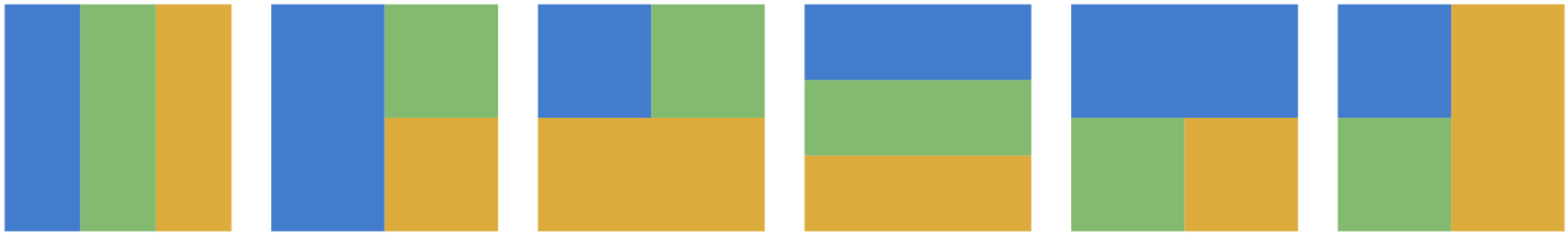
Permutation Patterns 2021

Pattern-avoiding rectangulations and permutations



Arturo Merino and Torsten Mütze

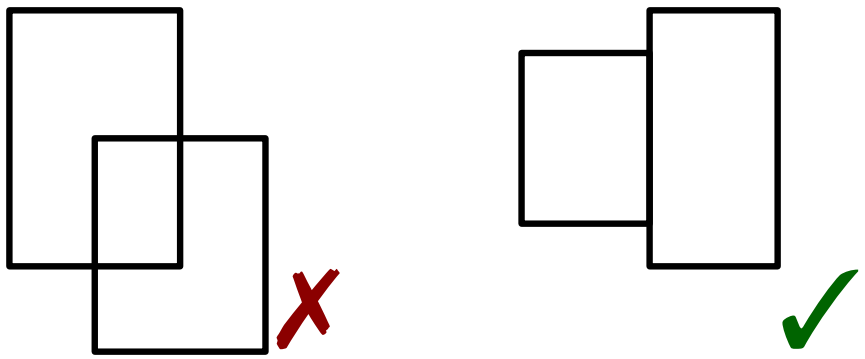
Generic rectangulations [Reading 12]

Generic rectangulation: partition of the unit square into rectangles such that:

Generic rectangulations [Reading 12]

Generic rectangulation: partition of the unit square into rectangles such that:

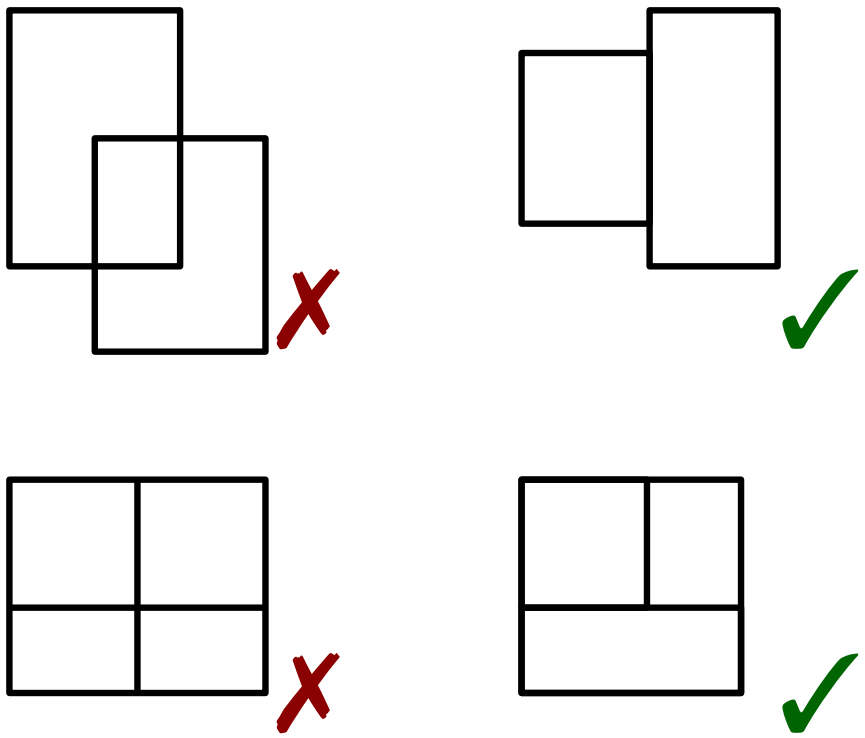
- the rectangles are interior disjoint, and



Generic rectangulations [Reading 12]

Generic rectangulation: partition of the unit square into rectangles such that:

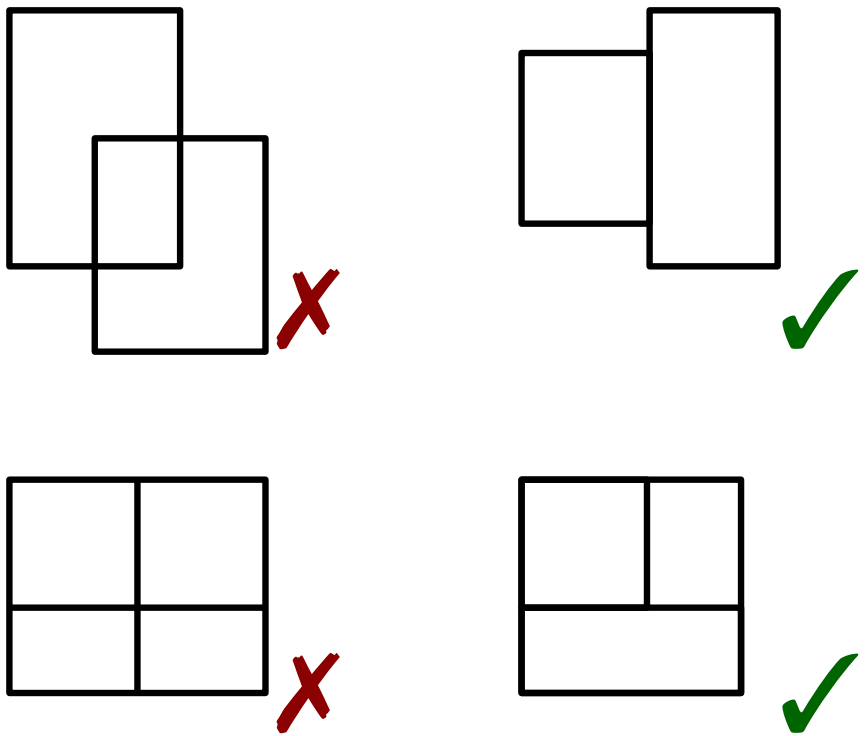
- the rectangles are interior disjoint, and
- no four rectangles share a corner.

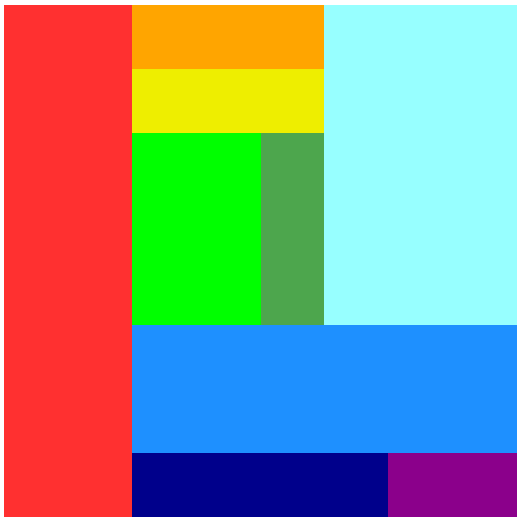


Generic rectangulations [Reading 12]

Generic rectangulation: partition of the unit square into rectangles such that:

