Pseudo-Inversion on Formal Languages

Da-Jung Cho[1] Yo-Sub Han[1] Shin-Dong Kang[1]
Hwee Kim[1] Sang-Ki Ko[1] Kai Salomaa[2]

[1]Department of Computer Science
Yonsei University

[2]School of Computing
Queen's University

Unconventional Computation & Natural Computation, 2014

Outline

1. Introduction
   - Motivation
   - Notations
   - Problems
2. Main Results
   - Closure Properties of Pseudo-Inversion
   - Pseudo-Inversion-Free Decidability Problems
3. Conclusion
   - Summary

Inversion Operation

- For evolutionary biology, inversion operation is well-studied.

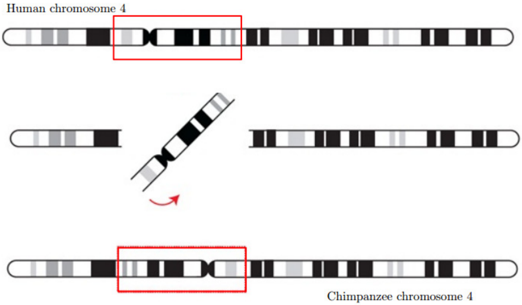


Figure: An example of chromosomal inversion between human and chimpanzee [Miller and Reis,1982]

Inversion Operation

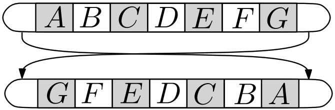
- For evolutionary biology, inversion operation is well-studied.

Figure: How similar are they?

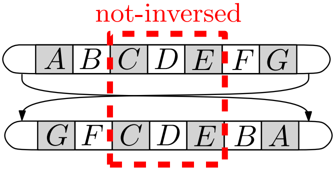
Related Works

- Inversion
  - Yokomori and Kobayashi, *"DNA evolutionary linguistics and RNA structure modeling: A computational approach"*, 1995.
  - Dassow et al., *"Operations and language generating devices suggested by the genome evolution"*, 2002.
- Hairpin inversion
  - Daley et al., *"Closure and decidability properties of some language classes with respect to ciliate bio-operations"*, 2003.
  - Daley et al., *"Families of languages defined by ciliate bio-operations"*, 2004.
- Alignment and string matching with inversion
  - Schöniger and Waterman, *"A local algorithm for DNA sequence alignment with inversions"*, 1992.
  - Vellozo et al., *"Alignment with non-overlapping inversions in $O(n^3)$-time"*, 2006.
  - Cantone et al., *"Efficient string-matching allowing for non-overlapping inversions"*, 2013.

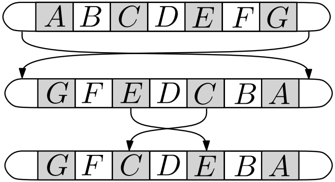
Three Cases of Inversion in Practice



(a) Complete inversion

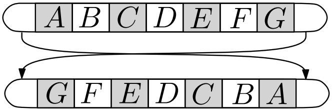


(b) Incomplete inversion

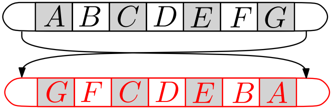


(c) Inversions

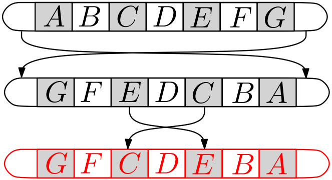
Three Cases of Inversion in Practice



(a) Complete inversion



(b) Incomplete inversion



(c) Inversions

Pseudo-inversion

Pseudo-Inversion

Definition

For a string $w = uxv \in \Sigma^*$, we define the pseudo-inversion of w to be

$$\mathbb{PI}(w) = \{v^R x u^R \mid u, x, v \in \Sigma^* \text{ and } vu \neq \lambda\}.$$

