

Studying permutation classes using the substitution decomposition

Mathilde Bouvel
(Institut für Mathematik, Universität Zürich)



**Universität
Zürich**^{UZH}

Combinatorics and interactions,
Introductory school at CIRM, Jan. 2017

Permutation patterns and permutation classes

Permutations

Permutation of size n = Bijection from $[1..n]$ to itself.

Set \mathfrak{S}_n , and $\mathfrak{S} = \bigcup_n \mathfrak{S}_n$.

Permutations

Permutation of size n = Bijection from $[1..n]$ to itself.

Set \mathfrak{S}_n , and $\mathfrak{S} = \bigcup_n \mathfrak{S}_n$.

- Two-line notation:

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 8 & 3 & 6 & 4 & 2 & 5 & 7 \end{pmatrix}$$

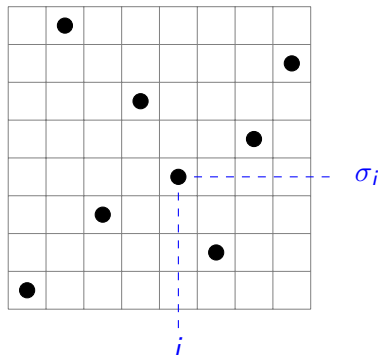
- One-line or **word** notation:

$$\sigma = 1 \ 8 \ 3 \ 6 \ 4 \ 2 \ 5 \ 7$$

- Description as a product of cycles:

$$\sigma = (1) (2 \ 8 \ 7 \ 5 \ 4 \ 6) (3)$$

- Graphical description, or **diagram**:



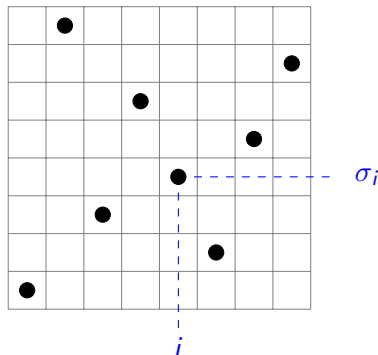
Permutations

Permutation of size n = Bijection from $[1..n]$ to itself.

Set \mathfrak{S}_n , and $\mathfrak{S} = \bigcup_n \mathfrak{S}_n$.

- Two-line notation:
$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 1 & 8 & 3 & 6 & 4 & 2 & 5 & 7 \end{pmatrix}$$
- One-line or **word** notation:
$$\sigma = 1 \ 8 \ 3 \ 6 \ 4 \ 2 \ 5 \ 7$$
- Description as a product of cycles:
$$\sigma = (1) (2 \ 8 \ 7 \ 5 \ 4 \ 6) (3)$$

- Graphical description, or **diagram**:

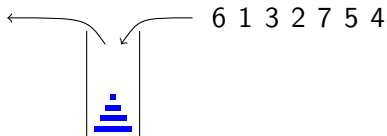


This talk is about **permutation patterns** and **permutation classes**.

The origin of permutation patterns: Stack sorting

The stack sorting operator S

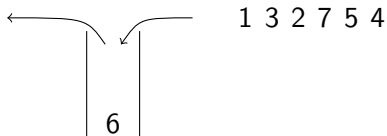
Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

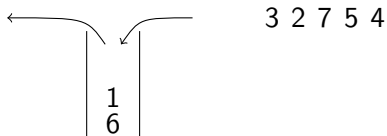
Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

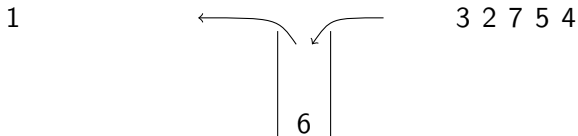
Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

