Transducer Descriptions of DNA Code Properties and Undecidability of Antimorphic Problems

Lila Kari[1] Stavros Konstantinidis[2] Steffen Kopecki[1,2]

Western University · Canada

Saint Mary's University Since 1802
One University. One World Yours.

Descriptional Complexity of Formal Systems 2015
Waterloo, Ontario, Canada — June 25–27

Properties of DNA Libraries

- *DNA alphabet* $\Delta = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$
- *Watson-Crick complement* $\delta(\mathtt{A}) = \mathtt{T}$, $\delta(\mathtt{T}) = \mathtt{A}$, $\delta(\mathtt{C}) = \mathtt{G}$, $\delta(\mathtt{G}) = \mathtt{C}$
- δ is *antimorphic*: $\delta(uv) = \delta(v)\delta(u)$
- *DNA library* $L \subseteq \Delta^+$ can satisfy all kinds of properties

(a)

GTACCTTCAC →
← CATGGAAGTG

(b)

AGTCCAGAAGTTCCTGAATCC →
← TCTTCAAGGA

Properties of DNA Libraries

- *DNA alphabet* $\Delta = \{\mathtt{A}, \mathtt{C}, \mathtt{G}, \mathtt{T}\}$
- *Watson-Crick complement* $\delta(\mathtt{A}) = \mathtt{T}$, $\delta(\mathtt{T}) = \mathtt{A}$, $\delta(\mathtt{C}) = \mathtt{G}$, $\delta(\mathtt{G}) = \mathtt{C}$
- δ is *antimorphic*: $\delta(uv) = \delta(v)\delta(u)$
- *DNA library* $L \subseteq \Delta^+$ can satisfy all kinds of properties

(a) GTACCTTCAC →
← CATGGAAGTG

(b) AGTCCAGAAGTTCCTGAATCC →
← TCTTCAAGGA

Example properties

- L is *δ-nonoverlapping* if $L \cap \delta(L) = \emptyset$.
  (a) forbidden, (b) allowed
- L is *δ-compliant* if $\forall w \in \delta(L), x, y \in A^*: xwy \in L \implies xy = \varepsilon$.
  (a) allowed, (b) forbidden
- L is *strictly δ-compliant* if it is δ-nonoverlapping and δ-compliant.
  (a) and (b) forbidden

Formal Models

Define a formal model that allows us to (efficiently) solve (some of) the following problems:

Satisfaction given a property P and a (regular) library L:
decide if L satisfies P.

Maximality given a property P and a (regular) library L with $L \in P$:
decide if L is maximal with respect to P.

Construction given a property P and $n \in \mathbb{N}$:
construct a library L such that $|L| \geq n$ and $L \in P$.

Formal Models

Define a formal model that allows us to (efficiently) solve (some of) the following problems:

Satisfaction given a property P and a (regular) library L:
decide if L satisfies P.

Maximality given a property P and a (regular) library L with $L \in P$:
decide if L is maximal with respect to P.

Construction given a property P and $n \in \mathbb{N}$:
construct a library L such that $|L| \geq n$ and $L \in P$.

Two existing models

- transducer code properties (not suitable for DNA properties)
- trajectory DNA code properties

Formal Models

Define a formal model that allows us to (efficiently) solve (some of) the following problems:

Satisfaction given a property P and a (regular) library L:
decide if L satisfies P.

Maximality given a property P and a (regular) library L with $L \in P$:
decide if L is maximal with respect to P.

Construction given a property P and $n \in \mathbb{N}$:
construct a library L such that $|L| \geq n$ and $L \in P$.

Two existing models

- transducer code properties (not suitable for DNA properties)
- trajectory DNA code properties

Generalization

- alphabet A with at least two letters
- (anti-)morphic permutation θ

Classic Transducer Properties

- A *transducer* $\mathbf{t}\colon A^* \to A^*$ is a finite automaton with output.
- $\mathbf{t}$ is *input-altering* if $w \notin \mathbf{t}(w)$ for all $w \in A^+$.
- $\mathbf{t}$ is *input-preserving* if $w \in \mathbf{t}(w)$ for all $w \in A^+$.

