Permutation Pattern Avoidance and Constraint Satisfaction Problems

AAA90 : Novi Sad, Serbia

Tom Hanika

Universität Kassel

June 6, 2015

The Forbidden Permutation Constraint Satisfaction Problem

For a given set of permutations P.

The Forbidden Permutation Constraint Satisfaction Problem

For a given set of permutations P.

Problem (FPCSP(P))

The Forbidden Permutation Constraint Satisfaction Problem

For a given set of permutations P.

Problem (FPCSP(P))

Input: *A tuple (X, E_1, E_2) where E_1, E_2 are binary relations on the finite set X.*

The Forbidden Permutation Constraint Satisfaction Problem

For a given set of permutations P.

Problem (FPCSP(P))

Input: *A tuple (X, E_1, E_2) where E_1, E_2 are binary relations on the finite set X.*

Question: *Are there linear orders $L_1 \supseteq E_1$, $L_2 \supseteq E_2$ such that for any $p \in P$ the permutation represented by L_1 and L_2 does not involve p?*

The Forbidden Permutation Constraint Satisfaction Problem

For a given set of permutations P.

Problem (FPCSP(P))

Input: *A tuple (X, E_1, E_2) where E_1, E_2 are binary relations on the finite set X.*

Question: *Are there linear orders $L_1 \supseteq E_1$, $L_2 \supseteq E_2$ such that for any $p \in P$ the permutation represented by L_1 and L_2 does not involve p?*

The Forbidden Permutation Constraint Satisfaction Problem

For a given set of permutations P.

Problem (FPCSP(P))

Input: *A tuple* (X, E_1, E_2) *where* E_1, E_2 *are binary relations on the finite set* X.

Question: *Are there linear orders* $L_1 \supseteq E_1$, $L_2 \supseteq E_2$ *such that for any* $p \in P$ *the permutation represented by* L_1 *and* L_2 *does not involve* p?

The Forbidden Permutation Constraint Satisfaction Problem

For a given set of permutations P.

Problem (FPCSP(P))

Input: *A tuple (X, E_1, E_2) where E_1, E_2 are binary relations on the finite set X.*

Question: *Are there linear orders $L_1 \supseteq E_1$, $L_2 \supseteq E_2$ such that for any $p \in P$ the permutation represented by L_1 and L_2 does not involve p?*

Goal for this talk

The Forbidden Permutation Constraint Satisfaction Problem

For a given set of permutations P.

Problem (FPCSP(P))

Input: *A tuple (X, E_1, E_2) where E_1, E_2 are binary relations on the finite set X.*

Question: *Are there linear orders $L_1 \supseteq E_1$, $L_2 \supseteq E_2$ such that for any $p \in P$ the permutation represented by L_1 and L_2 does not involve p?*

Goal for this talk

- Structural properties leading to an algorithm for $P = \{231\}$.

Permutations etc.

What is a permutation on X ?

Common definition (permutation)

A permutation on X is a bijection $\pi : X \to X$.

What is a permutation on X ?

Common definition (permutation)

A permutation on X is a bijection $\pi : X \to X$.

Definition (Permutation)

A permutation on a finite set X is a pair of linear orders on X. (X, L_1, L_2).

What is a permutation on X ?

Common definition (permutation)

A permutation on X is a bijection $\pi : X \to X$.

Definition (Permutation)

A permutation on a finite set X is a pair of linear orders on X. (X, L_1, L_2).

Example ($X = \{1, 2, 3, 4, 5\}$)

What is a permutation on X ?

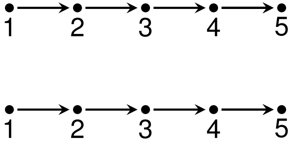
Common definition (permutation)

A permutation on X is a bijection $\pi : X \to X$.

Definition (Permutation)

A permutation on a finite set X is a pair of linear orders on X. (X, L_1, L_2).

Example ($X = \{1, 2, 3, 4, 5\}$)



$\pi = 12345$

What is a permutation on X ?

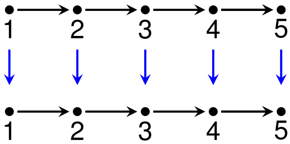
Common definition (permutation)

A permutation on X is a bijection $\pi : X \to X$.

Definition (Permutation)

A permutation on a finite set X is a pair of linear orders on X. (X, L_1, L_2).

Example ($X = \{1, 2, 3, 4, 5\}$)



$\pi = 12345$

What is a permutation on X ?

