Recognizing merge classes

Generalized coloring of permutations

Michal Opler
Joint work with Vít Jelínek and Pavel Valtr

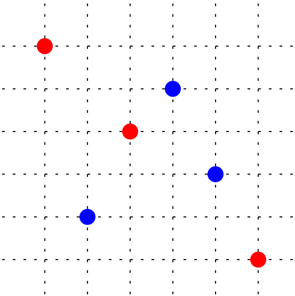
Computer Science Institute of Charles University in Prague

Permutation Patterns
July 2018

Merges

Definition

Permutation π is a merge of permutations σ and τ if the elements of π can be colored red and blue, so that the red elements are a copy of σ and the blue ones of τ.



One possible merge of 132 and 321 is 624531.

Recognition problems

Definition

For two permutation classes $\mathcal{C}$ and $\mathcal{D}$, let $\mathcal{C} \odot \mathcal{D}$ be the class of permutations obtained by merging a $\sigma \in \mathcal{C}$ with a $\tau \in \mathcal{D}$.

Recognition problems

Definition

For two permutation classes $\mathcal{C}$ and $\mathcal{D}$, let $\mathcal{C} \odot \mathcal{D}$ be the class of permutations obtained by merging a $\sigma \in \mathcal{C}$ with a $\tau \in \mathcal{D}$.

Definition

For a permutation class $\mathcal{C}$, $\mathcal{C}$-recognition is the decision problem to determine whether a given permutation belongs to $\mathcal{C}$.

Example: $\mathrm{Av}(k \ldots 1)$-recognition is the usual coloring problem.

Recognition problems

Definition

For two permutation classes $\mathcal{C}$ and $\mathcal{D}$, let $\mathcal{C} \odot \mathcal{D}$ be the class of permutations obtained by merging a $\sigma \in \mathcal{C}$ with a $\tau \in \mathcal{D}$.

Definition

For a permutation class $\mathcal{C}$, $\mathcal{C}$-recognition is the decision problem to determine whether a given permutation belongs to $\mathcal{C}$.

Example: Av($k \dots 1$)-recognition is the usual coloring problem.

Question

How hard is the $(\mathcal{C} \odot \mathcal{D})$-recognition problem for various choices of $\mathcal{C}$ and $\mathcal{D}$.

Generalized coloring of graphs

Definition

For a fixed k-tuple $\mathcal{G}_1, \dots, \mathcal{G}_k$ of graph classes, a generalized coloring of a graph is an assignment of colors $1, 2, \dots, k$ to its vertices so that the vertices of color i induce a subgraph from $\mathcal{G}_i$.

In particular, if all the $\mathcal{G}_i$ are equal to the class of edgeless graphs, this notion reduces to the classical notion of k-coloring.

Generalized coloring of graphs

Definition

For a fixed k-tuple $\mathcal{G}_1, \ldots, \mathcal{G}_k$ of graph classes, a generalized coloring of a graph is an assignment of colors $1, 2, \ldots, k$ to its vertices so that the vertices of color i induce a subgraph from $\mathcal{G}_i$.

In particular, if all the $\mathcal{G}_i$ are equal to the class of edgeless graphs, this notion reduces to the classical notion of k-coloring.

Theorem (Farrugia)

If all the $\mathcal{G}_i$ are hereditary and additive (i.e., closed under taking induced subgraphs and forming disjoint unions) then the problem is NP-hard, except the trivially polynomial case when $k = 2$ and both $\mathcal{G}_1$ and $\mathcal{G}_2$ are equal to the class of edgeless graphs.