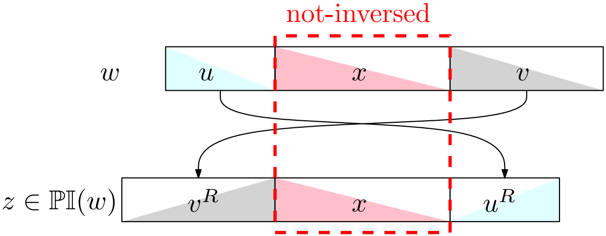


Figure: An example of $\mathbb{PI}$. For $w = uxv$, $z = v^R x u^R \in \mathbb{PI}(w)$

Pseudo-Inversion

Definition

For a string $w = uxv \in \Sigma^*$, we define the pseudo-inversion of w to be

$$\mathbb{PI}(w) = \{v^R x u^R \mid u, x, v \in \Sigma^* \text{ and } vu \neq \lambda\}.$$

Given a language L,

$$\mathbb{PI}(L) = \bigcup_{w \in L} \mathbb{PI}(w).$$

Iterated Pseudo-Inversion

Given a string $w = uxv$ over Σ,

- $\mathbb{PI}^1(w) = \mathbb{PI}(w)$,
- $\mathbb{PI}^{i+1}(w) = \mathbb{PI}(\mathbb{PI}^i(w))$.

Definition

Iterated pseudo-inversion of w is defined as

$$\mathbb{PI}^*(w) = \bigcup_{i=1}^{\infty} \mathbb{PI}^i(w).$$

Given a language L,

$$\mathbb{PI}^*(L) = \bigcup_{w \in L} \mathbb{PI}^*(w).$$

Pseudo-Inversion-Free

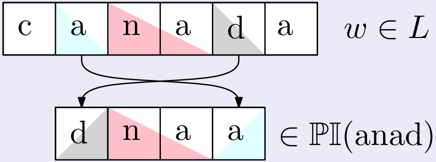
Definition

A given language $L \in \Sigma^*$ is *pseudo-inversion-free* if no string in L is a pseudo-inversion substring of any other string in L.

Example

Let $L = \{canada, dnaa\}$.

- Since *dnaa* is a pseudo-inversion substring of *canada*,



- L is not pseudo-inversion-free.

Problems

1. Closure property
   - Closure properties of pseudo-inversion
   - Closure properties of iterated pseudo-inversion

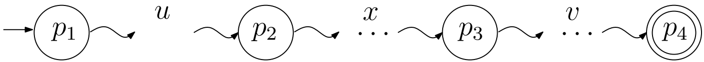
2. Decision problem
   - Given two strings w and z of same length, is $z \in \mathbb{PI}(w)$?
   - Given a language L, is L pseudo-inversion-free?

Closure Property of $\mathbb{PI}(L)$

Theorem

If L is a regular language, then $\mathbb{PI}(L)$ is also regular.

FA $A = (Q, \Sigma, \delta, Q_0, F)$



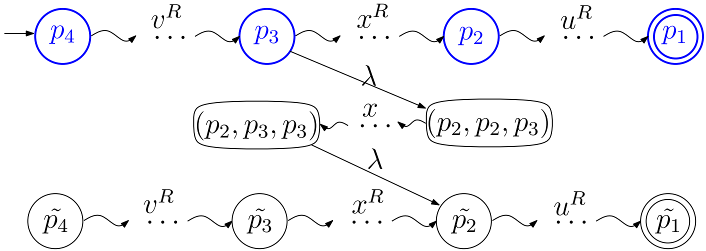
- $Q = \{p_1, p_2, p_3, p_4\}$
- $Q_0 = \{p_1\}$
- $F = \{p_4\}$

Closure Property of $\mathbb{PI}(L)$

Theorem

If L is a regular language, then $\mathbb{PI}(L)$ is also regular.

FA $B = (P, \Sigma, \gamma, P_0, F_B)$



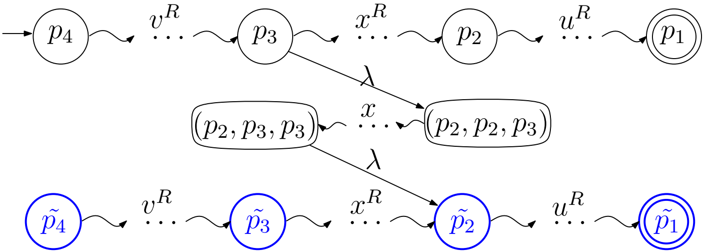
- $P = Q \cup \tilde{Q} \cup Q^3$
- $P_0 = \{p_4\}$
- $F_B = \{p_1, \tilde{p_1}\}$

Closure Property of $\mathbb{PI}(L)$

Theorem

If L is a regular language, then $\mathbb{PI}(L)$ is also regular.

