |w|)$ is equivalent to Match(α, w)

⇒ **MatchSimon is NP-hard**

Not strict! Solution trivial in many cases!

($h(\alpha) \sim_{k+1} w$ allowed!)

Motivates to consider stricter version!

2

MatchStrictSimon

MatchStrictSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$$h(\alpha) \sim_k w \;\&\; h(\alpha) \not\sim_{k+1} w$$

2

MatchStrictSimon

MatchStrictSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$$h(\alpha) \sim_k w \;\&\; h(\alpha) \not\sim_{k+1} w$$

MatchStrictSimon is NP-hard!

Use same reduction approach used for MatchUniv

2

MatchStrictSimon

MatchStrictSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$$h(\alpha) \sim_k w \;\&\; h(\alpha) \not\sim_{k+1} w$$

MatchStrictSimon is NP-hard!

Use same reduction approach used for MatchUniv

- encode 3CNFSAT instance in pattern α similarly
- set k exactly the same way

2

MatchStrictSimon

MatchStrictSimon

Given

Pattern α

Number k

Word w

Problem

Can we find substitution h with

$$h(\alpha) \sim_k w \;\&\; h(\alpha) \nsim_{k+1} w$$

MatchStrictSimon is NP-hard!

Use same reduction approach used for MatchUniv

- encode 3CNFSAT instance in pattern α similarly
- set k exactly the same way

3CNFSAT instance solvable if and only if MatchStrictSimon(α, k, w) true

2

NP-Containment Sketch

2

NP-Containment Sketch

MatchSimon

MatchStrictSimon

2

NP-Containment Sketch

MatchSimon **MatchStrictSimon**

$$\mathbf{k} \geq |w|$$

NP-Containment Sketch

MatchSimon **MatchStrictSimon**

$$\mathbf{k} \geq |w|$$

Check if α matches w

NP-Containment Sketch

MatchSimon	**MatchStrictSimon**
$\mathbf{k} \geq \lvert w \rvert$	
Check if α matches w	Always answer no!

2

NP-Containment Sketch

MatchSimon	**MatchStrictSimon**
$\mathbf{k} \geq \lvert w \rvert$	
Check if α matches w	Always answer no!
$\boldsymbol{k} < \lvert \boldsymbol{w} \rvert$	

2

NP-Containment Sketch

MatchSimon	**MatchStrictSimon**
$\mathbf{k} \geq \lvert w \rvert$	
Check if α matches w	Always answer no!
$\boldsymbol{k} < \lvert \boldsymbol{w} \rvert$	
Encode variable substitutions in words of maximum size	
$\boldsymbol{k}^{\boldsymbol{\sigma}}$	$(\boldsymbol{k}+\mathbf{1})^{\boldsymbol{\sigma}}$

2

NP-Containment Sketch

MatchSimon	**MatchStrictSimon**
$\mathbf{k} \geq \lvert w \rvert$	
Check if α matches w	Always answer no!
$\boldsymbol{k} < \lvert \boldsymbol{w} \rvert$	
Encode variable substitutions in words of maximum size	
$\boldsymbol{k^{\sigma}}$	$\boldsymbol{(k+1)^{\sigma}}$
Verify encoding by checking substitution of length at most	
$\boldsymbol{\alpha \cdot k^{\sigma}}$	$\boldsymbol{\alpha \cdot (k+1)^{\sigma}}$

2

NP-Containment Sketch

MatchSimon	MatchStrictSimon
$\mathbf{k} \geq \lvert w \rvert$	
Check if α matches w	Always answer no!
$\boldsymbol{k} < \lvert \boldsymbol{w} \rvert$	
Encode variable substitutions in words of maximum size	
$\boldsymbol{k^{\sigma}}$	$\boldsymbol{(k+1)^{\sigma}}$
Verify encoding by checking substitution of length at most	
$\boldsymbol{\alpha \cdot k^{\sigma}}$	$\boldsymbol{\alpha \cdot (k+1)^{\sigma}}$