Definition (Dudzinski, Konstantinidis, 2012)

1.) L has the *input-altering property* described by i.-a. $\mathbf{t}$ if

$$L \cap \mathbf{t}(L) = \emptyset.$$

2.) L has the *input-preserving property* described by i.-p. $\mathbf{t}$ if

$$\forall w \in L\colon\ w \notin \mathbf{t}(L \setminus w).$$

Example

- $\mathbf{t}_{\mathrm{pr}}$ is i.-a. and describes the prefix codes.
- $\mathbf{t}_{1\mathrm{s}}$ is i.-p. and describes the 1-substitution error-detection.

θ-Transducer Properties

- Let $\theta\colon A^* \to A^*$ be an (anti-)morphic permutation.
- $\mathbf{t}$ is *θ-input-altering* if $\theta(w) \notin \mathbf{t}(w)$ for all $w \in A^+$.
- $\mathbf{t}$ is *θ-input-preserving* if $\theta(w) \in \mathbf{t}(w)$ for all $w \in A^+$.

Definition

1.) L satisfies the *strict θ-transducer property ($\mathcal{S}$-property)* $\mathcal{S}_{\theta,\mathbf{t}}$ if

$$\theta(L) \cap \mathbf{t}(L) = \emptyset.$$

2.) L satisfies the *weak θ-transducer property ($\mathcal{W}$-property)* $\mathcal{W}_{\theta,\mathbf{t}}$ if

$$\forall w \in L\colon\ \theta(w) \notin \mathbf{t}(L \setminus w).$$

Remark

Let $\theta = \mathrm{id}$ be the (morphic) identity.

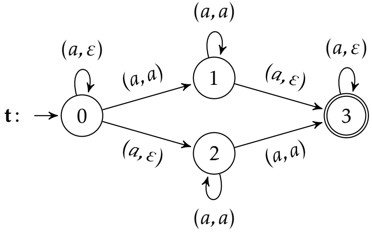
1.) If $\mathbf{t}$ is id-i.-a., then $\mathcal{S}_{\mathrm{id},\mathbf{t}}$ is the i.-a. transducer property.

2.) If $\mathbf{t}$ is id-i.-p., then $\mathcal{W}_{\mathrm{id},\mathbf{t}}$ is the i.-p. transducer property.

Describing DNA Code Properties

- L is *θ-compliant* if $\forall w \in \theta(L), x, y \in A^*\colon xwy \in L \implies xy = \varepsilon$.

Example

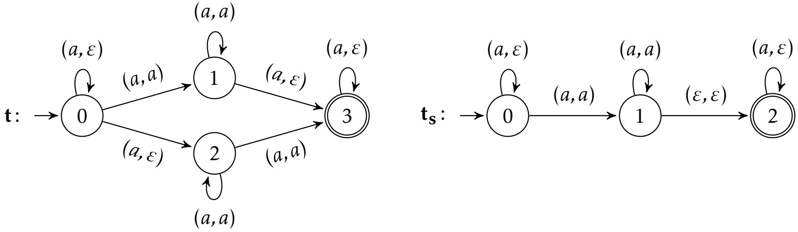


- $\mathbf{t}$ is θ-i.-a. and $\mathcal{S}_{\theta,\mathbf{t}} = \mathcal{W}_{\theta,\mathbf{t}}$ is the θ-compliant property:
  $\theta(L) \cap \mathbf{t}(L) = \emptyset \iff \forall w \in L\colon \theta(w) \notin \mathbf{t}(L \setminus w) \iff$ L is θ-compliant

Describing DNA Code Properties

- L is *θ-compliant* if $\forall w \in \theta(L), x, y \in A^*\colon xwy \in L \implies xy = \varepsilon$.
- L is *strictly θ-compliant* if $\forall w \in \theta(L), x, y \in A^*\colon xwy \notin L$.

Example



- $\mathbf{t}$ is θ-i.-a. and $\mathcal{S}_{\theta,\mathbf{t}} = \mathcal{W}_{\theta,\mathbf{t}}$ is the θ-compliant property:

  $\theta(L) \cap \mathbf{t}(L) = \emptyset \iff \forall w \in L\colon \theta(w) \notin \mathbf{t}(L \setminus w) \iff$ L is θ-compliant

- $\mathcal{S}_{\theta,\mathbf{t_s}}$ is the strictly θ-compliant property:

  $\theta(L) \cap \mathbf{t_s}(L) = \emptyset \iff$ L is strictly θ-compliant

First Observations

Proposition

The θ-nonoverlapping property is not describable by any input-preserving transducer.