Prior results

Facts

($\mathcal{C} \odot \mathcal{D}$)-recognition is tractable, i.e. polynomially solvable, whenever

- *$\mathcal{C}$-recognition is tractable and $\mathcal{D}$ is finite,*
- *$\mathcal{C} = Av(1 \dots k)$ and $\mathcal{D} = Av(1 \dots l)$, and*
- *$\mathcal{C} = Av(1 \dots k)$ and $\mathcal{D} = Av(l \dots 1)$.*

Prior results

Facts

$(\mathcal{C} \odot \mathcal{D})$-recognition is tractable, i.e. polynomially solvable, whenever

- *$\mathcal{C}$-recognition is tractable and $\mathcal{D}$ is finite,*
- *$\mathcal{C} = Av(1 \dots k)$ and $\mathcal{D} = Av(1 \dots l)$, and*
- *$\mathcal{C} = Av(1 \dots k)$ and $\mathcal{D} = Av(l \dots 1)$.*

Observation

$(\mathcal{C} \odot \mathcal{D})$-recognition is tractable, whenever $\mathcal{C} \odot \mathcal{D}$ has a finite basis.

Prior results

Facts

$(\mathcal{C} \odot \mathcal{D})$-recognition is tractable, i.e. polynomially solvable, whenever

- *$\mathcal{C}$-recognition is tractable and $\mathcal{D}$ is finite,*
- *$\mathcal{C} = Av(1 \dots k)$ and $\mathcal{D} = Av(1 \dots l)$, and*
- *$\mathcal{C} = Av(1 \dots k)$ and $\mathcal{D} = Av(l \dots 1)$.*

Observation

$(\mathcal{C} \odot \mathcal{D})$-recognition is tractable, whenever $\mathcal{C} \odot \mathcal{D}$ has a finite basis.

Theorem (Ekim et al.)

There is a polynomial algorithm for

- *$(\mathcal{L} \odot Av(21))$-recognition and*
- *$(\mathcal{L} \odot \overline{\mathcal{L}})$-recognition, where*

$\mathcal{L}$ is the class of layered permutations and $\overline{\mathcal{L}}$ is the class of co-layered permutations.

Online recognition motivation

Problem: Decide whether π of length n belongs to $\mathrm{Av}(12) \odot \mathrm{Av}(21)$.

Online recognition motivation

Problem: Decide whether π of length n belongs to $\mathrm{Av}(12) \odot \mathrm{Av}(21)$.

Definition

Let S_k be set of pairs (a, b) such that $\pi_1, \ldots \pi_k$ can be decomposed into

- a decreasing sequence with smallest (last) value a and
- an increasing sequence with largest (last) value b.

Moreover $S_0 = \{(+\infty, -\infty)\}$.

Online recognition motivation

Problem: Decide whether π of length n belongs to $\text{Av}(12) \odot \text{Av}(21)$.

Definition

Let S_k be set of pairs (a, b) such that $\pi_1, \ldots \pi_k$ can be decomposed into

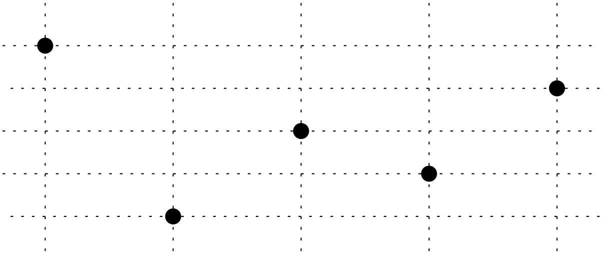
- a decreasing sequence with smallest (last) value a and
- an increasing sequence with largest (last) value b.

Moreover $S_0 = \{(+\infty, -\infty)\}$.

Observation

π belongs to $\text{Av}(12) \odot \text{Av}(21)$ if and only if S_n is not empty.

Algorithm: Incrementally compute $S_1, \ldots, S_n$.



π_i		5	1	3	2	4
S_i	$(+\infty, -\infty)$	$(5, -\infty)$ $(+\infty, 5)$	$(1, -\infty)$ $(5, 1)$ $(1, 5)$	$(1, 3)$ $(5, 3)$ $(3, 1)$	$(2, 3)$ $(3, 2)$ $(2, 1)$	$(2, 4)$ $(3, 4)$

Run of the algorithm on permutation 51324.

Nondeterministic point of view

Suppose we are trying to solve the same problem nondeterministically.