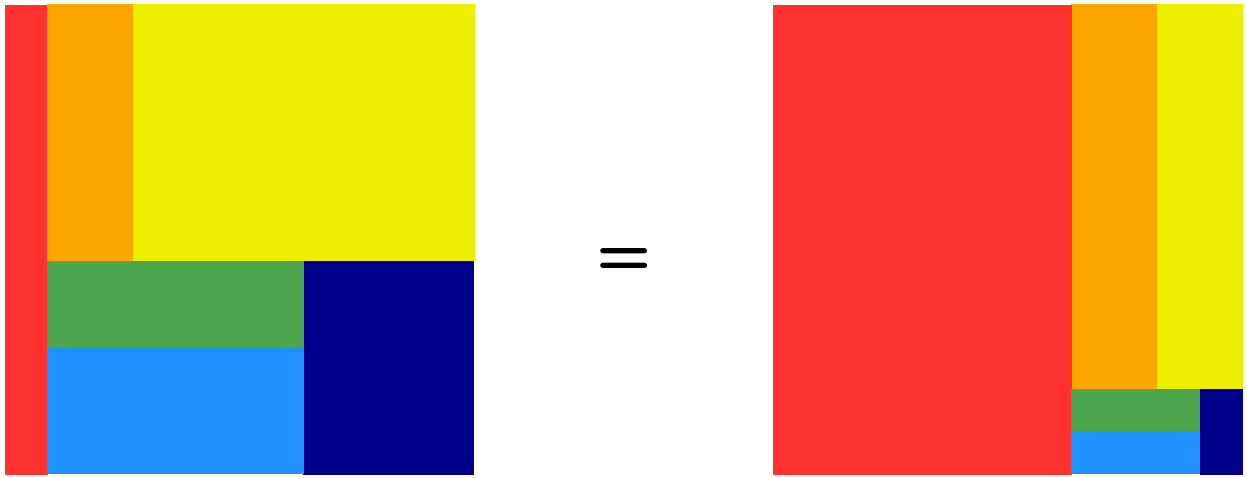
- the rectangles are interior disjoint, and
- no four rectangles share a corner.





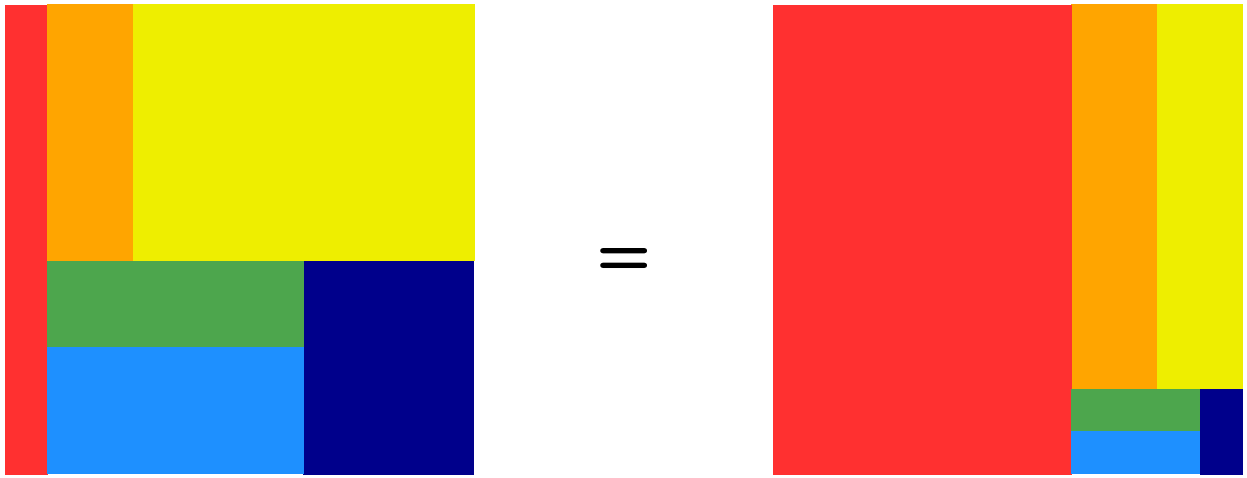
Generic rectangulations [Reading 12]

Generic rectangulation: up to combinatorial equivalence.



Generic rectangulations [Reading 12]

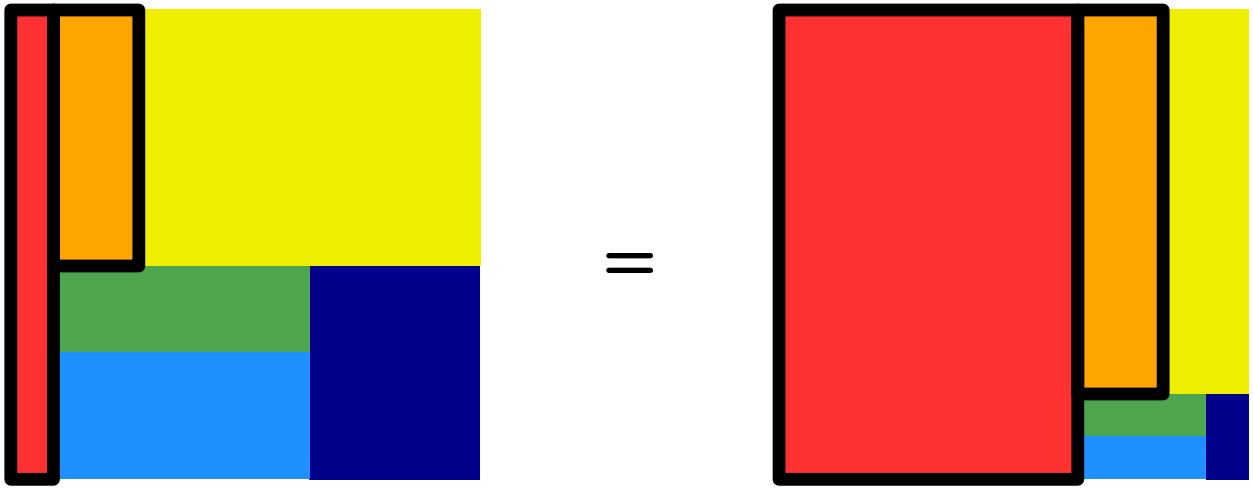
Generic rectangulation: up to combinatorial equivalence.



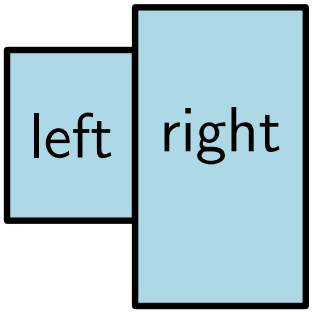
Two generic rectangulations are **equivalent** iff they share adjacencies.

Generic rectangulations [Reading 12]

Generic rectangulation: up to combinatorial equivalence.

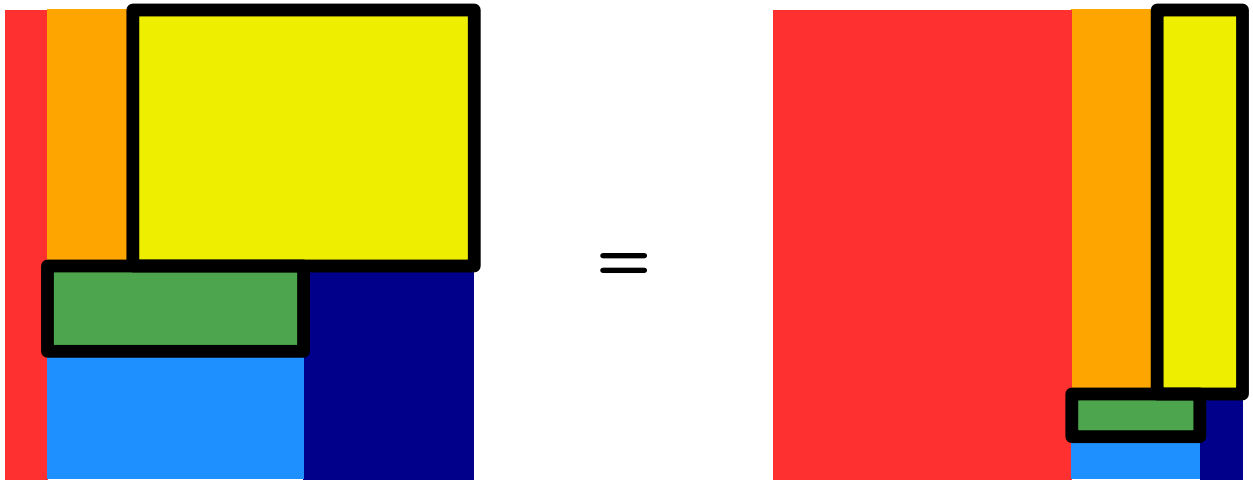


Two generic rectangulations are **equivalent** iff they share adjacencies.