- L is *θ-nonoverlapping* if $L \cap \theta(L) = \emptyset$.

Remark

θ-nonoverlapping is realized by $\mathbf{t}_{\mathrm{id}}$:

$$L \in \mathcal{S}_{\theta,\mathbf{t}_{\mathrm{id}}} \iff \theta(L) \cap \mathbf{t}_{\mathrm{id}}(L) = \emptyset.$$

First Observations

Proposition

The θ-nonoverlapping property is not describable by any input-preserving transducer.

- L is *θ-nonoverlapping* if $L \cap \theta(L) = \emptyset$.

Remark

θ-nonoverlapping is realized by $\mathbf{t}_{\mathrm{id}}$:

$$L \in \mathcal{S}_{\theta,\mathbf{t}_{\mathrm{id}}} \iff \theta(L) \cap \mathbf{t}_{\mathrm{id}}(L) = \emptyset.$$

Proposition

The θ-free property is not a θ-transducer property.

- L is *θ-free* if and only if $L^2 \cap A^+\theta(L)A^+ = \emptyset$.

Example

$K = \{\mathtt{ACGT}, \mathtt{CCAC}, \mathtt{GTAA}\}$ is not δ-free as $\mathtt{ACGT} = \delta(\mathtt{ACGT})$ and $\mathtt{CCACGTAA} \in \Delta^+\mathtt{ACGT}\Delta^+$.

DNA Code Properties via Trajectories

Let $\bar{e}_1, \bar{e}_2 \subseteq \{0,1\}^*$ be (regular) trajectories.

$$\Phi^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}(L) = ((L \rightsquigarrow_{\bar{e}_1} A^*) \cap A^+) \sqcup\!\sqcup_{\bar{e}_2} A^*$$

$\rightsquigarrow$ scattered deletion
$\sqcup\!\sqcup$ shuffle (scattered insertion)

Let $\bar{e}_1, \bar{e}_2 \subseteq \{0,1\}^*$ be (regular) trajectories.

$$\Phi^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}(L) = ((L \rightsquigarrow_{\bar{e}_1} A^*) \cap A^+) \sqcup\!\sqcup_{\bar{e}_2} A^*$$

$\rightsquigarrow$ scattered deletion

$\sqcup\!\sqcup$ shuffle (scattered insertion)

Definition (Domaratzki, 2007)

The *strictly bond-free property described by* $(\bar{e}_1, \bar{e}_2)$ is

$$\mathcal{B}^{\mathbf{s}}_{\theta}(\bar{e}_1,\bar{e}_2) = \left\{ L \subseteq A^* \;\middle|\; \theta(L) \cap \Phi^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}(L) = \emptyset \right\}.$$

N. b.: there is also a *bond-free property*.

Example

$\mathcal{B}^{\mathbf{s}}_{\theta}(1^*0^+1^*, 0^+)$ is the strictly θ-compliant property.

θ-Trajectory vs. θ-Transducer Properties

Theorem

Every regular θ-trajectory property is a θ-transducer $\mathcal{S}$-property.

Idea: there are transducers that implement all necessary operations

$$\mathbf{t}_{\rightsquigarrow_{\bar{e}_1}}(X) = X \rightsquigarrow_{\bar{e}_1} A^*, \qquad \mathbf{t}_{\sqcup\!\sqcup_{\bar{e}_2}}(X) = X \sqcup\!\sqcup_{\bar{e}_2} A^*, \qquad \mathbf{t}_+(X) = X \cap A^+$$

$$\Longrightarrow \mathbf{t}^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2} := \mathbf{t}_{\sqcup\!\sqcup_{\bar{e}_2}} \circ \mathbf{t}_+ \circ \mathbf{t}_{\rightsquigarrow_{\bar{e}_1}} = \Phi^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}$$

$$\mathcal{S}_{\theta,\mathbf{t}^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}} = \mathcal{B}^{\mathbf{s}}_{\theta}(\bar{e}_1,\bar{e}_2) = \left\{ L \subseteq A^* \;\middle|\; \theta(L) \cap \Phi^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}(L) = \emptyset \right\}$$

θ-Trajectory vs. θ-Transducer Properties

Theorem

Every regular θ-trajectory property is a θ-transducer $\mathcal{S}$-property.