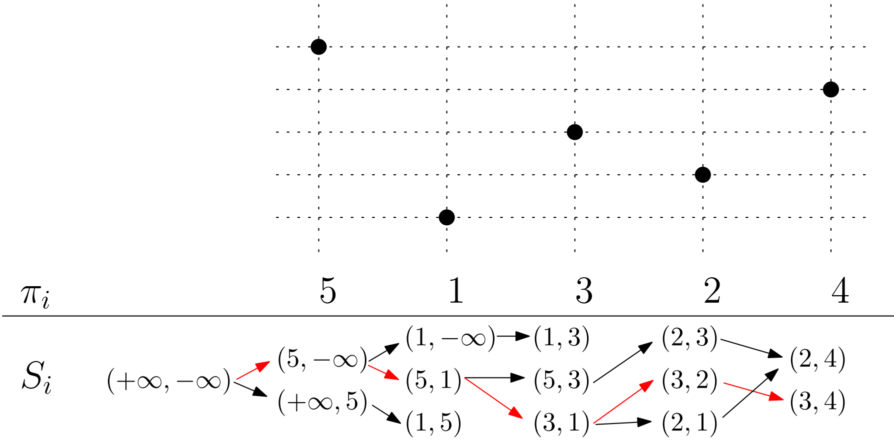
Nondeterministic point of view

Suppose we are trying to solve the same problem nondeterministically. Let A be a nondeterministic algorithm that stores in memory only pair of the last values in each sequence.

- A initializes its memory state to $(+\infty, -\infty)$.
- For $k = 1 \dots n$
  - A guesses whether π_k belongs to increasing or decreasing sequence, and
  - updates the pair of last values or terminates unsuccessfully.
  - A accepts π if there is valid pair in its memory after receiving π_n.

Observation

There is an accepting computation of A on π if and only if π belongs to $\mathrm{Av}(12) \odot \mathrm{Av}(21)$.



Possible run of the nondeterministic algorithm that decomposes 51324 into 53 and 124.

NLOL-recognition

We say that a nondeterministic algorithm A is a nondeterministic logspace on-line recognizer of $\mathcal{C}$ if it recognizes $\mathcal{C}$ in the following setting:

- A receives an integer n, then
- A receives *one-by-one* a sequence of distinct values $\pi_1, \ldots, \pi_k$ from the set $[n]$, terminated by a special symbol `EOF`.
- Afterwards, A answers whether $\pi_1, \ldots, \pi_k$ is order-isomorphic to some $\pi' \in \mathcal{C}$.
- During the computation A can read the input sequence only once and access only $O(\log n)$ bits of memory.

NLOL-recognition

We say that a nondeterministic algorithm A is a nondeterministic logspace on-line recognizer of $\mathcal{C}$ if it recognizes $\mathcal{C}$ in the following setting:

- A receives an integer n, then
- A receives *one-by-one* a sequence of distinct values $\pi_1, \ldots, \pi_k$ from the set $[n]$, terminated by a special symbol `EOF`.
- Afterwards, A answers whether $\pi_1, \ldots, \pi_k$ is order-isomorphic to some $\pi' \in \mathcal{C}$.
- During the computation A can read the input sequence only once and access only $O(\log n)$ bits of memory.

Definition

A permutation class $\mathcal{C}$ is nondeterministically logspace on-line recognizable, or NLOL-recognizable for short, if there is an NLOL recognizer A of $\mathcal{C}$.

NLOL-recognition

We say that a nondeterministic algorithm A is a nondeterministic logspace on-line recognizer of $\mathcal{C}$ if it recognizes $\mathcal{C}$ in the following setting:

- A receives an integer n, then
- A receives *one-by-one* a sequence of distinct values $\pi_1, \ldots, \pi_k$ from the set $[n]$, terminated by a special symbol `EOF`.
- Afterwards, A answers whether $\pi_1, \ldots, \pi_k$ is order-isomorphic to some $\pi' \in \mathcal{C}$.
- During the computation A can read the input sequence only once and access only $O(\log n)$ bits of memory.