FA $B = (P, \Sigma, \gamma, P_0, F_B)$



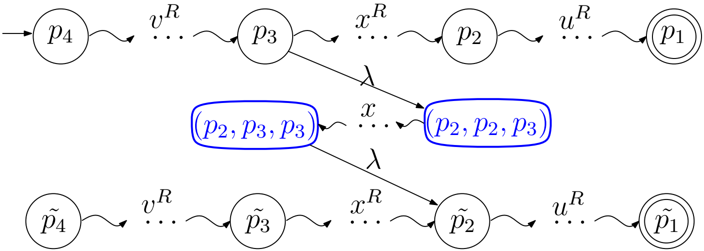
- $P = Q \cup \tilde{Q} \cup Q^3$
- $P_0 = \{p_4\}$
- $F_B = \{p_1, \tilde{p_1}\}$

Closure Property of $\mathbb{PI}(L)$

Theorem

If L is a regular language, then $\mathbb{PI}(L)$ is also regular.

FA $B = (P, \Sigma, \gamma, P_0, F_B)$

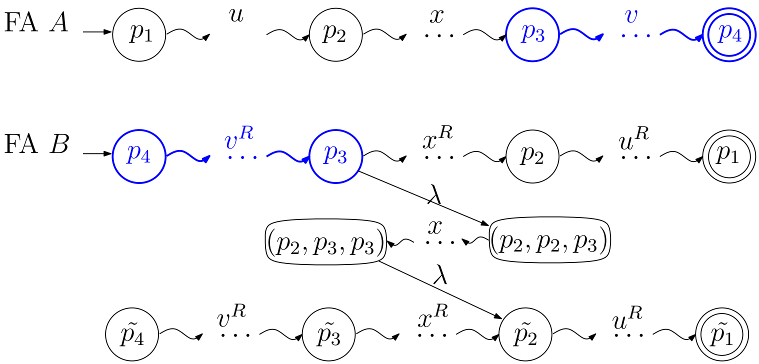


- $P = Q \cup \tilde{Q} \cup Q^3$
- $P_0 = \{p_4\}$
- $F_B = \{p_1, \tilde{p_1}\}$

Closure Property of $\mathbb{PI}(L)$

Theorem

If L is a regular language, then $\mathbb{PI}(L)$ is also regular.

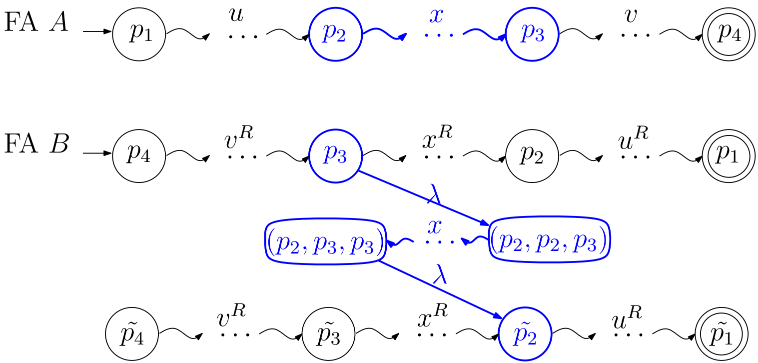


For all $q, p \in Q, a \in \Sigma$, if $p \in \delta(q, a)$, then $q \in \gamma(p, a)$.

Closure Property of $\mathbb{PI}(L)$

Theorem

If L is a regular language, then $\mathbb{PI}(L)$ is also regular.