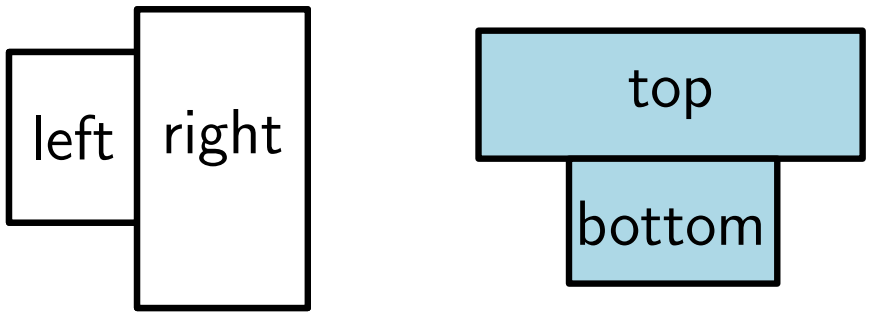


Generic rectangulations [Reading 12]

Generic rectangulation: up to combinatorial equivalence.

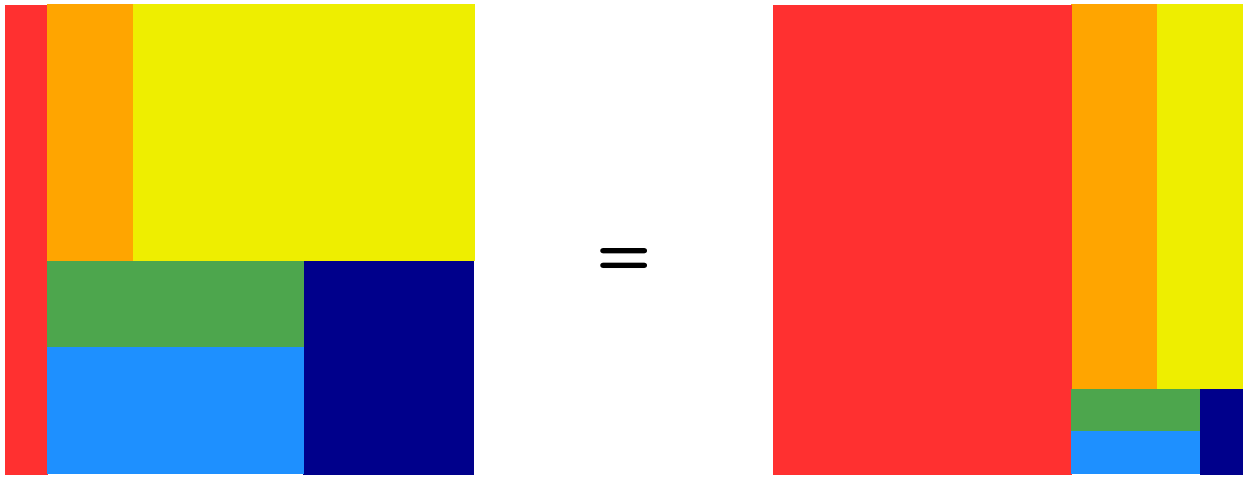


Two generic rectangulations are **equivalent** iff they share adjacencies.

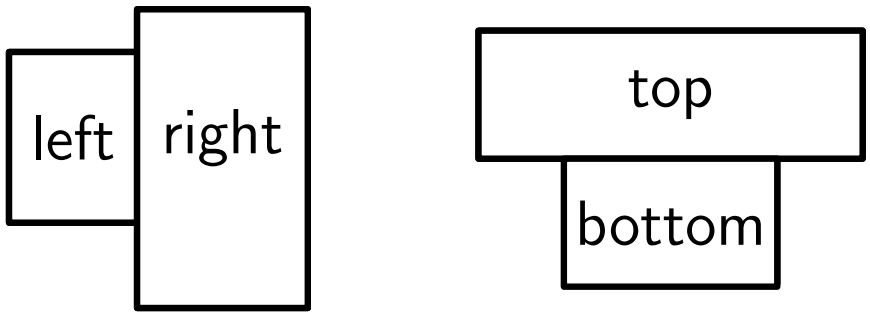


Generic rectangulations [Reading 12]

Generic rectangulation: up to combinatorial equivalence.



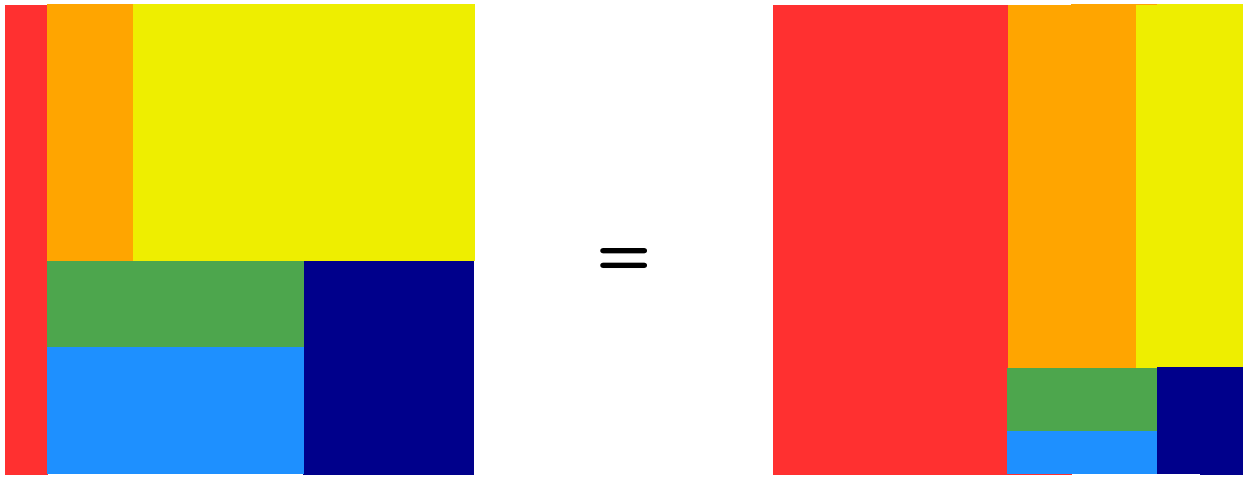
Two generic rectangulations are **equivalent** iff they share adjacencies.



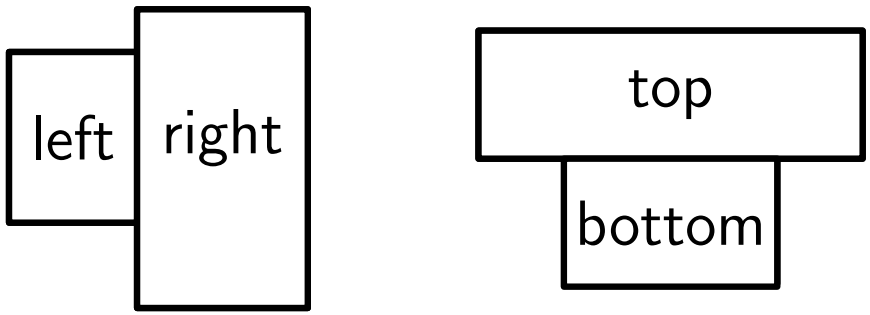
We can **stretch** one rectangulation into another without changing adjacencies

Generic rectangulations [Reading 12]

Generic rectangulation: up to combinatorial equivalence.



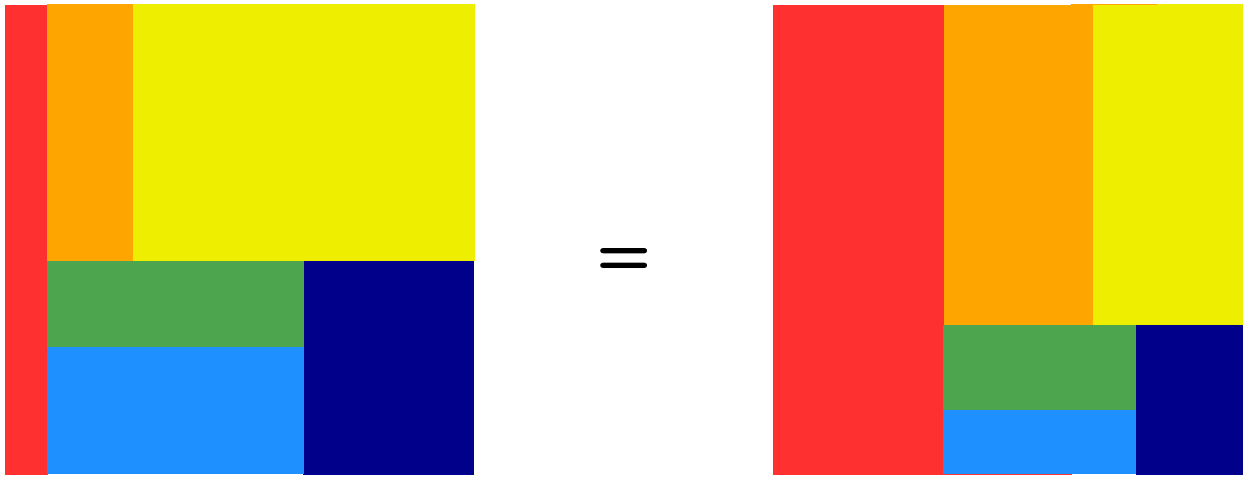
Two generic rectangulations are **equivalent** iff they share adjacencies.