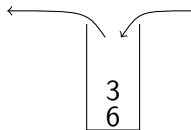


The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

1



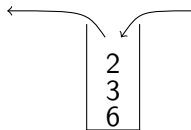
2 7 5 4

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

1



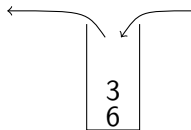
7 5 4

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

1 2



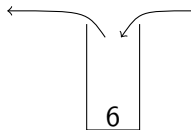
7 5 4

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

1 2 3



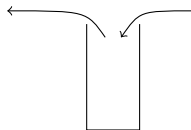
7 5 4

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

1 2 3 6



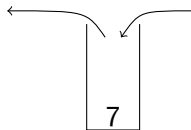
7 5 4

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

1 2 3 6



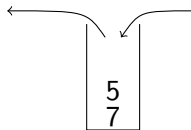
5 4

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

1 2 3 6



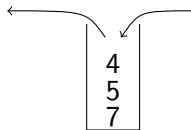
4

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

1 2 3 6

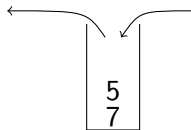


The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

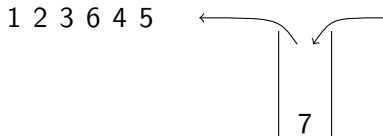
1 2 3 6 4



The origin of permutation patterns: Stack sorting

The stack sorting operator S

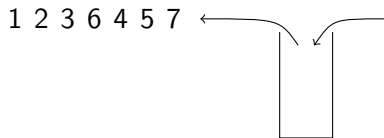
Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

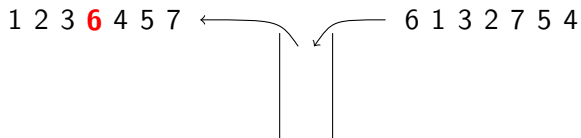
Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

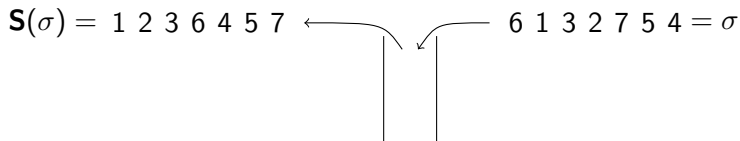
Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



Equivalently, $S(\varepsilon) = \varepsilon$ and $S(LnR) = S(L)S(R)n$, where $n = \max(LnR)$

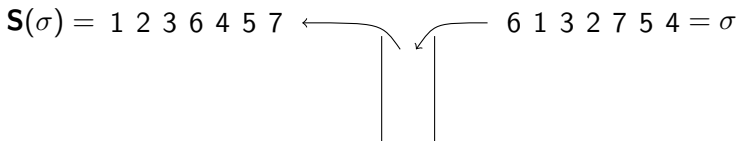
First result on permutation patterns [Knuth 68] :

A permutation σ is stack-sortable iff σ **avoids the pattern 231**

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



Equivalently, $S(\varepsilon) = \varepsilon$ and $S(LnR) = S(L)S(R)n$, where $n = \max(LnR)$

First result on permutation patterns [Knuth 68] :

A permutation σ is stack-sortable iff σ **avoids the pattern 231**

meaning that there are **no** $i < j < k$ such that $\sigma_k < \sigma_i < \sigma_j$,

or equivalently **no** subsequences $\cdots \sigma_i \cdots \sigma_j \cdots \sigma_k \cdots$ of σ whose elements are in the **same relative order** as 231.

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.

$$S(\sigma) = 1\ 2\ 3\ \mathbf{6}\ 4\ 5\ 7 \quad \leftarrow \quad \mathbf{6}\ 1\ 3\ 2\ \mathbf{7}\ 5\ \mathbf{4} = \sigma$$

Equivalently, $S(\varepsilon) = \varepsilon$ and $S(LnR) = S(L)S(R)n$, where $n = \max(LnR)$

First result on permutation patterns [Knuth 68] :