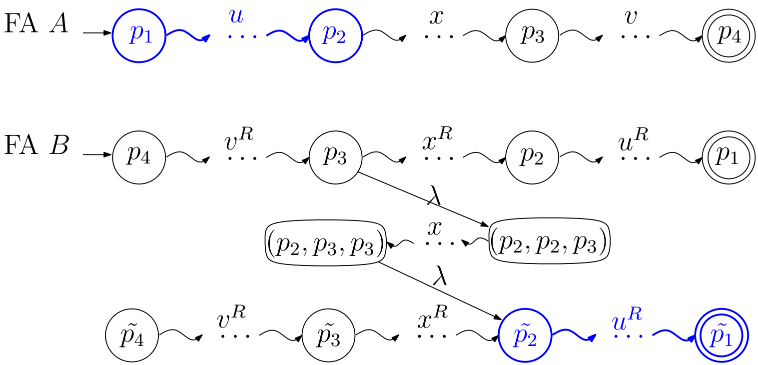


- For all $q, p \in Q$, $(q, q, p) \in \gamma(p, \lambda)$.
- For all $q, p, r_1, r_2 \in Q$, $a \in \Sigma$, if $r_2 \in \delta(r_1, a)$, then $(q, r_2, p) \in \gamma((q, r_1, p), a)$.
- For all $q, p \in Q$, $\tilde{q} \in \gamma((q, p, p), \lambda)$.

Closure Property of $\mathbb{PI}(L)$

Theorem

If L is a regular language, then $\mathbb{PI}(L)$ is also regular.



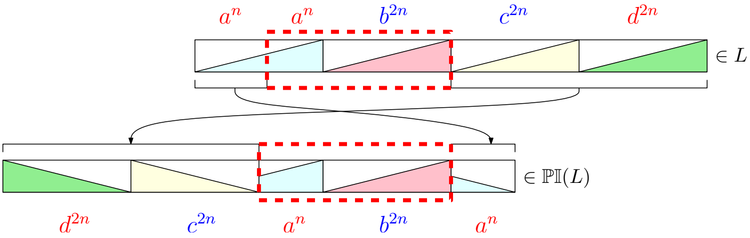
For all $q, p \in Q, a \in \Sigma$, if $p \in \delta(q, a)$, then $\tilde{q} \in \gamma(\tilde{p}, a)$.

Closure Property of $\mathbb{PI}(L)$

Theorem

Context-free languages are not closed under the pseudo-inversion.

- Given a context-free language $L = \{a^i b^j c^j d^i \mid i, j \geq 1\}$,



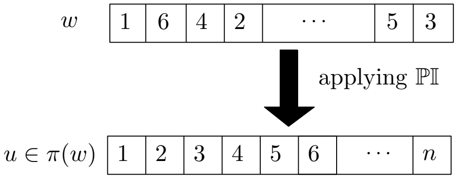
- we pick a string $w = d^{2n}c^{2n}a^n b^{2n}a^n \in \mathbb{PI}(L)$.

Property of Iterated Pseudo-Inversion

Given a string w, let $\pi(w)$ be the set of all *permutations* of w.

Theorem

Given a string w over Σ, $\mathbb{PI}^*(w) = \pi(w)$.

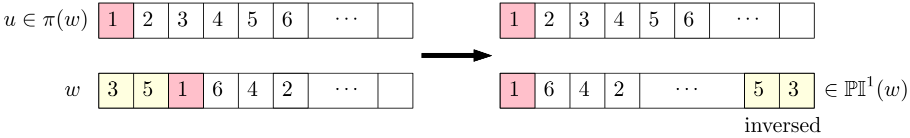


Property of Iterated Pseudo-Inversion

Given a string *w*, let $\pi(w)$ be the set of all *permutations* of *w*.

Theorem

Given a string w over Σ, $\mathbb{PI}^*(w) = \pi(w)$.

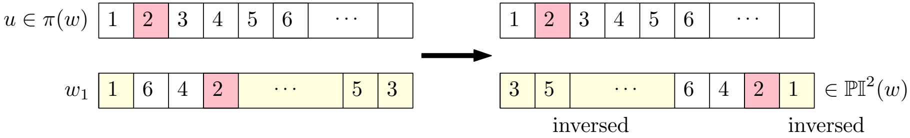


Property of Iterated Pseudo-Inversion

Given a string w, let $\pi(w)$ be the set of all *permutations* of w.