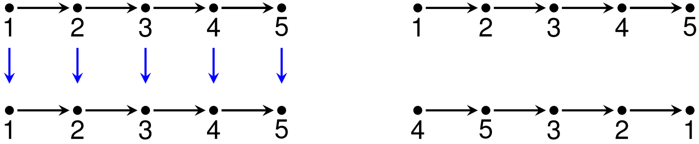
Common definition (permutation)

A permutation on X is a bijection $\pi : X \to X$.

Definition (Permutation)

A permutation on a finite set X is a pair of linear orders on X. (X, L_1, L_2).

Example ($X = \{1, 2, 3, 4, 5\}$)



$\pi = 12345$ $\pi = 45321$

What is a permutation on X ?

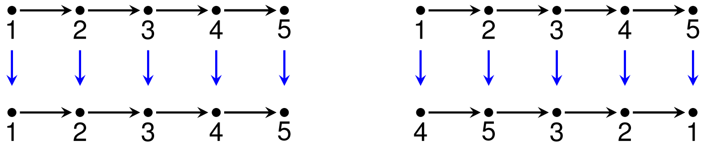
Common definition (permutation)

A permutation on X is a bijection $\pi : X \to X$.

Definition (Permutation)

A permutation on a finite set X is a pair of linear orders on X. (X, L_1, L_2).

Example ($X = \{1, 2, 3, 4, 5\}$)



$\pi = 12345$ $\pi = 45321$

What is a permutation on X ?

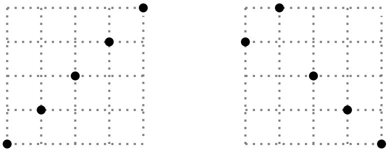
Common definition (permutation)

A permutation on X is a bijection $\pi : X \to X$.

Definition (Permutation)

A permutation on a finite set X is a pair of linear orders on X. (X, L_1, L_2).

Example ($X = \{1, 2, 3, 4, 5\}$)



$\pi = 12345$ $\pi = 45321$

What is a permutation on X ?

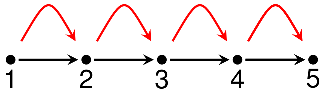
Common definition (permutation)

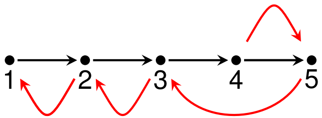
A permutation on X is a bijection $\pi : X \to X$.

Definition (Permutation)

A permutation on a finite set X is a pair of linear orders on X. (X, L_1, L_2).

Example ($X = \{1, 2, 3, 4, 5\}$)





$\pi = 12345$ $\pi = 45321$

One permutation to another

Definition (Sequences and order isomorphic)

$\sigma = s_1 s_2 s_3 \dots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_\mathbb{N})$.
Sequences $\sigma = s_1 s_2 \dots s_k$, $\tau = t_1 t_2 \dots t_k$ are order isomorphic ($\sigma \cong \tau$), if

$$\forall i, j \in \{1, \dots, k\} : s_i \leq_\mathbb{N} s_j \Leftrightarrow t_i \leq_\mathbb{N} t_j.$$

One permutation to another

Definition (Sequences and order isomorphic)

$\sigma = s_1 s_2 s_3 \dots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_{\mathbb{N}})$.
Sequences $\sigma = s_1 s_2 \dots s_k$, $\tau = t_1 t_2 \dots t_k$ are order isomorphic ($\sigma \cong \tau$), if

$$\forall i, j \in \{1, \dots, k\} : s_i \leq_{\mathbb{N}} s_j \Leftrightarrow t_i \leq_{\mathbb{N}} t_j.$$

Definition (Involvement)

Seq. σ is involved in ($\preceq$) seq. τ, if $\sigma \cong \tau'$ for some subsequence τ' of τ.

One permutation to another

Definition (Sequences and order isomorphic)

$\sigma = s_1 s_2 s_3 \dots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_{\mathbb{N}})$.
Sequences $\sigma = s_1 s_2 \dots s_k$, $\tau = t_1 t_2 \dots t_k$ are order isomorphic ($\sigma \cong \tau$), if

$$\forall i, j \in \{1, \dots, k\} : s_i \leq_{\mathbb{N}} s_j \Leftrightarrow t_i \leq_{\mathbb{N}} t_j.$$

Definition (Involvement)

Seq. σ is involved in ($\preceq$) seq. τ, if $\sigma \cong \tau'$ for some subsequence τ' of τ.