A permutation σ is stack-sortable iff σ **avoids the pattern 231**

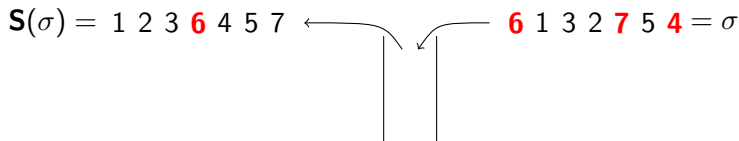
meaning that there are **no** $i < j < k$ such that $\sigma_k < \sigma_i < \sigma_j$,

or equivalently **no** subsequences $\cdots \sigma_i \cdots \sigma_j \cdots \sigma_k \cdots$ of σ whose elements are in the **same relative order** as 231.

The origin of permutation patterns: Stack sorting

The stack sorting operator S

Sort (or try to do so) using a **stack** satisfying the **Hanoi condition**.



Equivalently, $S(\varepsilon) = \varepsilon$ and $S(LnR) = S(L)S(R)n$, where $n = \max(LnR)$

First result on permutation patterns [Knuth 68] :

A permutation σ is stack-sortable iff σ **avoids the pattern 231**

meaning that there are **no** $i < j < k$ such that $\sigma_k < \sigma_i < \sigma_j$,

or equivalently **no** subsequences $\cdots \sigma_i \cdots \sigma_j \cdots \sigma_k \cdots$ of σ whose elements are in the **same relative order** as 231.

Next: other sorting devices and patterns [Even & Itai 71, Tarjan 72, Pratt 73]

Pattern relation \preceq :

$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists 1 \leq i_1 < \dots < i_k \leq n$ such that $\sigma_{i_1} \dots \sigma_{i_k}$ is in the **same relative order** (\equiv) as π .

Notation: $\pi \preceq \sigma$.

Permutation patterns

Pattern relation \preceq :

$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists 1 \leq i_1 < \dots < i_k \leq n$ such that $\sigma_{i_1} \dots \sigma_{i_k}$ is in the **same relative order** (\equiv) as π .

Notation: $\pi \preceq \sigma$.

Equivalently:

The **normalization** of $\sigma_{i_1} \dots \sigma_{i_k}$ on $[1..k]$ yields π .

Example: $2134 \preceq 312854796$
since $3157 \equiv 2134$.

Permutation patterns

Pattern relation \preceq :

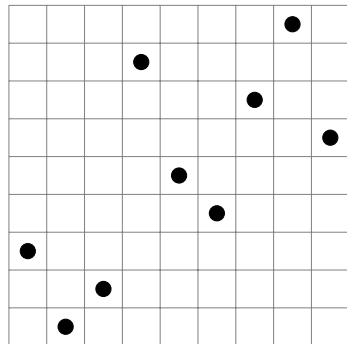
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists 1 \leq i_1 < \dots < i_k \leq n$ such that $\sigma_{i_1} \dots \sigma_{i_k}$ is in the **same relative order** (\equiv) as π .

Notation: $\pi \preceq \sigma$.

Equivalently:

The **normalization** of $\sigma_{i_1} \dots \sigma_{i_k}$ on $[1..k]$ yields π .

Example: $2134 \preceq 312854796$
since $3157 \equiv 2134$.



Permutation patterns

Pattern relation \preceq :

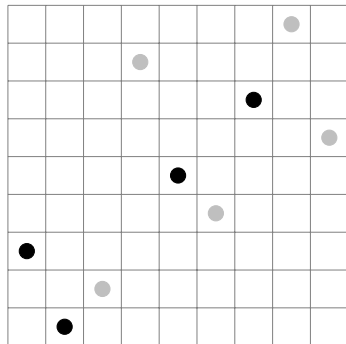
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists 1 \leq i_1 < \dots < i_k \leq n$ such that $\sigma_{i_1} \dots \sigma_{i_k}$ is in the **same relative order** (\equiv) as π .

Notation: $\pi \preceq \sigma$.

Equivalently:

The **normalization** of $\sigma_{i_1} \dots \sigma_{i_k}$ on $[1..k]$ yields π .