Definition

A permutation class $\mathcal{C}$ is nondeterministically logspace on-line recognizable, or NLOL-recognizable for short, if there is an NLOL recognizer A of $\mathcal{C}$.

Observation

For any permutation class NLOL-recognizable class $\mathcal{C}$, the $\mathcal{C}$-recognition problem is tractable.

Properties of NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are NLOL-recognizable classes, then the following classes are NLOL-recognizable as well:

Properties of NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are NLOL-recognizable classes, then the following classes are NLOL-recognizable as well:

(a) *The classes $\mathcal{C} \cap \mathcal{D}$ and $\mathcal{C} \cup \mathcal{D}$.*

Properties of NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are NLOL-recognizable classes, then the following classes are NLOL-recognizable as well:

(a) *The classes $\mathcal{C} \cap \mathcal{D}$ and $\mathcal{C} \cup \mathcal{D}$.*

(b) *The classes $\mathcal{C}^r$ and $\mathcal{C}^c$, i.e. the classes of reverses and complements.*

Properties of NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are NLOL-recognizable classes, then the following classes are NLOL-recognizable as well:

(a) *The classes $\mathcal{C} \cap \mathcal{D}$ and $\mathcal{C} \cup \mathcal{D}$.*

(b) *The classes $\mathcal{C}^r$ and $\mathcal{C}^c$, i.e. the classes of reverses and complements.*

(c) *The classes $\mathcal{C}^{\oplus}$ and $\mathcal{C}^{\ominus}$, i.e., the sum-closure and skew-closure of $\mathcal{C}$.*

Properties of NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are NLOL-recognizable classes, then the following classes are NLOL-recognizable as well:

(a) *The classes $\mathcal{C} \cap \mathcal{D}$ and $\mathcal{C} \cup \mathcal{D}$.*

(b) *The classes $\mathcal{C}^r$ and $\mathcal{C}^c$, i.e. the classes of reverses and complements.*

(c) *The classes $\mathcal{C}^{\oplus}$ and $\mathcal{C}^{\ominus}$, i.e., the sum-closure and skew-closure of $\mathcal{C}$.*

(d) *The classes $\mathcal{C} \oplus \mathcal{D}$ and $\mathcal{C} \ominus \mathcal{D}$, and more generally, any class of the form Grid(M), where M is a matrix of classes whose entries all belong to NLOL.*

Properties of NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are NLOL-recognizable classes, then the following classes are NLOL-recognizable as well:

(a) *The classes $\mathcal{C} \cap \mathcal{D}$ and $\mathcal{C} \cup \mathcal{D}$.*

(b) *The classes $\mathcal{C}^r$ and $\mathcal{C}^c$, i.e. the classes of reverses and complements.*

(c) *The classes $\mathcal{C}^{\oplus}$ and $\mathcal{C}^{\ominus}$, i.e., the sum-closure and skew-closure of $\mathcal{C}$.*

(d) *The classes $\mathcal{C} \oplus \mathcal{D}$ and $\mathcal{C} \ominus \mathcal{D}$, and more generally, any class of the form Grid(M), where M is a matrix of classes whose entries all belong to NLOL.*

(e) *The class $\mathcal{C} \odot \mathcal{D}$.*

Properties of NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are NLOL-recognizable classes, then the following classes are NLOL-recognizable as well:

(a) *The classes $\mathcal{C} \cap \mathcal{D}$ and $\mathcal{C} \cup \mathcal{D}$.*

(b) *The classes $\mathcal{C}^r$ and $\mathcal{C}^c$, i.e. the classes of reverses and complements.*

(c) *The classes $\mathcal{C}^{\oplus}$ and $\mathcal{C}^{\ominus}$, i.e., the sum-closure and skew-closure of $\mathcal{C}$.*

(d) *The classes $\mathcal{C} \oplus \mathcal{D}$ and $\mathcal{C} \ominus \mathcal{D}$, and more generally, any class of the form Grid(M), where M is a matrix of classes whose entries all belong to NLOL.*

(e) *The class $\mathcal{C} \odot \mathcal{D}$.*

Corollary

For any sequence of classes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k \in$ NLOL, the class $\mathcal{C}_1 \odot \mathcal{C}_2 \odot \cdots \odot \mathcal{C}_k$ is in NLOL, and therefore polynomially recognizable.