Idea: there are transducers that implement all necessary operations

$$\mathbf{t}_{\rightsquigarrow_{\bar{e}_1}}(X) = X \rightsquigarrow_{\bar{e}_1} A^*, \qquad \mathbf{t}_{\sqcup\!\sqcup_{\bar{e}_2}}(X) = X \sqcup\!\sqcup_{\bar{e}_2} A^*, \qquad \mathbf{t}_+(X) = X \cap A^+$$

$$\Longrightarrow \mathbf{t}^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2} := \mathbf{t}_{\sqcup\!\sqcup_{\bar{e}_2}} \circ \mathbf{t}_+ \circ \mathbf{t}_{\rightsquigarrow_{\bar{e}_1}} = \Phi^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}$$

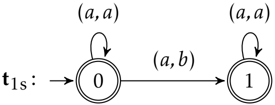
$$\mathcal{S}_{\theta,\mathbf{t}^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}} = \mathcal{B}^{\mathbf{s}}_{\theta}(\bar{e}_1,\bar{e}_2) = \left\{L \subseteq A^* \;\middle|\; \theta(L) \cap \Phi^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}(L) = \emptyset\right\}$$

Theorem

$\mathcal{H}$ is a θ-transducer property, but not a θ-trajectory one.

- $\mathcal{H} = \{L \subseteq A^* \mid \forall u,v \in L\colon \mathrm{Hamming}(u,\theta(v)) \geq 2\}$

Idea: $\mathcal{S}_{\theta,\mathbf{t}_{1s}}$ is the desired property.



A θ-trajectory property would require "del i"- and "ins i"-trajectories. But, "del i" could combine with "ins j".

(Un-)decidability of Satisfaction and Maximality

Given: NFA $\mathbf{a}$, transducer $\mathbf{t}$, antimorphic permutation θ

Problem	Property $\mathcal{S}_{\theta,\mathbf{t}}$		Property $\mathcal{W}_{\theta,\mathbf{t}}$	
	general	θ-i.-altering	general	θ-i.-preserving
Satisfaction	decidable* in $\mathcal{O}(\lvert\mathbf{t}\rvert\lvert\mathbf{a}\rvert^2)$		decidable†	decidable* in $\mathcal{O}(\lvert\mathbf{t}\rvert^2\lvert\mathbf{a}\rvert^4)$
Maximality	undecidable†	decidable†, PSPACE-hard		

* as in Dudzinski, Konstantinidis, 2012

† proven here

Given: NFA $\mathbf{a}$, transducer $\mathbf{t}$, antimorphic permutation θ

Problem	Property $\mathcal{S}_{\theta,\mathbf{t}}$ general	Property $\mathcal{S}_{\theta,\mathbf{t}}$ θ-i.-altering	Property $\mathcal{W}_{\theta,\mathbf{t}}$ general	Property $\mathcal{W}_{\theta,\mathbf{t}}$ θ-i.-preserving
Satisfaction	decidable* in $\mathcal{O}(\|\mathbf{t}\|\|\mathbf{a}\|^2)$		decidable†	decidable* in $\mathcal{O}(\|\mathbf{t}\|^2\|\mathbf{a}\|^4)$
Maximality	undecidable†	decidable†, PSPACE-hard		

* as in Dudzinski, Konstantinidis, 2012

† proven here

Remark

Transforming a trajectory property into an $\mathcal{S}$-properties and deciding satisfaction is asymptotically as fast as deciding the trajectory property directly $\mathcal{O}(|\bar{e}_1||\bar{e}_2||\mathbf{a}|^2) = \mathcal{O}(|\mathbf{t}^{\mathbf{s}}_{\bar{e}_1,\bar{e}_2}||\mathbf{a}|^2)$.

Deciding if L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$

Theorem

It is decidable if a regular L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$.