Example: $2134 \preceq 312854796$
since $3157 \equiv 2134$.



Permutation patterns

Pattern relation \preceq :

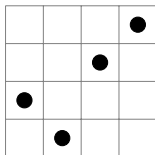
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists 1 \leq i_1 < \dots < i_k \leq n$ such that $\sigma_{i_1} \dots \sigma_{i_k}$ is in the **same relative order** (\equiv) as π .

Notation: $\pi \preceq \sigma$.

Equivalently:

The **normalization** of $\sigma_{i_1} \dots \sigma_{i_k}$ on $[1..k]$ yields π .

Example: $2134 \preceq 312854796$
since $3157 \equiv 2134$.



Permutation patterns

Pattern relation \preceq :

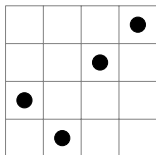
$\pi \in \mathfrak{S}_k$ is a pattern of $\sigma \in \mathfrak{S}_n$ if $\exists 1 \leq i_1 < \dots < i_k \leq n$ such that $\sigma_{i_1} \dots \sigma_{i_k}$ is in the **same relative order** (\equiv) as π .

Notation: $\pi \preceq \sigma$.

Equivalently:

The **normalization** of $\sigma_{i_1} \dots \sigma_{i_k}$ on $[1..k]$ yields π .

Example: $2134 \preceq 312854796$
since $3157 \equiv 2134$.



Crucial remark: \preceq is a **partial order** on \mathfrak{S} and “[\preceq] is even more interesting than the [sorting] networks we were characterizing” [Pratt 73].

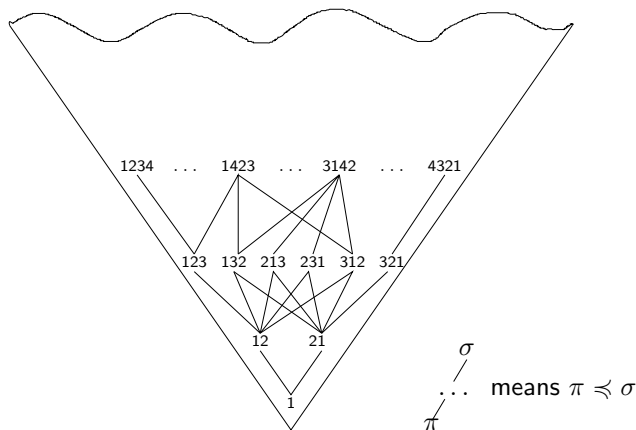
This is the key to defining permutation classes.

Permutation classes

- A **permutation class** is a set \mathcal{C} of permutations that is downward closed for \preceq , i.e. whenever $\pi \preceq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.

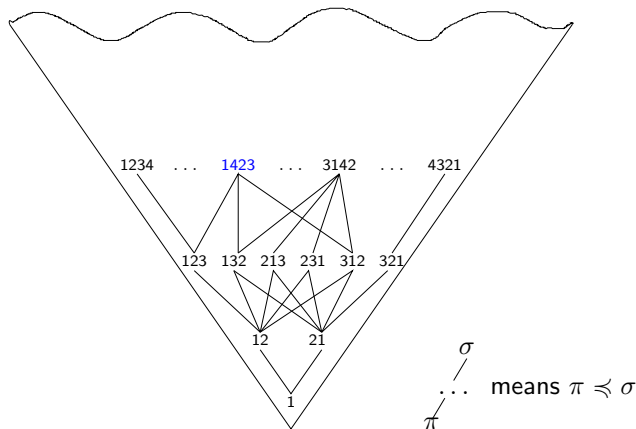
Permutation classes

- A **permutation class** is a set \mathcal{C} of permutations that is downward closed for \preceq , i.e. whenever $\pi \preceq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.