MatchSimon & MatchStrictSimon are NP-complete

2

NP-Containment Sketch

MatchSimon	MatchStrictSimon
$\mathbf{k} \geq \lvert w \rvert$	
Check if α matches w	Always answer no!
$\boldsymbol{k} < \lvert \boldsymbol{w} \rvert$	
Encode variable substitutions in words of maximum size	
$\boldsymbol{k^{\sigma}}$	$\boldsymbol{(k+1)^{\sigma}}$
Verify encoding by checking substitution of length at most	
$\boldsymbol{\alpha \cdot k^{\sigma}}$	$\boldsymbol{\alpha \cdot (k+1)^{\sigma}}$

MatchSimon & MatchStrictSimon are NP-complete

For regular patterns MatchSimon and MatchStrictSimon are in P.

3

WESimon

3

WESimon

WESimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$$h(\alpha) \sim_k h(\beta)$$

3

WESimon

WESimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$$h(\alpha) \sim_k h(\beta)$$

MatchSimon is particular case of **WESimon**

⇒ **WESimon is NP-hard**

3

WESimon

WESimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$$h(\alpha) \sim_k h(\beta)$$

MatchSimon is particular case of **WESimon**

⇒ **WESimon is NP-hard**

In NP? Yes!

- If $k \leq |\alpha| + |\beta|$ then same as MatchSimon
- If $k > |\alpha| + |\beta|$ and $\beta \in \Sigma^*$ then instance of MatchSimon!
- If $k > |\alpha| + |\beta|$ and both contain variables then always yes!

3

WESimon

WESimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$$h(\alpha) \sim_k h(\beta)$$

MatchSimon is particular case of **WESimon**

⇒ **WESimon is NP-hard**

In NP? Yes!

- If $k \leq |\alpha| + |\beta|$ then same as MatchSimon
- If $k > |\alpha| + |\beta|$ and $\beta \in \Sigma^*$ then instance of MatchSimon!
- If $k > |\alpha| + |\beta|$ and both contain variables then always yes!

⇒ **WESimon is NP-complete!**

3

WESimon

WESimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$$h(\alpha) \sim_k h(\beta)$$

MatchSimon is particular case of **WESimon**

⇒ **WESimon is NP-hard**

In NP? Yes!

- If $k \leq |\alpha| + |\beta|$ then same as MatchSimon
- If $k > |\alpha| + |\beta|$ and $\beta \in \Sigma^*$ then instance of MatchSimon!
- If $k > |\alpha| + |\beta|$ and both contain variables then always yes!

⇒ **WESimon is NP-complete!**

Again, not strict!

($h(\alpha) \sim_{k+1} h(\beta)$ allowed)

3

WEStrictSimon

WEStrictSimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$$\boldsymbol{h(\alpha) \sim_k h(\beta)} \;\&\; \boldsymbol{h(\alpha) \not\sim_{k+1} h(\beta)}$$

3

WEStrictSimon

WEStrictSimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$$h(\alpha) \sim_k h(\beta) \;\&\; h(\alpha) \not\sim_{k+1} h(\beta)$$

Similar to MatchStrictSimon!

- NP-hardness: similar 3CNFSAT reduction
- NP-contaiment: as long as $k \leq |\alpha| + |\beta|$

3

WEStrictSimon

WEStrictSimon

Given

Pattern α

Pattern β

Number k

Problem

Can we find substitution h with

$$h(\alpha) \sim_k h(\beta) \;\&\; h(\alpha) \not\sim_{k+1} h(\beta)$$

Similar to MatchStrictSimon!

- NP-hardness: similar 3CNFSAT reduction
- NP-containment: as long as $k \leq |\alpha| + |\beta|$

NP-containment open for k without upper bound!

What have we seen?