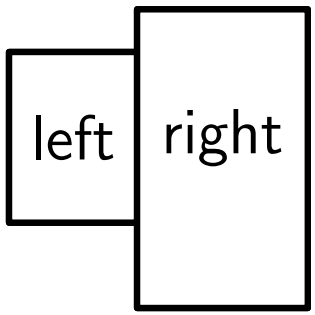
We can **stretch** one rectangulation into another without changing adjacencies

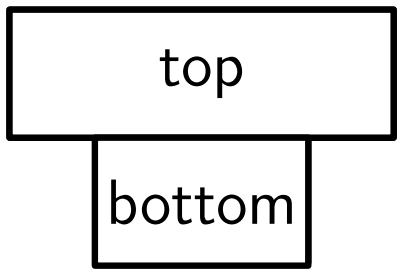
Generic rectangulations [Reading 12]

Generic rectangulation: up to combinatorial equivalence.



Two generic rectangulations are **equivalent** iff they share adjacencies.

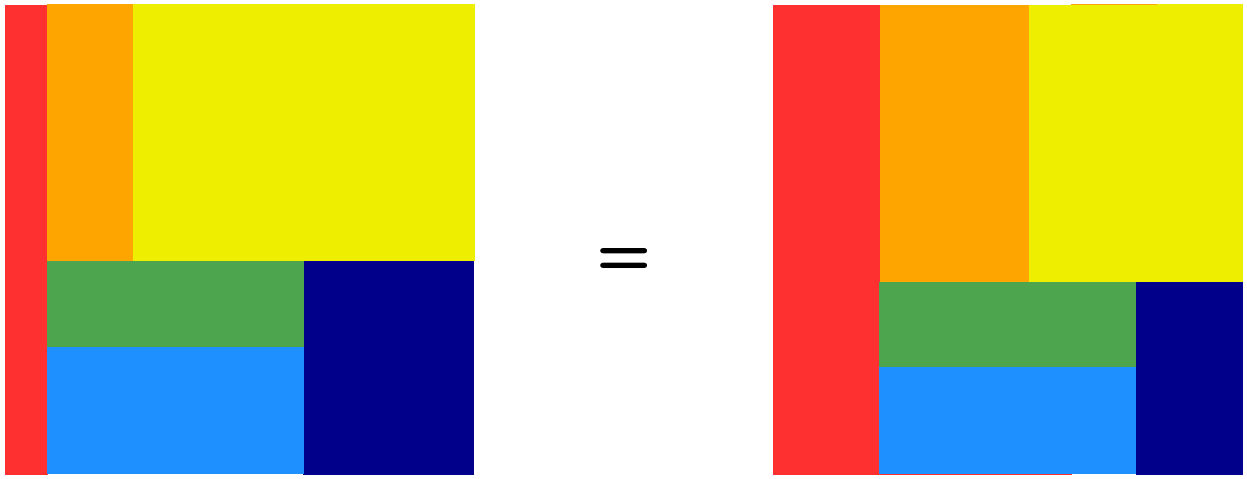




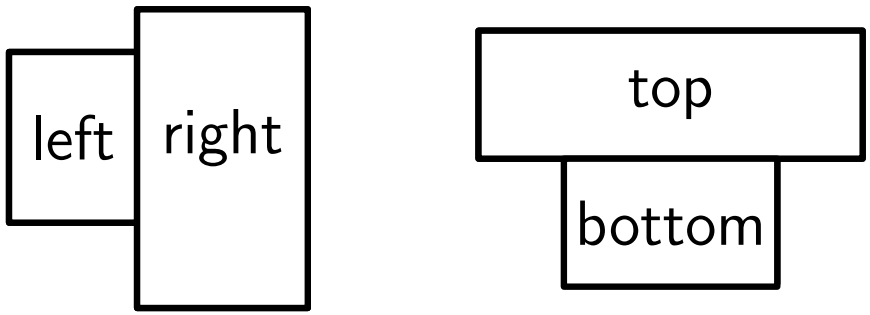
We can **stretch** one rectangulation into another without changing adjacencies

Generic rectangulations [Reading 12]

Generic rectangulation: up to combinatorial equivalence.



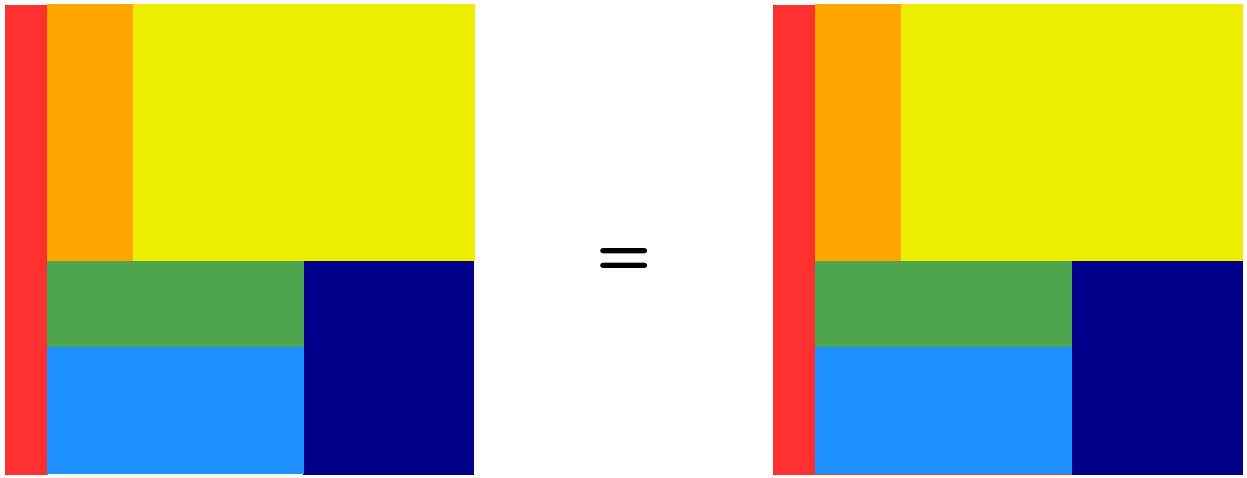
Two generic rectangulations are **equivalent** iff they share adjacencies.



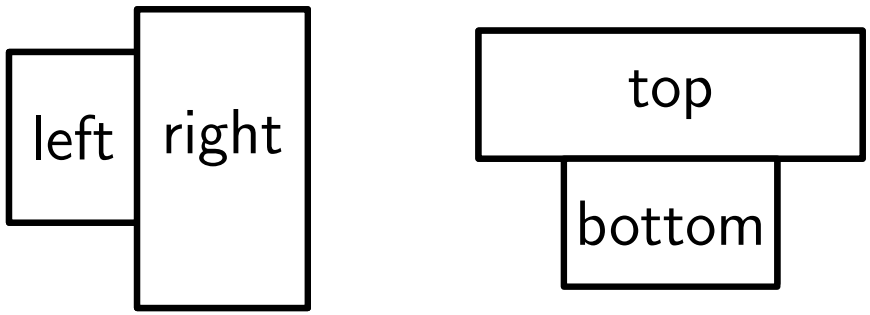
We can **stretch** one rectangulation into another without changing adjacencies

Generic rectangulations [Reading 12]

Generic rectangulation: up to combinatorial equivalence.



Two generic rectangulations are **equivalent** iff they share adjacencies.



We can **stretch** one rectangulation into another without changing adjacencies

Pattern avoidance

- In bijection with $\mathrm{Av}_n(3\underline{51}24 \wedge 3\underline{51}42 \wedge 24\underline{51}3 \wedge 42\underline{51}3)$

Pattern avoidance

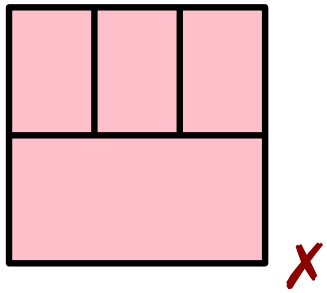
- In bijection with $\mathrm{Av}_n(3\underline{51}24 \wedge 3\underline{51}42 \wedge 24\underline{51}3 \wedge 42\underline{51}3)$
- Many interesting subfamilies of generic rectangulations.