Example

Given $\pi = 14352$, then

- $\sigma = 132$ is involved in π, since 142 is a subsequence of π.

One permutation to another

Definition (Sequences and order isomorphic)

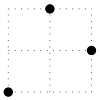
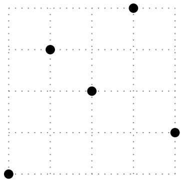
$\sigma = s_1 s_2 s_3 \dots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_{\mathbb{N}})$.
Sequences $\sigma = s_1 s_2 \dots s_k$, $\tau = t_1 t_2 \dots t_k$ are order isomorphic ($\sigma \cong \tau$), if

$$\forall i, j \in \{1, \dots, k\} : s_i \leq_{\mathbb{N}} s_j \Leftrightarrow t_i \leq_{\mathbb{N}} t_j.$$

Definition (Involvement)

Seq. σ is involved in ($\preceq$) seq. τ, if $\sigma \cong \tau'$ for some subsequence τ' of τ.

Example

132  $\preceq$  14352

One permutation to another

Definition (Sequences and order isomorphic)

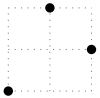
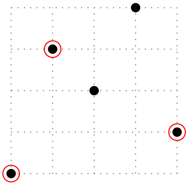
$\sigma = s_1 s_2 s_3 \dots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_{\mathbb{N}})$.
Sequences $\sigma = s_1 s_2 \dots s_k$, $\tau = t_1 t_2 \dots t_k$ are order isomorphic ($\sigma \cong \tau$), if

$$\forall i, j \in \{1, \dots, k\} : s_i \leq_{\mathbb{N}} s_j \Leftrightarrow t_i \leq_{\mathbb{N}} t_j.$$

Definition (Involvement)

Seq. σ is involved in ($\preceq$) seq. τ, if $\sigma \cong \tau'$ for some subsequence τ' of τ.

Example

132  $\preceq$  14352

One permutation to another

Definition (Sequences and order isomorphic)

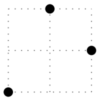
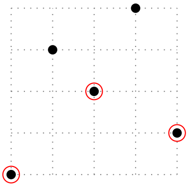
$\sigma = s_1 s_2 s_3 \dots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_{\mathbb{N}})$.
Sequences $\sigma = s_1 s_2 \dots s_k$, $\tau = t_1 t_2 \dots t_k$ are order isomorphic ($\sigma \cong \tau$), if

$$\forall i, j \in \{1, \dots, k\} : s_i \leq_{\mathbb{N}} s_j \Leftrightarrow t_i \leq_{\mathbb{N}} t_j.$$

Definition (Involvement)

Seq. σ is involved in ($\preceq$) seq. τ, if $\sigma \cong \tau'$ for some subsequence τ' of τ.

Example

132  $\preceq$  14352

One permutation to another

Definition (Sequences and order isomorphic)

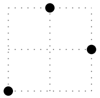
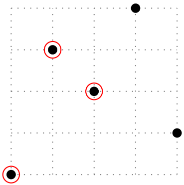
$\sigma = s_1 s_2 s_3 \dots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_{\mathbb{N}})$.
Sequences $\sigma = s_1 s_2 \dots s_k$, $\tau = t_1 t_2 \dots t_k$ are order isomorphic ($\sigma \cong \tau$), if

$$\forall i, j \in \{1, \dots, k\} : s_i \leq_{\mathbb{N}} s_j \Leftrightarrow t_i \leq_{\mathbb{N}} t_j.$$

Definition (Involvement)

Seq. σ is involved in ($\preceq$) seq. τ, if $\sigma \cong \tau'$ for some subsequence τ' of τ.

Example

132  $\preceq$  14352

One permutation to another

Definition (Sequences and order isomorphic)

$\sigma = s_1 s_2 s_3 \ldots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_{\mathbb{N}})$.
Sequences $\sigma = s_1 s_2 \ldots s_k$, $\tau = t_1 t_2 \ldots t_k$ are order isomorphic ($\sigma \cong \tau$), if

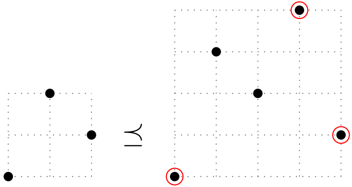
$$\forall i, j \in \{1, \ldots, k\} : s_i \leq_{\mathbb{N}} s_j \Leftrightarrow t_i \leq_{\mathbb{N}} t_j.$$

Definition (Involvement)

Seq. σ is involved in ($\preceq$) seq. τ, if $\sigma \cong \tau'$ for some subsequence τ' of τ.