Deciding if L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$

Theorem

It is decidable if a regular L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$.

Lemma

L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$ iff $(x,y) \in \mathbf{s} \implies \theta(x) = y$.

- $\mathbf{s} = \mathbf{t} \cap (L \times \theta(L)) \subseteq A^* \times A^*$

Deciding if L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$

Theorem

It is decidable if a regular L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$.

Lemma

L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$ iff $(x,y) \in \mathbf{s} \implies \theta(x) = y$.

- $\mathbf{s} = \mathbf{t} \cap (L \times \theta(L)) \subseteq A^* \times A^*$

Lemma

L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$ iff

$$\mathbf{s} \subseteq \bigcup_{\substack{x_1,x_2,x_3 \in A^* \\ |x_1 x_2 x_3| \leq |\mathbf{s}|}} \bigcup_{i \in \mathbb{N}} \left(x_1 (x_2)^i x_3, \theta(x_1 (x_2)^i x_3) \right).$$

Deciding if L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$

Theorem

It is decidable if a regular L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$.

Lemma

L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$ iff $(x,y) \in \mathbf{s} \implies \theta(x) = y$.

- $\mathbf{s} = \mathbf{t} \cap (L \times \theta(L)) \subseteq A^* \times A^*$

Lemma

L satisfies $\mathcal{W}_{\theta,\mathbf{t}}$ iff

$$\mathbf{s} \subseteq \bigcup_{\substack{x_1,x_2,x_3 \in A^* \\ |x_1x_2x_3| \le |\mathbf{s}|}} \bigcup_{i \in \mathbb{N}} \Big(x_1(x_2)^i x_3, \theta(x_1(x_2)^i x_3)\Big).$$

This can be tested by verifying that

1.) $\mathbf{s} \subseteq \bigcup_{\substack{x_1,x_2,x_3 \in A^* \\ |x_1x_2x_3| \le |\mathbf{s}|}} (x_1 x_2^* x_3) \times \theta(x_1 x_2^* x_3)$; and

2.) $|x| = |y|$ for all $(x,y) \in \mathbf{s}$.

Deciding Maximality

Let $L \in \mathcal{W}_{\theta,\mathbf{t}}$, or $L \in \mathcal{S}_{\theta,\mathbf{t}}$ and $\mathbf{t}$ is θ-input-altering.

Theorem

L is maximal with respect to $\mathcal{W}_{\theta,\mathbf{t}}$ (resp., $\mathcal{S}_{\theta,\mathbf{t}}$) iff

$$L \cup \theta^{-1}(\mathbf{t}(L)) \cup \mathbf{t}^{-1}(\theta(L)) = A^*.$$

Deciding Maximality

Let $L \in \mathcal{W}_{\theta,\mathbf{t}}$, or $L \in \mathcal{S}_{\theta,\mathbf{t}}$ and $\mathbf{t}$ is θ-input-altering.

Theorem

L is maximal with respect to $\mathcal{W}_{\theta,\mathbf{t}}$ (resp., $\mathcal{S}_{\theta,\mathbf{t}}$) iff

$$L \cup \theta^{-1}(\mathbf{t}(L)) \cup \mathbf{t}^{-1}(\theta(L)) = A^*.$$

Corollary

It is PSPACE-hard to decide if L is maximal w. r. t. $\mathcal{W}_{\theta,\mathbf{t}}$ (resp., $\mathcal{S}_{\theta,\mathbf{t}}$).

…because deciding universality of an NFA is PSPACE-hard.

Deciding Maximality

Let $L \in \mathcal{W}_{\theta,\mathbf{t}}$, or $L \in \mathcal{S}_{\theta,\mathbf{t}}$ and $\mathbf{t}$ is θ-input-altering.

Theorem

L is maximal with respect to $\mathcal{W}_{\theta,\mathbf{t}}$ (resp., $\mathcal{S}_{\theta,\mathbf{t}}$) iff

$$L \cup \theta^{-1}(\mathbf{t}(L)) \cup \mathbf{t}^{-1}(\theta(L)) = A^*.$$

Corollary

It is PSPACE-hard to decide if L is maximal w. r. t. $\mathcal{W}_{\theta,\mathbf{t}}$ (resp., $\mathcal{S}_{\theta,\mathbf{t}}$).