Theorem

Given a string w over Σ, $\mathbb{PI}^*(w) = \pi(w)$.

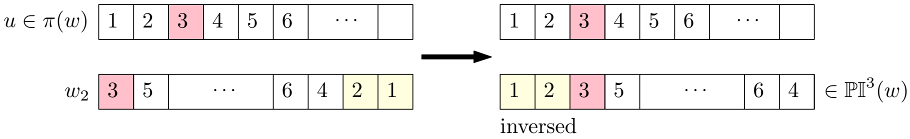


Property of Iterated Pseudo-Inversion

Given a string w, let $\pi(w)$ be the set of all *permutations* of w.

Theorem

Given a string w over Σ, $\mathbb{PI}^*(w) = \pi(w)$.

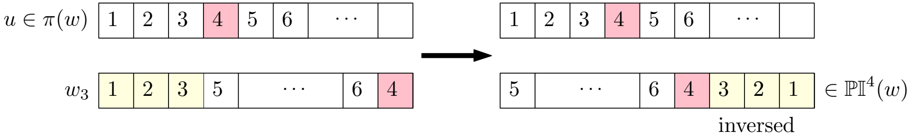


Property of Iterated Pseudo-Inversion

Given a string w, let $\pi(w)$ be the set of all *permutations* of w.

Theorem

Given a string w over Σ, $\mathbb{PI}^*(w) = \pi(w)$.

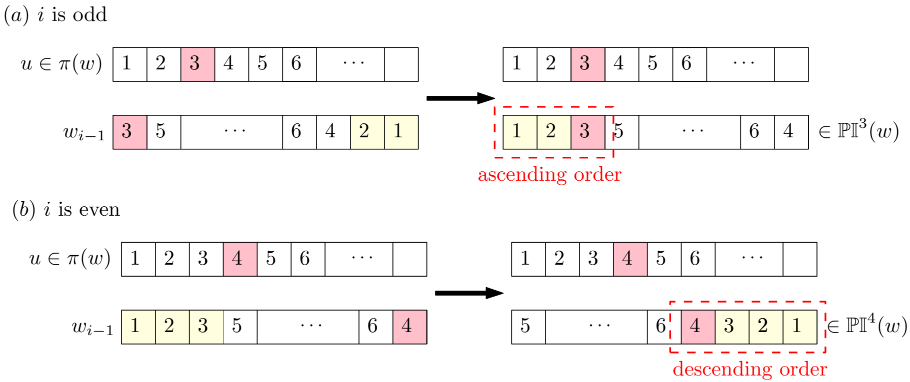


Property of Iterated Pseudo-Inversion

Given a string w, let $\pi(w)$ be the set of all *permutations* of w.

Theorem

Given a string w over Σ, $\mathbb{PI}^*(w) = \pi(w)$.



Closure of Iterated Pseudo-Inversion

Lemma

Regular languages and context-free languages are not closed under the iterated pseudo-inversion.

- Let $L = \{(abc)^*\}$. Then,

$$\mathbb{PI}^*(L) = \{w \in \{a, b, c\}^* \mid |w|_a = |w|_b = |w|_c\}.$$

- $\mathbb{PI}^*(L) \cap a^*b^*c^* = \{a^i b^i c^i \mid i \geq 0\}$ is not a CFL.

Note

Let L be context-free languages and R be regular languages. Then,

$$L \cap R = L.$$

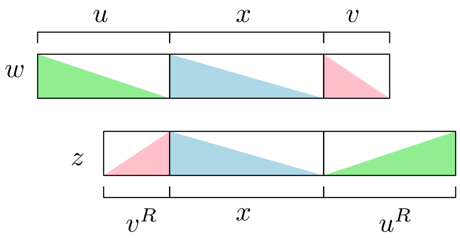
Given two strings w and z, is $z \in \mathbb{PI}(w)$?

Definition

For a string $w = uxv \in \Sigma^*, \mathbb{PI}(w) = \{v^R x u^R \mid u, x, v \in \Sigma^* \text{ and } vu \neq \lambda\}$.