2D-NLOL-recognizable classes

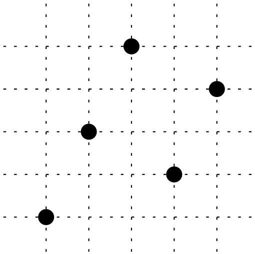
Definition

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ be a sequence of distinct points in general position. We say that the sequence is top-right monotone if for every $i \in [k]$ the point (x_i, y_i) is to the right or above all the previous points of the sequence.

2D-NLOL-recognizable classes

Definition

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ be a sequence of distinct points in general position. We say that the sequence is top-right monotone if for every $i \in [k]$ the point (x_i, y_i) is to the right or above all the previous points of the sequence.

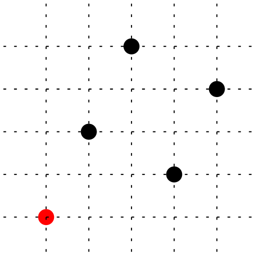


Possible sequence:

2D-NLOL-recognizable classes

Definition

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ be a sequence of distinct points in general position. We say that the sequence is top-right monotone if for every $i \in [k]$ the point (x_i, y_i) is to the right or above all the previous points of the sequence.

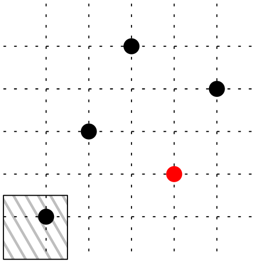


Possible sequence: $(1, 1)$

2D-NLOL-recognizable classes

Definition

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ be a sequence of distinct points in general position. We say that the sequence is top-right monotone if for every $i \in [k]$ the point (x_i, y_i) is to the right or above all the previous points of the sequence.

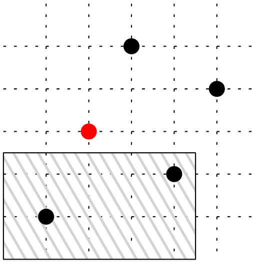


Possible sequence: $(1, 1)$, $(4, 2)$

2D-NLOL-recognizable classes

Definition

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ be a sequence of distinct points in general position. We say that the sequence is top-right monotone if for every $i \in [k]$ the point (x_i, y_i) is to the right or above all the previous points of the sequence.

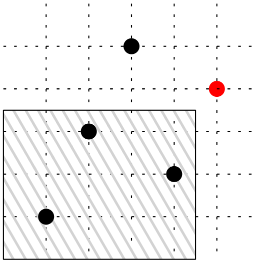


Possible sequence: $(1, 1)$, $(4, 2)$, $(2, 3)$

2D-NLOL-recognizable classes

Definition

Let $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ be a sequence of distinct points in general position. We say that the sequence is top-right monotone if for every $i \in [k]$ the point (x_i, y_i) is to the right or above all the previous points of the sequence.

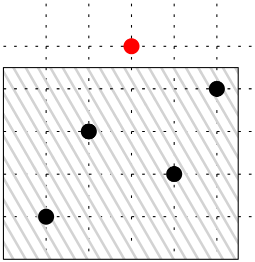


Possible sequence: $(1, 1)$, $(4, 2)$, $(2, 3)$, $(5, 4)$

2D-NLOL-recognizable classes

Definition

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$ be a sequence of distinct points in general position. We say that the sequence is top-right monotone if for every $i \in [k]$ the point (x_i, y_i) is to the right or above all the previous points of the sequence.