Pattern avoidance

- In bijection with $\mathrm{Av}_n(3\underline{51}24 \wedge 3\underline{51}42 \wedge 24\underline{51}3 \wedge 42\underline{51}3)$
- Many interesting subfamilies of generic rectangulations.
- Many arise from forbidding rectangulation patterns:

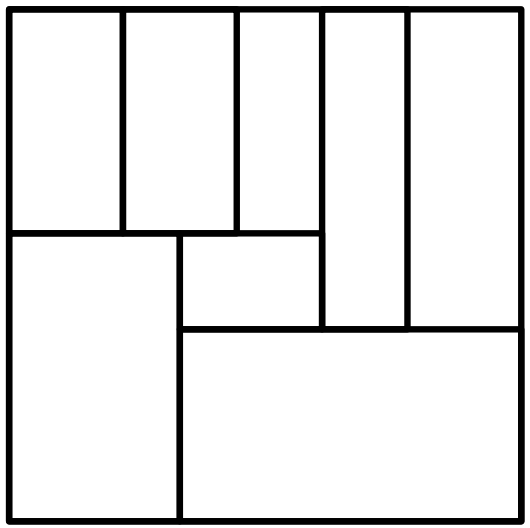
Pattern avoidance

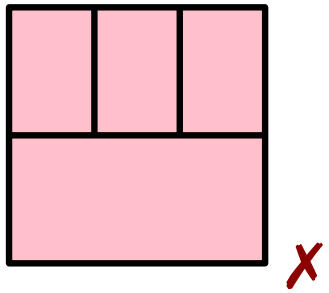
- In bijection with $\mathrm{Av}_n(3\underline{51}24 \wedge 3\underline{51}42 \wedge 24\underline{51}3 \wedge 42\underline{51}3)$
- Many interesting subfamilies of generic rectangulations.
- Many arise from forbidding rectangulation patterns:



Pattern avoidance

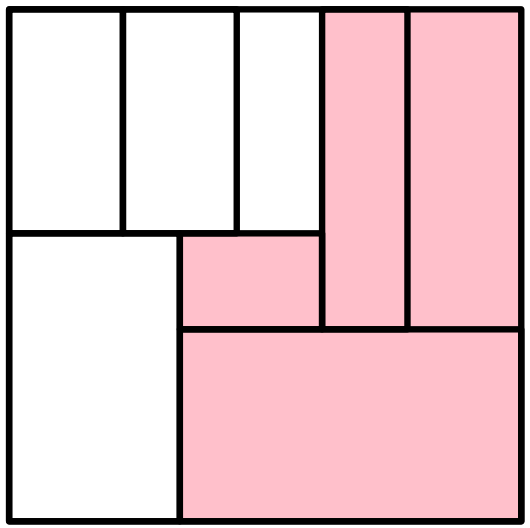
- In bijection with $\mathrm{Av}_n(3\underline{51}24 \wedge 3\underline{51}42 \wedge 24\underline{51}3 \wedge 42\underline{51}3)$
- Many interesting subfamilies of generic rectangulations.
- Many arise from forbidding rectangulation patterns:

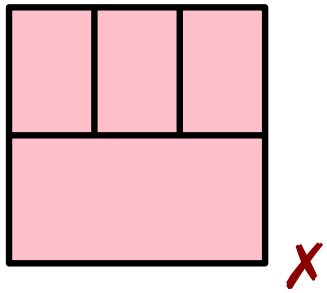




Pattern avoidance

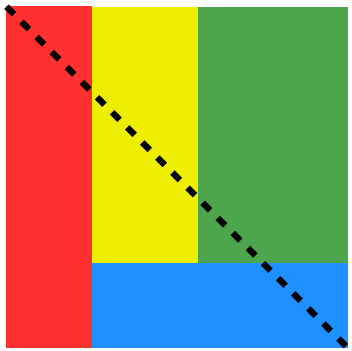
- In bijection with $\mathrm{Av}_n(3\underline{51}24 \wedge 3\underline{51}42 \wedge 24\underline{51}3 \wedge 42\underline{51}3)$
- Many interesting subfamilies of generic rectangulations.
- Many arise from forbidding rectangulation patterns:





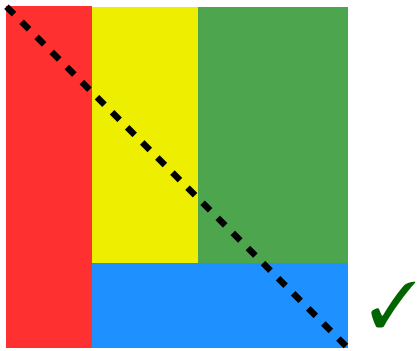
Diagonal rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06], [Law, Reading 12]

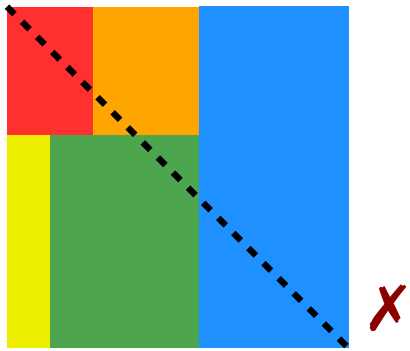
- Every rectangle intersects the main diagonal.



Diagonal rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06], [Law, Reading 12]

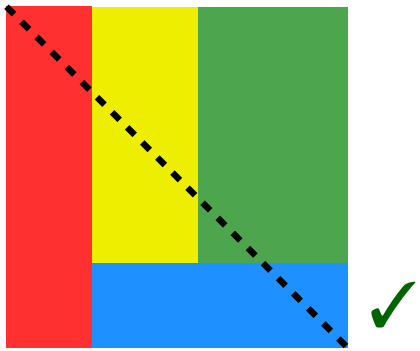
- Every rectangle intersects the main diagonal.

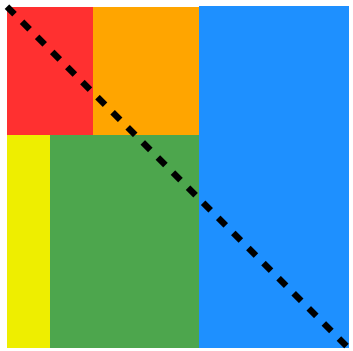




Diagonal rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06], [Law, Reading 12]

- Every rectangle intersects the main diagonal.

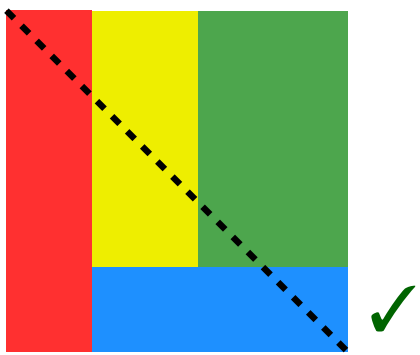
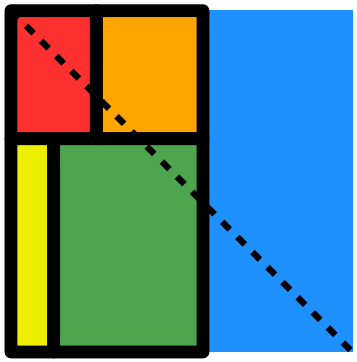

✓

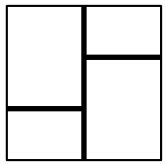
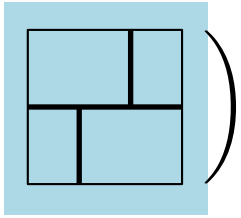

✗

- $\mathrm{Diag}_n = \mathrm{Av}_n\left(\quad \wedge \quad \right)$

Diagonal rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06], [Law, Reading 12]

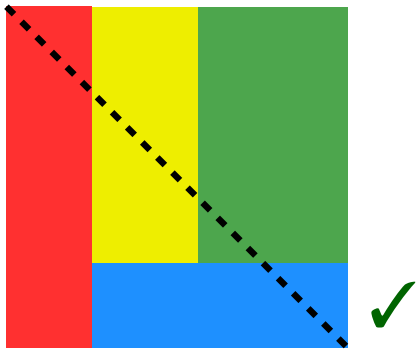
- Every rectangle intersects the main diagonal.

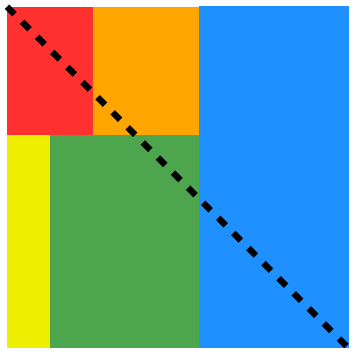



- $\mathrm{Diag}_n = \mathrm{Av}_n\Big($  $\wedge$  $\Big)$

Diagonal rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06], [Law, Reading 12]

- Every rectangle intersects the main diagonal.