Example

132 $\preceq$ 14352



One permutation to another

Definition (Sequences and order isomorphic)

$\sigma = s_1 s_2 s_3 \dots s_k$ is sequence $:\Leftrightarrow$ s_i distinct numbers taken from $(\mathbb{N}, \leq_{\mathbb{N}})$.
Sequences $\sigma = s_1 s_2 \dots s_k$, $\tau = t_1 t_2 \dots t_k$ are order isomorphic ($\sigma \cong \tau$), if

$$\forall i, j \in \{1, \dots, k\} : s_i \leq_{\mathbb{N}} s_j \Leftrightarrow t_i \leq_{\mathbb{N}} t_j.$$

Definition (Involvement)

Seq. σ is involved in ($\preceq$) seq. τ, if $\sigma \cong \tau'$ for some subsequence τ' of τ.

Definition

Let $\pi \in S_n$ and τ a permutation (pattern) of length k with $k \leq n$. The permutation π avoids τ if $\tau \not\preceq \pi$. Let T be a set of patterns.

$$\mathcal{A}_n(T) := \{\sigma \in S_n \mid \tau \not\preceq \sigma \text{ for all } \tau \in T\}$$

Separable and stack-sortable permutations

Direct sum and skew sum

Definition

For $\pi \in S_n$ and $\tau \in S_m$, the direct sum of π and τ is defined by

$$\pi \oplus \tau := \pi(1), \ldots, \pi(n), \tau(1) + n, \ldots, \tau(m) + n,$$

and the skew sum of π and τ by

$$\pi \ominus \tau := \pi(1) + m, \ldots, \pi(n) + m, \tau(1), \ldots, \tau(m).$$

Direct sum and skew sum

Definition

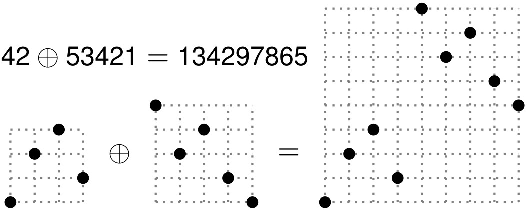
For $\pi \in S_n$ and $\tau \in S_m$, the direct sum of π and τ is defined by

$$\pi \oplus \tau := \pi(1), \ldots, \pi(n), \tau(1) + n, \ldots, \tau(m) + n,$$

and the skew sum of π and τ by

$$\pi \ominus \tau := \pi(1) + m, \ldots, \pi(n) + m, \tau(1), \ldots, \tau(m).$$

$1342 \oplus 53421 = 134297865$



Direct sum and skew sum

Definition

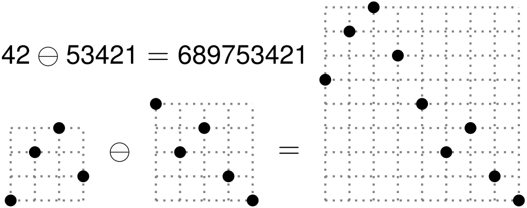
For $\pi \in S_n$ and $\tau \in S_m$, the direct sum of π and τ is defined by

$$\pi \oplus \tau := \pi(1), \ldots, \pi(n), \tau(1) + n, \ldots, \tau(m) + n,$$

and the skew sum of π and τ by

$$\pi \ominus \tau := \pi(1) + m, \ldots, \pi(n) + m, \tau(1), \ldots, \tau(m).$$

$$1342 \ominus 53421 = 689753421$$



Separable permutations

Definition (Separable permutation)

π a separable if either

- π is the identity permutation of length one, or
- there are separable permutations σ, τ such that $\pi = \sigma \oplus \tau$ or $\pi = \sigma \ominus \tau$.

Separable permutations

Definition (Separable permutation)

π a separable if either

- π is the identity permutation of length one, or
- there are separable permutations σ, τ such that $\pi = \sigma \oplus \tau$ or $\pi = \sigma \ominus \tau$.

Example (The permutation 765984132 is separable)

$$\begin{aligned} 765984132 &= 32154 \ominus 1 \ominus 132 \\ &= ((1 \ominus 1 \ominus 1) \oplus (1 \ominus 1)) \ominus 1 \ominus (1 \oplus (1 \ominus 1)) \end{aligned}$$

Separable permutations

Definition (Separable permutation)

π a separable if either

- π is the identity permutation of length one, or
- there are separable permutations σ, τ such that $\pi = \sigma \oplus \tau$ or $\pi = \sigma \ominus \tau$.