...because deciding universality of an NFA is PSPACE-hard.

Note

Let $L \in \mathcal{S}_{\theta,\mathbf{t}}$ and $\mathbf{t}$ is not θ-input-altering. If

$$w \notin L \cup \theta^{-1}(\mathbf{t}(L)) \cup \mathbf{t}^{-1}(\theta(L)),$$

then L may still be maximal provided $\theta(w) \in \mathbf{t}(w)$.

Undecidability of Maximality w. r. t. $\mathcal{S}_{\theta,\mathbf{t}}$

Post correspondence problem (PCP)

Given words $\alpha_1, \alpha_2, \ldots, \alpha_\ell \in A^+$ and $\beta_1, \beta_2, \ldots, \beta_\ell \in A^+$, decide if there is a non-empty sequence $i_1, i_2, \ldots, i_n$ such that

$$\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_n} = \beta_{i_1}\beta_{i_2}\cdots\beta_{i_n}.$$

Theorem

It is undecidable if a given transducer is θ-input-preserving.

Idea: define a transducer $\mathbf{t}$ such that

$$\theta(w) \notin \mathbf{t}(w) \iff w = \mathrm{bin}(u \cdot i_n \cdots i_1) \text{ with } u = \alpha_{i_1}\cdots\alpha_{i_n} = \beta_{i_1}\cdots\beta_{i_n}.$$

Undecidability of Maximality w. r. t. $\mathcal{S}_{\theta,\mathbf{t}}$

Post correspondence problem (PCP)

Given words $\alpha_1, \alpha_2, \ldots, \alpha_\ell \in A^+$ and $\beta_1, \beta_2, \ldots, \beta_\ell \in A^+$, decide if there is a non-empty sequence $i_1, i_2, \ldots, i_n$ such that

$$\alpha_{i_1} \alpha_{i_2} \cdots \alpha_{i_n} = \beta_{i_1} \beta_{i_2} \cdots \beta_{i_n}.$$

Theorem

It is undecidable if a given transducer is θ-input-preserving.

Idea: define a transducer $\mathbf{t}$ such that

$$\theta(w) \notin \mathbf{t}(w) \iff w = \mathrm{bin}(u \cdot i_n \cdots i_1) \text{ with } u = \alpha_{i_1} \cdots \alpha_{i_n} = \beta_{i_1} \cdots \beta_{i_n}.$$

Corollary

It is undecidable if $\emptyset$ is maximal with respect to $\mathcal{S}_{\theta,\mathbf{t}}$.

Undecidability of the θ-PCP

θ-Post correspondence problem (θ-PCP)

Given words $\alpha_1, \alpha_2, \ldots, \alpha_\ell \in A^+$ and $\beta_1, \beta_2, \ldots, \beta_\ell \in A^+$, decide if there is a non-empty sequence $i_1, i_2, \ldots, i_n$ such that

$$\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_n} = \theta(\beta_{i_1}\beta_{i_2}\cdots\beta_{i_n}).$$

Theorem

The θ-PCP is undecidable.

Undecidability of the θ-PCP

θ-Post correspondence problem (θ-PCP)

Given words $\alpha_1, \alpha_2, \ldots, \alpha_\ell \in A^+$ and $\beta_1, \beta_2, \ldots, \beta_\ell \in A^+$, decide if there is a non-empty sequence $i_1, i_2, \ldots, i_n$ such that

$$\alpha_{i_1} \alpha_{i_2} \cdots \alpha_{i_n} = \theta(\beta_{i_1} \beta_{i_2} \cdots \beta_{i_n}).$$

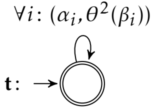
Theorem

The θ-PCP is undecidable.

Corollary

It is undecidable if a given transducer is θ-input-altering.

Idea:



$$w \in \theta^{-1}(\mathbf{t}(w)) \iff \alpha_{i_1} \cdots \alpha_{i_n} = w = \theta^{-1}\left(\theta^2(\beta_{i_1} \cdots \beta_{i_n})\right) = \theta(\beta_{i_1} \cdots \beta_{i_n})$$

Questions?