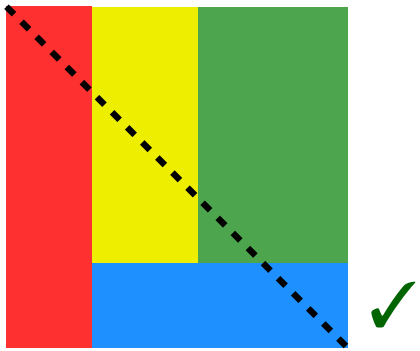

✓

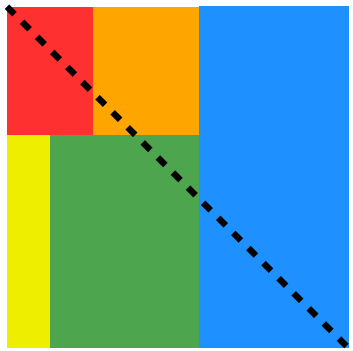

✗

- $\mathrm{Diag}_n = \mathrm{Av}_n\left(\quad \wedge \quad \right)$

Diagonal rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06], [Law, Reading 12]

- Every rectangle intersects the main diagonal.


✓

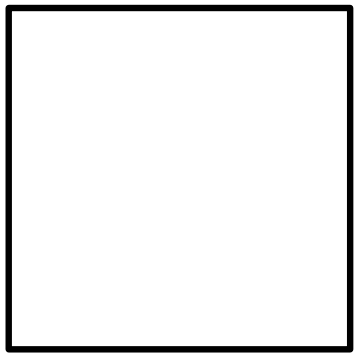

✗

- $\mathrm{Diag}_n = \mathrm{Av}_n\left(\quad \wedge \quad \right)$

- In bijection to $\mathrm{Av}(2\underline{41}3 \wedge 3\underline{14}2)$, $\mathrm{Av}(2\underline{41}3 \wedge 3\underline{41}2)$ and $\mathrm{Av}(2\underline{14}3 \wedge 3\underline{14}2)$.

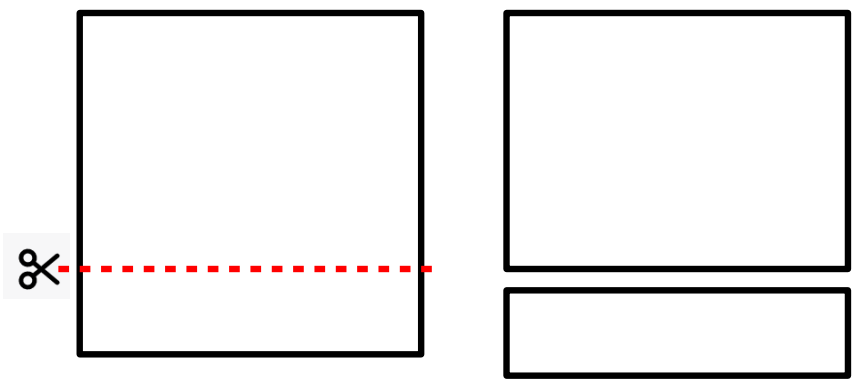
Guillotine rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06],[Asinowski et al. 13]

- Obtained by a sequence of *guillotine cuts*.



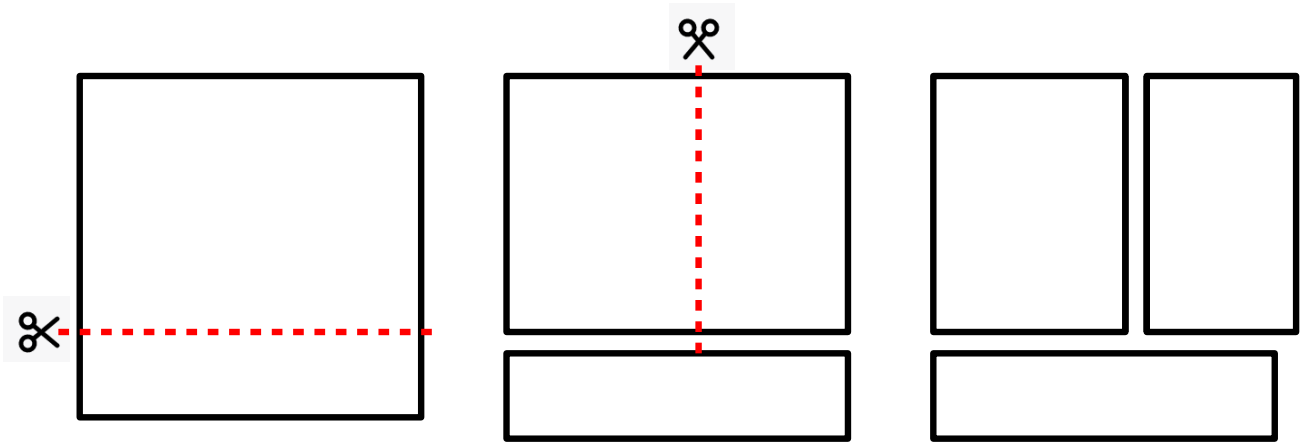
Guillotine rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06],[Asinowski et al. 13]

- Obtained by a sequence of *guillotine cuts*.



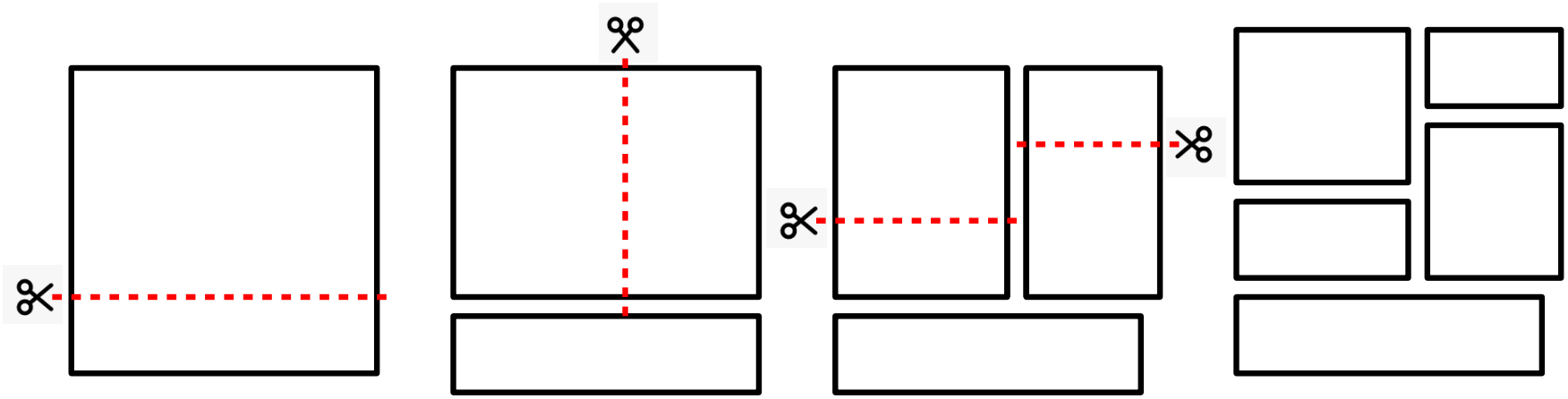
Guillotine rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06],[Asinowski et al. 13]

- Obtained by a sequence of *guillotine cuts*.



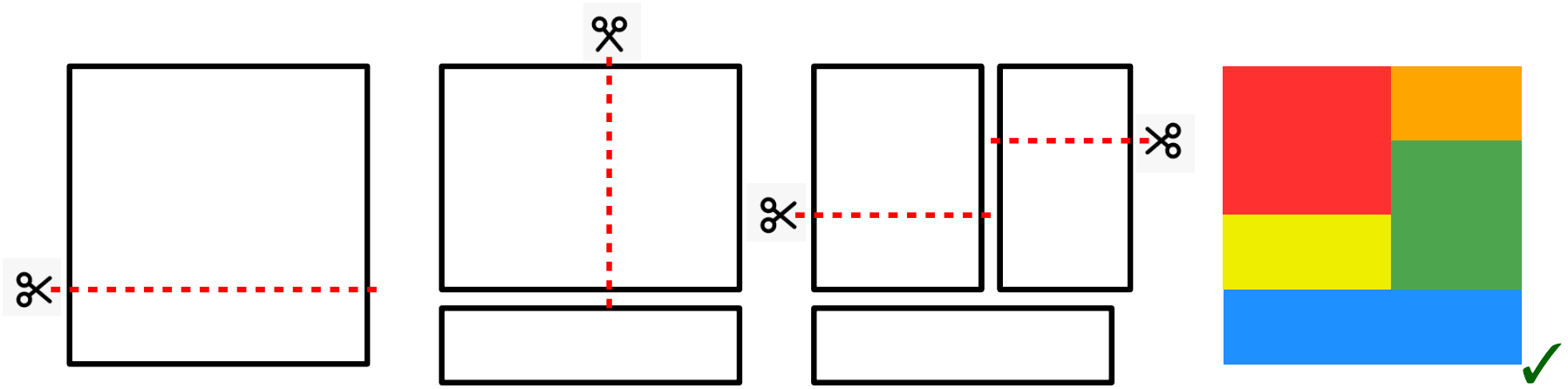
Guillotine rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06],[Asinowski et al. 13]

- Obtained by a sequence of *guillotine cuts*.