Example (The permutation 765984132 is separable)

$$\begin{aligned} 765984132 &= 32154 \ominus 1 \ominus 132 \\ &= ((1 \ominus 1 \ominus 1) \oplus (1 \ominus 1)) \ominus 1 \ominus (1 \oplus (1 \ominus 1)) \end{aligned}$$

Separable permutations

Definition (Separable permutation)

π a separable if either

- π is the identity permutation of length one, or
- there are separable permutations σ, τ such that $\pi = \sigma \oplus \tau$ or $\pi = \sigma \ominus \tau$.

Example (The permutation 765984132 is separable)

$$\begin{aligned} 765984132 &= 32154 \ominus 1 \ominus 132 \\ &= ((1 \ominus 1 \ominus 1) \oplus (1 \ominus 1)) \ominus 1 \ominus (1 \oplus (1 \ominus 1)) \end{aligned}$$

Theorem (Folklore)

Permutations that do not involve 2413 *and* 3142 *are separable, i. e.,* $\mathcal{A}(3124, 2413)$.

Stack-sortable permutations

Definition (Stack sortable)

$\pi \in S_n$ is stack-sortable if it may be sorted by an algorithm using a single stack data structure only.

Stack-sortable permutations

Definition (Stack sortable)

$\pi \in S_n$ is stack-sortable if it may be sorted by an algorithm using a single stack data structure only.

Theorem (Knuth (1968))

$\pi \in S_n$ is stack sortable, iff $\pi \in \mathcal{A}_n(231)$.

Stack-sortable permutations

Definition (Stack sortable)

$\pi \in S_n$ is stack-sortable if it may be sorted by an algorithm using a single stack data structure only.

Theorem (Knuth (1968))

$\pi \in S_n$ is stack sortable, iff $\pi \in \mathcal{A}_n(231)$.

Remark

Since $231 \preceq 3142$ and $231 \preceq 2413$ the $\mathcal{A}_{\leq n}(231) \subseteq \mathcal{A}_{\leq n}(\{3142, 2413\})$.
Further, for $\sigma, \tau \in \mathcal{A}_{\leq n}(231)$:

Stack-sortable permutations

Definition (Stack sortable)

$\pi \in S_n$ is stack-sortable if it may be sorted by an algorithm using a single stack data structure only.

Theorem (Knuth (1968))

$\pi \in S_n$ is stack sortable, iff $\pi \in \mathcal{A}_n(231)$.

Remark

Since $231 \preceq 3142$ and $231 \preceq 2413$ the $\mathcal{A}_{\leq n}(231) \subseteq \mathcal{A}_{\leq n}(\{3142, 2413\})$. Further, for $\sigma, \tau \in \mathcal{A}_{\leq n}(231)$:

- $\sigma \oplus \tau \in \mathcal{A}_{\leq n}(231)$

Stack-sortable permutations

Definition (Stack sortable)

$\pi \in S_n$ is stack-sortable if it may be sorted by an algorithm using a single stack data structure only.

Theorem (Knuth (1968))

$\pi \in S_n$ is stack sortable, iff $\pi \in \mathcal{A}_n(231)$.

Remark

Since $231 \preceq 3142$ and $231 \preceq 2413$ the $\mathcal{A}_{\leq n}(231) \subseteq \mathcal{A}_{\leq n}(\{3142, 2413\})$. Further, for $\sigma, \tau \in \mathcal{A}_{\leq n}(231)$:

- $\sigma \oplus \tau \in \mathcal{A}_{\leq n}(231)$
- $\sigma \ominus \tau \in \mathcal{A}_{\leq n}(231)$, *iff σ is not ascending.*

FPCSP($\{231\}$)

FPCSP($\{231\}$)

Problem (FPCSP($\{231\}$))

Input: *A tuple* $(\underline{n}, E_1, E_2)$ *where* $\underline{n} = \{1, \ldots, n\}$*, and* E_1, E_2 *are binary relations on* $\underline{n}$*.*

Question: *Are there linear orders* $L_1 \supseteq E_1$*,* $L_2 \supseteq E_2$ *such that* $\Pi(\underline{n}, L_1, L_2) \in \mathcal{A}_n(231)$*.*

FPCSP($\{231\}$)

Problem (FPCSP($\{231\}$))

Input: *A tuple* $(\underline{n}, E_1, E_2)$ *where* $\underline{n} = \{1, \ldots, n\}$*, and* E_1, E_2 *are binary relations on* $\underline{n}$*.*

Question: *Are there linear orders* $L_1 \supseteq E_1$*,* $L_2 \supseteq E_2$ *such that* $\Pi(\underline{n}, L_1, L_2) \in \mathcal{A}_n(231)$*.*

Theorem

FPCSP($\{231\}$) *is solvable in polynomial time.*

Graph tools

Definition (Strongly connected)

Directed graph $G = (V, E)$ is strongly connected if for any two vertices $v, w \in V$ there is a directed path in G from v to w.