Theorem

Given two strings w and z of length n, we can determine whether or not $z \in \mathbb{PI}(w)$ in $O(n)$ time.



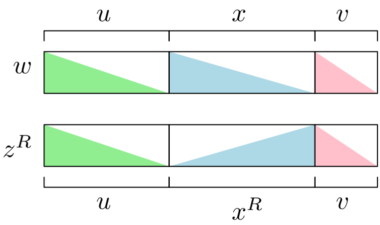
Given two strings w and z, is $z \in \mathbb{PI}(w)$?

Definition

For a string $w = uxv \in \Sigma^*, \mathbb{PI}(w) = \{v^R x u^R \mid u, x, v \in \Sigma^* \text{ and } vu \neq \lambda\}$.

Theorem

Given two strings w and z of length n, we can determine whether or not $z \in \mathbb{PI}(w)$ in $O(n)$ time.



Given two strings w and z, is $z \in \mathbb{PI}(w)$?

Definition

For a string $w = uxv \in \Sigma^*$, $\mathbb{PI}(w) = \{v^R x u^R \mid u, x, v \in \Sigma^* \text{ and } vu \neq \lambda\}$.

Theorem

Given two strings w and z of length n, we can determine whether or not $z \in \mathbb{PI}(w)$ in $O(n)$ time.

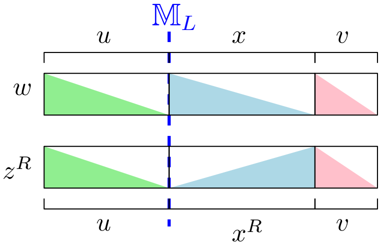


Figure: Let $\mathbb{M}_L$ denote the left maximum matching index and $\mathbb{M}_R$ denote the right maximum matching index.

Given two strings w and z, is $z \in \mathbb{PI}(w)$?

Definition

For a string $w = uxv \in \Sigma^*, \mathbb{PI}(w) = \{v^R x u^R \mid u, x, v \in \Sigma^* \text{ and } vu \neq \lambda\}$.

Theorem

Given two strings w and z of length n, we can determine whether or not $z \in \mathbb{PI}(w)$ in $O(n)$ time.

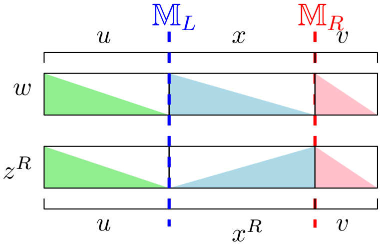


Figure: Let $\mathbb{M}_L$ denote the left maximum matching index and $\mathbb{M}_R$ denote the right maximum matching index.

Pseudo-Inversion-Freeness for Regular Language

Definition

A given language $L \in \Sigma^*$ is *pseudo-inversion-free* if no string in L is a pseudo-inversion substring of any other string in L.

For example, $L = \{canada, dnaa\}$ is not pseudo-inversion-free.

Theorem

Given an FA of size n recognizing a regular language L, we can determine whether or not L is pseudo-inversion-free in $O(n^4)$.

- L is pseudo-inversion-free if

$$\Sigma^* \cdot \mathbb{PI}(L) \cdot \Sigma^* \cap L = \emptyset.$$

Pseudo-Inversion-Freeness for CFL

Theorem

It is undecidable to determine whether or not a given context-free language L is pseudo-inversion-free.

- Code (Prefix, suffix, infix, *k*-intercodes) decision problems for context-free language are undecidable [Jürgensen and Konstantinidis, 1997].
- Post Correspondence Problem
  - is an undecidable problem [Post, 1946]
  - can be used to prove undecidability of many problem
- We reduced PCP to this problem.

Summary

- $\mathbb{PI}(w) = \{v^R x u^R \mid w = uvv, u, x, v \in \Sigma^* \text{ and } vu \neq \lambda\}$
- Closure properties of PI
  - Regular languages: closed
  - Context-free languages: not closed
- Properties of iterated PI
  - Iterated PI is equivalent to the permutation
  - Regular languages and context-free languages are not closed under iterated PI
- Decision problems of PI-freeness
  - Regular languages: decidable
  - Context-free languages: undecidable

Thank you!