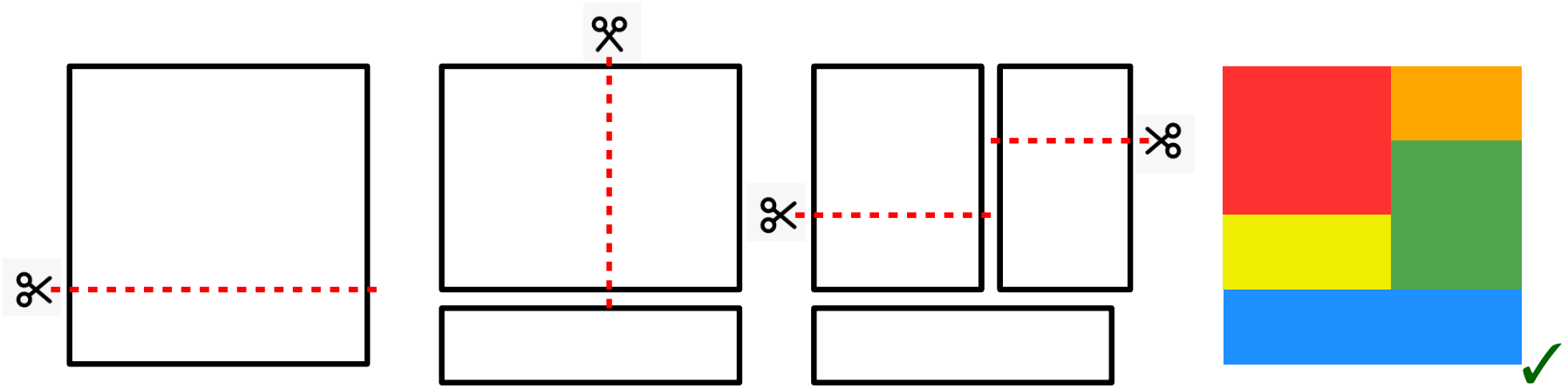
Guillotine rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06],[Asinowski et al. 13]

- Obtained by a sequence of *guillotine cuts*.



Guillotine rectangulations [Yao, Chen, Graham 03], [Ackerman, Barequet, Pinter 06],[Asinowski et al. 13]

- Obtained by a sequence of *guillotine cuts*.



- $\mathrm{Guill}_n = \mathrm{Av}_n\left(\quad \wedge \quad \right)$

Generation of rectangulations

- **Generation:** Output each object exactly once.

Generation of rectangulations

- **Generation:** Output each object exactly once.
- **This work:** Generation of many families of rectangulations.

Generation of rectangulations

- **Generation:** Output each object exactly once.
- **This work:** Generation of many families of rectangulations.
- **Objective:** Generate each new object as fast as possible.

Generation of rectangulations

- **Generation:** Output each object exactly once.
- **This work:** Generation of many families of rectangulations.
- **Objective:** Generate each new object as fast as possible.
  Ideally $\mathcal{O}(1)$ time between generated objects.

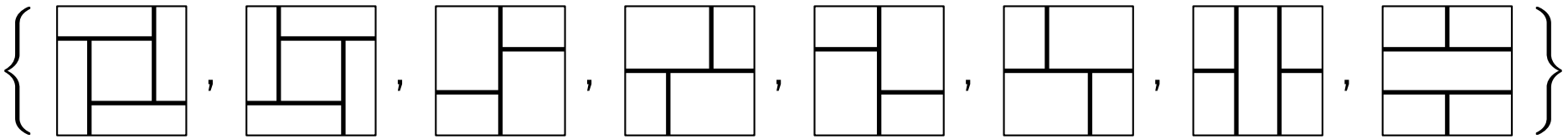
Generation of rectangulations

- **Generation:** Output each object exactly once.
- **This work:** Generation of many families of rectangulations.
- **Objective:** Generate each new object as fast as possible.
  Ideally $\mathcal{O}(1)$ time between generated objects.

delay

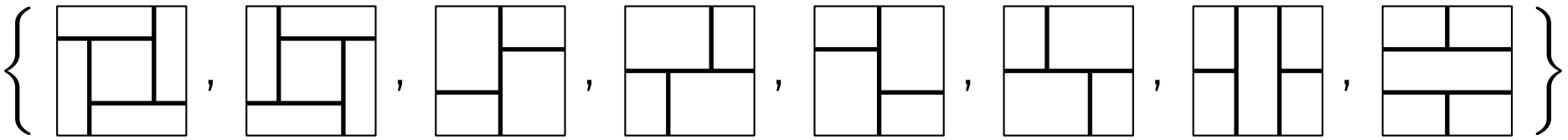
Main results

Efficient generation for any combination of the following forbidden patterns



Main results

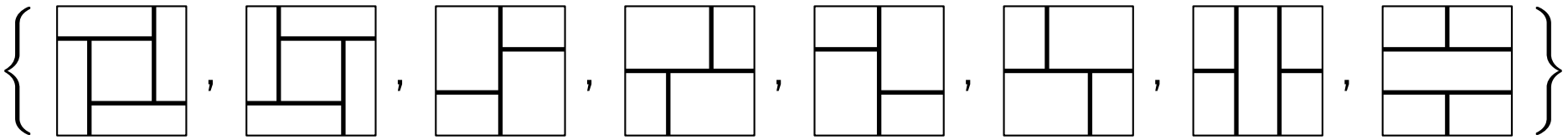
Efficient generation for any combination of the following forbidden patterns



Family	Delay

Main results

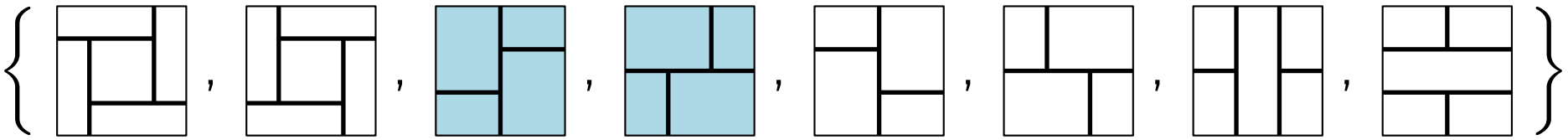
Efficient generation for any combination of the following forbidden patterns



Family	Delay
Generic	$\mathcal{O}(1)$ amortized delay

Main results

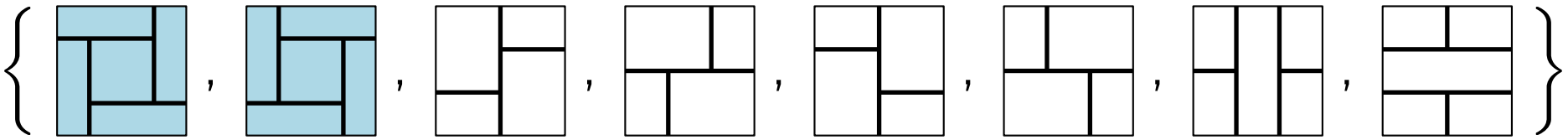
Efficient generation for any combination of the following forbidden patterns



Family	Delay
Generic	$\mathcal{O}(1)$ amortized delay
Diagonal	$\mathcal{O}(1)$ delay

Main results

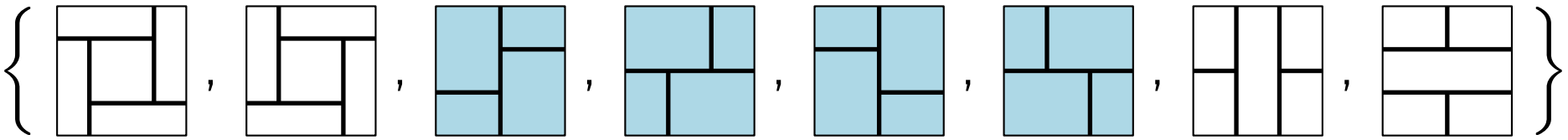
Efficient generation for any combination of the following forbidden patterns