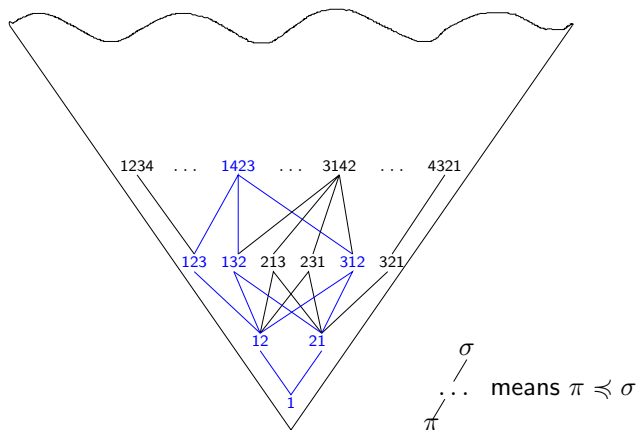
Permutation classes

- A **permutation class** is a set \mathcal{C} of permutations that is downward closed for \preceq , i.e. whenever $\pi \preceq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.



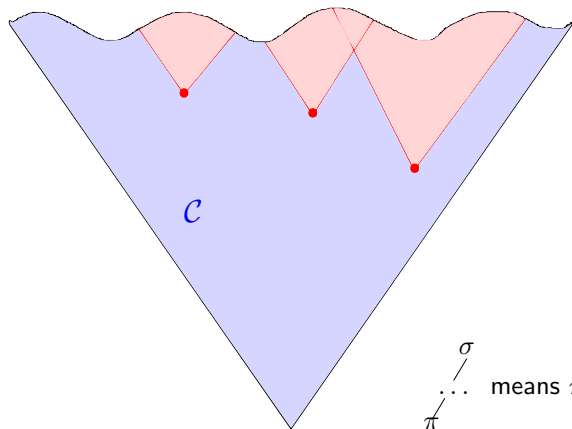
Permutation classes

- A **permutation class** is a set \mathcal{C} of permutations that is downward closed for \preceq , i.e. whenever $\pi \preceq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.



Permutation classes

- A **permutation class** is a set \mathcal{C} of permutations that is downward closed for \preceq , i.e. whenever $\pi \preceq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.



Permutation classes

- A **permutation class** is a set \mathcal{C} of permutations that is downward closed for \preceq , i.e. whenever $\pi \preceq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.
- **Notations:** $Av(\pi)$ = the set of permutations that avoid the pattern π
$$Av(B) = \bigcap_{\pi \in B} Av(\pi)$$
- **Fact:** For every permutation class \mathcal{C} , $\mathcal{C} = Av(B)$ for $B = \{\sigma \notin \mathcal{C} : \forall \pi \preceq \sigma \text{ such that } \pi \neq \sigma, \pi \in \mathcal{C}\}$.
 B is an antichain (set of elements incomparable for \preceq), called the **basis** of \mathcal{C} .

Permutation classes

- A **permutation class** is a set \mathcal{C} of permutations that is downward closed for \preceq , i.e. whenever $\pi \preceq \sigma$ and $\sigma \in \mathcal{C}$, then $\pi \in \mathcal{C}$.
- **Notations:** $Av(\pi)$ = the set of permutations that avoid the pattern π
$$Av(B) = \bigcap_{\pi \in B} Av(\pi)$$
- **Fact:** For every permutation class \mathcal{C} , $\mathcal{C} = Av(B)$ for $B = \{\sigma \notin \mathcal{C} : \forall \pi \preceq \sigma \text{ such that } \pi \neq \sigma, \pi \in \mathcal{C}\}$.
 B is an antichain (set of elements incomparable for \preceq), called the **basis** of \mathcal{C} .
- **Remarks:**
 - Conversely, every set $Av(B)$ is a permutation class.
 - There exist **infinite antichains** in the permutation pattern poset, hence some permutation classes have **infinite basis**.