Possible sequence: $(1, 1)$, $(4, 2)$, $(2, 3)$, $(5, 4)$, $(3, 4)$

2D-NLOL-recognizable classes

We say that a nondeterministic algorithm A is a 2D-NLOL-recognizer of $\mathcal{C}$ if it recognizes $\mathcal{C}$ in the following setting[1]:

- A receives an integer n, then
- A receives *one-by-one* a sequence of a top-right monotone sequence of points in general position from $[n] \times [n]$ as their input, terminated by a special symbol `EOF`.
- Afterwards, A answers whether the set of points corresponds to some $\pi \in C$.
- During the computation A can read the input sequence only once and access only $O(\log n)$ bits of memory.

[1] Omitting few ugly technical conditions.

2D-NLOL-recognizable classes

We say that a nondeterministic algorithm A is a 2D-NLOL-recognizer of $\mathcal{C}$ if it recognizes $\mathcal{C}$ in the following setting[1]:

- A receives an integer n, then
- A receives *one-by-one* a sequence of a top-right monotone sequence of points in general position from $[n] \times [n]$ as their input, terminated by a special symbol `EOF`.
- Afterwards, A answers whether the set of points corresponds to some $\pi \in C$.
- During the computation A can read the input sequence only once and access only $O(\log n)$ bits of memory.

Definition

A permutation class $\mathcal{C}$ is 2D-NLOL-recognizable if there is a 2D-NLOL recognizer A of $\mathcal{C}$.

[1]Omitting few ugly technical conditions.

2D-NLOL-recognizable classes

We say that a nondeterministic algorithm A is a 2D-NLOL-recognizer of $\mathcal{C}$ if it recognizes $\mathcal{C}$ in the following setting[1]:

- A receives an integer n, then
- A receives *one-by-one* a sequence of a top-right monotone sequence of points in general position from $[n] \times [n]$ as their input, terminated by a special symbol `EOF`.
- Afterwards, A answers whether the set of points corresponds to some $\pi \in C$.
- During the computation A can read the input sequence only once and access only $O(\log n)$ bits of memory.

Definition

A permutation class $\mathcal{C}$ is 2D-NLOL-recognizable if there is a 2D-NLOL recognizer A of $\mathcal{C}$.

Observation

If a permutation class $\mathcal{C}$ is 2D-NLOL-recognizable then it is NLOL-recognizable.

[1] Omitting few ugly technical conditions.

2D-NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are 2D-NLOL-recognizable classes, then the following classes are 2D-NLOL-recognizable as well:

(a) *The classes $\mathcal{C} \cap \mathcal{D}$ and $\mathcal{C} \cup \mathcal{D}$.*

(b) *The class $\mathcal{C}^{-1}$, which contains the inverses of the permutations of $\mathcal{C}$.*

(c) *The classes $\mathcal{C}^{\oplus}$ and $\mathcal{C}^{\ominus}$, i.e., the sum-closure and skew-closure of $\mathcal{C}$.*

(d) *The classes $\mathcal{C} \oplus \mathcal{D}$ and $\mathcal{C} \ominus \mathcal{D}$, and more generally, any class of the form Grid(M), where M is a matrix of classes whose entries all belong to 2D-NLOL.*

(e) *The class $\mathcal{C} \odot \mathcal{D}$.*

2D-NLOL-recognizable classes

Lemma

If $\mathcal{C}$ and $\mathcal{D}$ are 2D-NLOL-recognizable classes, then the following classes are 2D-NLOL-recognizable as well:

(a) *The classes $\mathcal{C} \cap \mathcal{D}$ and $\mathcal{C} \cup \mathcal{D}$.*

(b) *The class $\mathcal{C}^{-1}$, which contains the inverses of the permutations of $\mathcal{C}$.*

(c) *The classes $\mathcal{C}^{\oplus}$ and $\mathcal{C}^{\ominus}$, i.e., the sum-closure and skew-closure of $\mathcal{C}$.*