Introduction & Problems

Complexity Results

Conclusion

Conclusion

Conclusion

1. **MatchUniv** ⇒ NP-complete

Conclusion

1. **MatchUniv** ⇒ NP-complete

2. **MatchSimon**
(strict and not strict variant) ⇒ both NP-complete

Conclusion

1. **MatchUniv** ⇒ NP-complete
2. **MatchSimon** (strict and not strict variant) ⇒ both NP-complete
3. **WESimon** (strict and not strict variant) ⇒ WESimon NP-complete
   ⇒ WEStrictSimon NP-complete for $k \leq |\alpha| + |\beta|$

Conclusion

1. **MatchUniv** ⇒ NP-complete

2. **MatchSimon**
(strict and not strict variant) ⇒ both NP-complete

3. **WESimon**
(strict and not strict variant) ⇒ WESimon NP-complete
⇒ WEStrictSimon NP-complete for $k \leq |\alpha| + |\beta|$

What are the parameterised complexities of these problems w.r.t.
→ the size of the alphabet?
→ the number of variables of the considered patterns?

Conclusion

1. **MatchUniv** ⇒ NP-complete

2. **MatchSimon**
(strict and not strict variant) ⇒ both NP-complete

3. **WESimon**
(strict and not strict variant) ⇒ WESimon NP-complete
⇒ WEStrictSimon NP-complete for $k \leq |\alpha| + |\beta|$

What are the parameterised complexities of these problems w.r.t.
→ the size of the alphabet?
→ the number of variables of the considered patterns?

Thank You!

MatchUniv NP-Hardness

Final Assemblage of Gadgets

$$\alpha = \underbrace{\pi_{\#}\pi_{\$}}\ \underbrace{\pi_1^z\pi_1^u \dots \pi_n^z\pi_n^u}\ \underbrace{\xi_1 \dots \xi_n}\ \underbrace{\delta_1 \dots \delta_m}$$

Binarisation Gadgets

$h(z_i), h(u_i) \in \{1,0\}^*$

$\Rightarrow \boldsymbol{\iota(h(\pi_{\#}\pi_{\$}))} = \mathbf{2}$

Boolean Gadgets

$h(z_i), h(u_i) \in L(0^*|1^*)$

$\Rightarrow \boldsymbol{\iota(h(\pi_1^z\pi_1^u \dots \pi_n^z\pi_n^u))} = \mathbf{4n}$

Complementation Gadgets

$\boldsymbol{z_i} \in \{\mathbf{1}\}^+$ and $\boldsymbol{u_i} \in \{\mathbf{0}\}^+$

or

$\boldsymbol{z_i} \in \{\mathbf{0}\}^+$ and $\boldsymbol{u_i} \in \{\mathbf{1}\}^+$

$\Rightarrow \boldsymbol{\iota(h(\xi_1 \dots \xi_n))} = \mathbf{n}$

Clause Gadgets

$\forall j \in [m]: \iota\left(h\left(\delta_j\right)\right) = 1$

$\Rightarrow \boldsymbol{\iota(h(\delta_1 \dots \delta_m))} = \mathbf{m}$

3CNFSAT Instance solvable if and only if

$$\iota\big(h(\alpha)\big) = 5n + m + 2$$

(Otherwise, this universality can't be reached!)

MatchUniv is NP-Hard!

Universality Signatures

Subsequence Universality Signature $s(w)$

Consider word $w \in \Sigma^*$
Then $s(w) = (\gamma, \mathcal{K}, \mathcal{R})$ with

γ permutation of $\mathrm{alph}(w)$
(first occurrence of each letter in w)

$\mathcal{K}$ array of length $|\sigma|$
($\mathcal{K}[i]$ is equal to universality of suffix of w after first occurrence of $\gamma[i]$)

$\mathcal{R}$ array of length $|\sigma|$
($\mathcal{R}[i]$ contains alphabet of rest of suffix of w after first occurrence of $\gamma[i]$)

Example

$w = \mathbf{b}\ \mathbf{c}\ \mathrm{c}\ \mathbf{a}\ \mathrm{a}\ \mathrm{b}\ \mathrm{c}\ \mathrm{b}\ \mathrm{a}\ \mathrm{c}\ \mathrm{a}\ \mathrm{a}$

$\gamma = (\mathbf{b}, \mathbf{c}, \mathbf{a})$

$\mathcal{K} = (\mathbf{2}, \mathbf{2}, \mathbf{2})$

$\mathcal{R} = (\{\mathbf{a}, \mathbf{c}\}, \{\mathbf{a}, \mathbf{c}\}, \{\mathbf{a}\})$