Graph tools

Definition (Strongly connected)

Directed graph $G = (V, E)$ is strongly connected if for any two vertices $v, w \in V$ there is a directed path in G from v to w.

Lemma (Stack-sortable)

For (V, E_1, E_2) where E_1, E_2 acyclic binary relations on finite set V with $|V| \geq 2$.
If there are two linear extensions L_1, L_2 for E_1, E_2 such that $\pi := \Pi(V, L_1, L_2)$ is a stack-sortable permutation, then either

Graph tools

Definition (Strongly connected)

Directed graph $G = (V, E)$ is strongly connected if for any two vertices $v, w \in V$ there is a directed path in G from v to w.

Lemma (Stack-sortable)

For (V, E_1, E_2) where E_1, E_2 acyclic binary relations on finite set V with $|V| \geq 2$.
If there are two linear extensions L_1, L_2 for E_1, E_2 such that $\pi := \Pi(V, L_1, L_2)$ is a stack-sortable permutation, then either

- *$G := (V, E_1 \cup E_2)$ has more than one scc, or*

Graph tools

Definition (Strongly connected)

Directed graph $G = (V, E)$ is strongly connected if for any two vertices $v, w \in V$ there is a directed path in G from v to w.

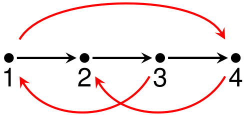
Lemma (Stack-sortable)

For (V, E_1, E_2) where E_1, E_2 acyclic binary relations on finite set V with $|V| \geq 2$.
If there are two linear extensions L_1, L_2 for E_1, E_2 such that $\pi := \Pi(V, L_1, L_2)$ is a stack-sortable permutation, then either

- *$G := (V, E_1 \cup E_2)$ has more than one scc, or*
- *$G' := (V, E_1 \cup E_2^{-1})$ has more than one strongly connected component and a final scc C has no accents (i.e., there is a solution to FPCSP(12)).*

Outline of algorithm by example

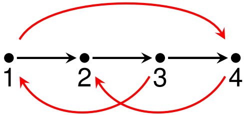
Example 1



Outline of algorithm by example

1. Identify strongly connected components

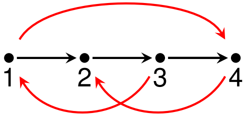
Example 1



Outline of algorithm by example

1. Identify strongly connected components
2. If only one, check inverted with E_2

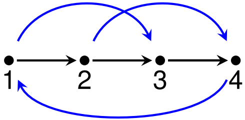
Example 1



Outline of algorithm by example

1. Identify strongly connected components
2. If only one, check inverted with E_2

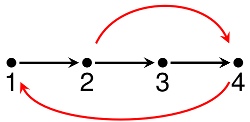
Example 1



Outline of algorithm by example

1. Identify strongly connected components
2. If only one, check inverted with E_2

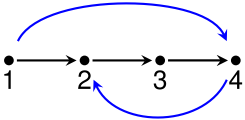
Example 2



Outline of algorithm by example

1. Identify strongly connected components
2. If only one, check inverted with E_2

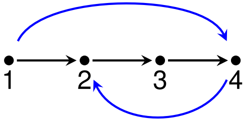
Example 2



Outline of algorithm by example

1. Identify strongly connected components
2. If only one, check inverted with E_2
3. If more than one scc, check a final scc for not accending calls on terminal scc and complement.

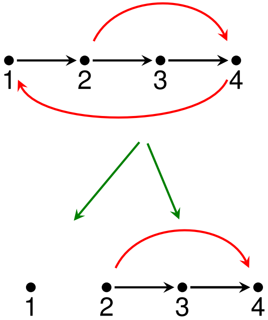
Example 2



Outline of algorithm by example

1. Identify strongly connected components
2. If only one, check inverted with E_2
3. If more than one scc, check a final scc for not accending calls on terminal scc and complement.

Example 2



Thank you very much!