A biased overview of important results

Specific enumeration results

For \mathcal{C} a permutation class, \mathcal{C}_n is the set of permutations of size n in \mathcal{C} and $C(z) = \sum_n |\mathcal{C}_n| z^n$ is its generating function.

Specific enumeration results

For \mathcal{C} a permutation class, \mathcal{C}_n is the set of permutations of size n in \mathcal{C} and $C(z) = \sum_n |\mathcal{C}_n| z^n$ is its generating function.

- One excluded pattern:
 - of size 3: By symmetry, focus on $Av(321)$ and $Av(231)$ only.
 - Description of $Av(321)$ [MacMahon 1915] and $Av(231)$ [Knuth 68].
 - Enumeration by the Catalan numbers in both cases.
 - Bijections: [Simion, Schmidt 85] [Claesson, Kitaev 08].
 - But these two classes have a very different structure.

Specific enumeration results

For \mathcal{C} a permutation class, \mathcal{C}_n is the set of permutations of size n in \mathcal{C} and $C(z) = \sum_n |\mathcal{C}_n| z^n$ is its generating function.

- One excluded pattern:
 - of size 3: By symmetry, focus on $Av(321)$ and $Av(231)$ only.
 - of size 4: Only three different enumerations. Representatives are:
 - $Av(1342)$ [Bóna 97], algebraic generating function
 - $Av(1234)$ [Gessel 90], holonomic (or D -finite) generating function
 - $Av(1324)$... remains an open problem

Specific enumeration results

For \mathcal{C} a permutation class, \mathcal{C}_n is the set of permutations of size n in \mathcal{C} and $C(z) = \sum_n |\mathcal{C}_n| z^n$ is its generating function.

- One excluded pattern:
 - of size 3: By symmetry, focus on $Av(321)$ and $Av(231)$ only.
 - of size 4: Only three different enumerations.
- Systematic enumeration of $Av(B)$ when B contains small excluded patterns (size 3 or 4).

Often combining general methods briefly discussed later.

[Simion&Schmidt, Gessel, Bóna, Gire, Guibert, Stankova, West... in the nineties]

[Albert, Atkinson, Brignall, Callan, Kremer, Pantone, Shiu, Vatter, ... nowadays]

Specific enumeration results

For \mathcal{C} a permutation class, \mathcal{C}_n is the set of permutations of size n in \mathcal{C} and $C(z) = \sum_n |\mathcal{C}_n| z^n$ is its generating function.

- One excluded pattern:
 - of size 3: By symmetry, focus on $Av(321)$ and $Av(231)$ only.
 - of size 4: Only three different enumerations.
- Systematic enumeration of $Av(B)$ when B contains small excluded patterns (size 3 or 4).

Often combining general methods briefly discussed later.

[Simion&Schmidt, Gessel, Bóna, Gire, Guibert, Stankova, West... in the nineties]

[Albert, Atkinson, Brignall, Callan, Kremer, Pantone, Shiu, Vatter, ... nowadays]

- Enumeration of classes (with more excluded patterns) appearing in a different context (e.g. indices of Schubert varieties [Albert, Brignall 13])

Growth rates of permutation classes

- Upper growth rate: $\overline{Gr}(\mathcal{C}) = \limsup_n \sqrt[n]{|\mathcal{C}_n|}$
- Lower growth rate: $\underline{Gr}(\mathcal{C}) = \liminf_n \sqrt[n]{|\mathcal{C}_n|}$

Marcus-Tardos theorem (2004, former Stanley-Wilf conjecture):

$\overline{Gr}(\mathcal{C}) < \infty$ for any class $\mathcal{C} \neq \mathfrak{S}$.

That is to say, permutation classes grow at most exponentially.

Growth rates of permutation classes

- Upper growth rate: $\overline{Gr}(\mathcal{C}) = \limsup_n \sqrt[n]{|\mathcal{C}_n|}$
- Lower growth rate: $\underline{Gr}(\mathcal{C}) = \liminf_n \sqrt[n]{|\mathcal{C}_n|}$

Marcus-Tardos theorem (2004, former Stanley-Wilf conjecture):

$\overline{Gr}(\mathcal{C}) < \infty$ for any class $\mathcal{C} \neq \mathfrak{S}$.