(d) *The classes $\mathcal{C} \oplus \mathcal{D}$ and $\mathcal{C} \ominus \mathcal{D}$, and more generally, any class of the form Grid(M), where M is a matrix of classes whose entries all belong to 2D-NLOL.*

(e) *The class $\mathcal{C} \odot \mathcal{D}$.*

Theorem

If $\mathcal{C}$ is a 2D-NLOL-recognizable class and $\mathcal{D}$ is any class of the set $\{Av(2413, 3142), Av(213), Av(231), Av(132), Av(312)\}$, then $\mathcal{C} \odot \mathcal{D}$ is polynomially recognizable.

Other results

Other results

Proposition

For a non-monotonic separable permutation σ, the class $Av(\sigma) \odot Av(21)$ has an infinite basis.

Other results

Proposition

For a non-monotonic separable permutation σ, the class $Av(\sigma) \odot Av(21)$ has an infinite basis.

Theorem

For any simple permutation α of order at least 4, the $(Av(\alpha) \odot Av(\alpha))$-recognition problem is NP-complete.

Other results

Proposition

For a non-monotonic separable permutation σ, the class $Av(\sigma) \odot Av(21)$ has an infinite basis.

Theorem

For any simple permutation α of order at least 4, the $(Av(\alpha) \odot Av(\alpha))$-recognition problem is NP-complete.

Proposition

The $(Av(\sigma) \odot Av(21))$-recognition problem is tractable whenever $\sigma = (1 \dots k) \oplus (l \dots 1) \oplus (1 \dots m)$ for $k \geq 0$, $l \geq 2$ and $m \geq 1$.

Summary

Summary

- There is a large set NLOL of permutation classes such that for any $\mathcal{C}$, $\mathcal{D} \in$ NLOL the $(\mathcal{C} \odot \mathcal{D})$-recognition is tractable. Specific examples include layered permutations, co-layered permutations, merge of k increasing sequences, separable permutations with constant depth of their separating tree etc.

Summary

- There is a large set NLOL of permutation classes such that for any $\mathcal{C}$, $\mathcal{D} \in$ NLOL the $(\mathcal{C} \odot \mathcal{D})$-recognition is tractable. Specific examples include layered permutations, co-layered permutations, merge of k increasing sequences, separable permutations with constant depth of their separating tree etc.
- For slightly restricted set 2D-NLOL of permutation classes we can recognize its merges with "treelike" permutation classes that include separable permutations and Av(213) (plus all its symmetries).

Summary

- There is a large set NLOL of permutation classes such that for any $\mathcal{C}$, $\mathcal{D} \in$ NLOL the $(\mathcal{C} \odot \mathcal{D})$-recognition is tractable. Specific examples include layered permutations, co-layered permutations, merge of k increasing sequences, separable permutations with constant depth of their separating tree etc.
- For slightly restricted set 2D-NLOL of permutation classes we can recognize its merges with "treelike" permutation classes that include separable permutations and Av(213) (plus all its symmetries).
- The problem becomes hard for forbidden patterns that are simple and of length at least 4.

Open questions

Open questions

Question

What is the complexity of $(\mathcal{C} \odot \mathcal{D})$-recognition when $\mathcal{C}$ and $\mathcal{D}$ are any two (possibly identical) classes from the set $\{\mathrm{Av}(2413, 3142), \mathrm{Av}(213), \mathrm{Av}(231), \mathrm{Av}(132), \mathrm{Av}(312)\}$?

Open questions

Question

What is the complexity of $(\mathcal{C} \odot \mathcal{D})$-recognition when $\mathcal{C}$ and $\mathcal{D}$ are any two (possibly identical) classes from the set $\{\mathrm{Av}(2413, 3142), \mathrm{Av}(213), \mathrm{Av}(231), \mathrm{Av}(132), \mathrm{Av}(312)\}$?

Question

For which classes $\mathcal{C}$ is the $(\mathcal{C} \odot \mathrm{Av}(21))$-recognition polynomial?

Thank you!