Family	Delay
Generic	$\mathcal{O}(1)$ amortized delay
Diagonal	$\mathcal{O}(1)$ delay
Guillotine	$\mathcal{O}(n)$ delay

Main results

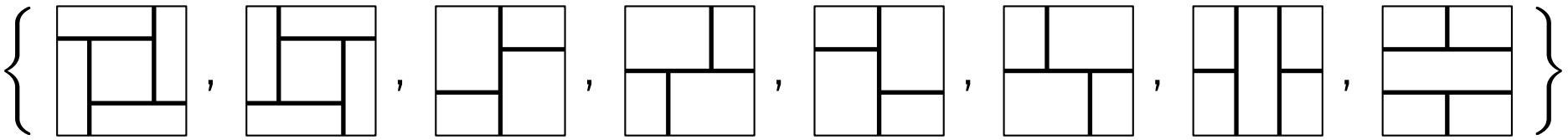
Efficient generation for any combination of the following forbidden patterns



Family	Delay
Generic	$\mathcal{O}(1)$ amortized delay
Diagonal	$\mathcal{O}(1)$ delay
Guillotine	$\mathcal{O}(n)$ delay
Area universal	$\mathcal{O}(n)$ delay

Main results

Efficient generation for any combination of the following forbidden patterns

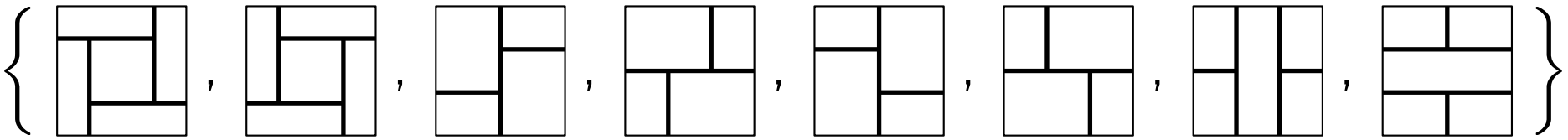


Family	Delay
Generic	$\mathcal{O}(1)$ amortized delay
Diagonal	$\mathcal{O}(1)$ delay
Guillotine	$\mathcal{O}(n)$ delay
Area universal	$\mathcal{O}(n)$ delay

+ generation framework for many other forbidden patterns.

Main results

Efficient generation for any combination of the following forbidden patterns



Family	Delay
Generic	$\mathcal{O}(1)$ amortized delay
Diagonal	$\mathcal{O}(1)$ delay
Guillotine	$\mathcal{O}(n)$ delay
Area universal	$\mathcal{O}(n)$ delay

+ generation framework for many other forbidden patterns.

C++ implementation on the Combinatorial Object Server:
`www.combos.org/rect`

Open questions

- Further study of pattern-avoiding rectangulations.

Open questions

- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.

Open questions

- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.

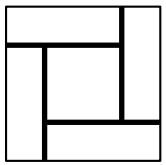
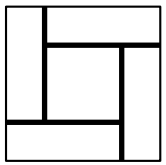
- Expressivity of pattern-avoiding rectangulations v/s permutations.

Open questions

- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.
- Expressivity of pattern-avoiding rectangulations v/s permutations.
  - v/s mesh patterns? v/s vincular?

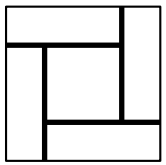
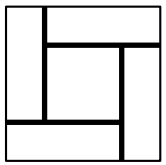
Open questions

- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.

- Expressivity of pattern-avoiding rectangulations v/s permutations.
  - v/s mesh patterns? v/s vincular?
  - $\mathsf{av}_n\left(\right.$  $\wedge$  $\left. \right) = ?$

Open questions

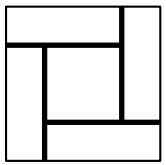
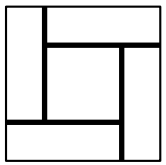
- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.

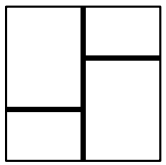
- Expressivity of pattern-avoiding rectangulations v/s permutations.
  - v/s mesh patterns? v/s vincular?
  - $\mathrm{av}_n\left(\right.$  $\wedge$  $\left.\right) = ?$

- Bijective conjectures:

Open questions

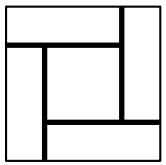
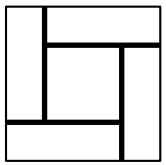
- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.

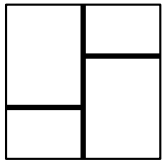
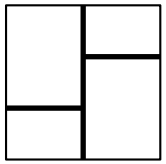
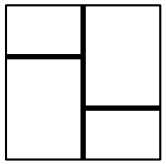
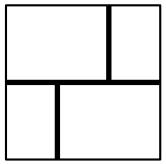
- Expressivity of pattern-avoiding rectangulations v/s permutations.
  - v/s mesh patterns? v/s vincular?
  - $\mathrm{av}_n\left(\right.$  $\wedge$  $\left.\right) = ?$

- Bijective conjectures:
  - $\mathrm{av}_n\left(\right.$  $\left.\right) = \mathrm{av}_n(21\overline{3}54)?$

Open questions

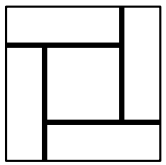
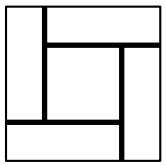
- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.

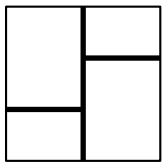
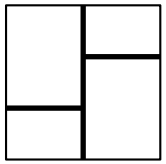
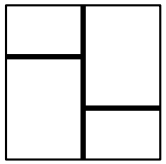
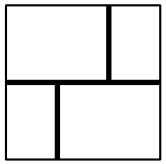
- Expressivity of pattern-avoiding rectangulations v/s permutations.
  - v/s mesh patterns? v/s vincular?
  - $\mathrm{av}_n\left(\right.$  $\wedge$  $\left.\right) = ?$

- Bijective conjectures:
  - $\mathrm{av}_n\left(\right.$  $\left.\right) = \mathrm{av}_n(21\overline{3}54)?$
  - $\mathrm{av}_n\left(\right.$  $\wedge$  $\wedge$  $\left.\right) = \mathrm{av}_n(2\underline{41}3 \wedge 3\underline{14}2 \wedge 3\underline{41}2)?$

Open questions

- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.

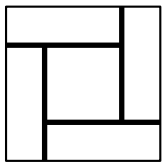
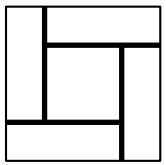
- Expressivity of pattern-avoiding rectangulations v/s permutations.
  - v/s mesh patterns? v/s vincular?
  - $\mathrm{av}_n\left(\right.$  $\wedge$  $\left.\right) = ?$

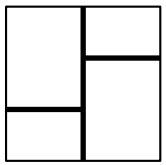
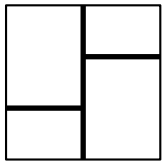
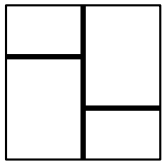
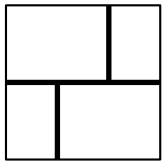
- Bijective conjectures:
  - $\mathrm{av}_n\left(\right.$  $\left.\right) = \mathrm{av}_n(21\overline{3}54)?$
  - $\mathrm{av}_n\left(\right.$  $\wedge$  $\wedge$  $\left.\right) = \mathrm{av}_n(2\underline{41}3 \wedge 3\underline{14}2 \wedge 3\underline{41}2)?$

  $+$ more

Thanks for your attention!

- Further study of pattern-avoiding rectangulations.
  - **cf.** pattern avoidance in binary trees.

- Expressivity of pattern-avoiding rectangulations v/s permutations.
  - v/s mesh patterns? v/s vincular?
  - $\mathsf{av}_n\left(\right.$  $\wedge$  $\left.\right) = ?$

- Bijective conjectures:
  - $\mathsf{av}_n\left(\right.$  $\left.\right) = \mathsf{av}_n(21\overline{3}54)$?
  - $\mathsf{av}_n\left(\right.$  $\wedge$  $\wedge$  $\left.\right) = \mathsf{av}_n(2\underline{41}3 \wedge 3\underline{14}2 \wedge 3\underline{41}2)$?

  $+$ more