That is to say, permutation classes grow at most exponentially.

Conjecture: For any class \mathcal{C} , $\overline{Gr}(\mathcal{C}) = \underline{Gr}(\mathcal{C})$. Growth rate, denoted $Gr(\mathcal{C})$.

This holds for all principal classes, *i.e.*, $\mathcal{C} = Av(\pi)$,

and more generally for all sum-closed or skew-closed classes.

Growth rates of permutation classes

- Upper growth rate: $\overline{Gr}(\mathcal{C}) = \limsup_n \sqrt[n]{|\mathcal{C}_n|}$
- Lower growth rate: $\underline{Gr}(\mathcal{C}) = \liminf_n \sqrt[n]{|\mathcal{C}_n|}$

Marcus-Tardos theorem (2004, former Stanley-Wilf conjecture):

$\overline{Gr}(\mathcal{C}) < \infty$ for any class $\mathcal{C} \neq \mathfrak{S}$.

That is to say, permutation classes grow at most exponentially.

Conjecture: For any class \mathcal{C} , $\overline{Gr}(\mathcal{C}) = \underline{Gr}(\mathcal{C})$. Growth rate, denoted $Gr(\mathcal{C})$.

Arratia's (false) conjecture:

$Gr(Av(\pi)) \leq (k-1)^2 = Gr(Av(k \dots 21))$ for $|\pi| = k$

- $Gr(Av(1324)) > 9.47$ [Albert, Elder, Reznitzer, Westcott, Zabrocki 06]
- Remark: $Gr(Av(1324))$ is > 9.81 [Bevan 15], < 13.74 [Bóna 15] and conjectured to be ≈ 11.60 [Conway, Guttmann 15]
- $Gr(Av(\pi))$ is typically exponential in $|\pi|$ [Fox, 2017+]

Growth rates of permutation classes

- Upper growth rate: $\overline{Gr}(\mathcal{C}) = \limsup_n \sqrt[n]{|\mathcal{C}_n|}$
- Lower growth rate: $\underline{Gr}(\mathcal{C}) = \liminf_n \sqrt[n]{|\mathcal{C}_n|}$

Marcus-Tardos theorem (2004, former Stanley-Wilf conjecture):

$$\overline{Gr}(\mathcal{C}) < \infty \text{ for any class } \mathcal{C} \neq \mathfrak{S}.$$

That is to say, permutation classes grow at most exponentially.

Conjecture: For any class \mathcal{C} , $\overline{Gr}(\mathcal{C}) = \underline{Gr}(\mathcal{C})$. Growth rate, denoted $Gr(\mathcal{C})$.

Arratia's (false) conjecture:

$$Gr(Av(\pi)) \leq (k-1)^2 = Gr(Av(k \dots 21)) \text{ for } |\pi| = k$$

Classification of growth rates:

Exactly which numbers can occur as (upper) growth rates is known, except between $\xi \approx 2.305$ and $\lambda < 2.36$ [Vatter and collaborators].

- Before ξ : countably many growth rates, all characterized
- After λ : all real numbers

Nature of the generating functions of permutation classes

A *variety of behaviors* can occur: rational, algebraic, D-finite, non D-finite.

- For $Av(231)$ and $Av(321)$: Catalan numbers, *algebraic* GF. But:
 - All proper subclasses of $Av(231)$ are *rational* [Albert, Atkinson 05].
 - $Av(321)$ contains *non D-finite* subclasses.
 - However, every proper subclass of $Av(321)$ which has finite basis or is *wqo* is rational [Albert, Brignall, Ruškuc, Vatter 2017+].

Nature of the generating functions of permutation classes

- A **variety of behaviors** can occur: rational, algebraic, D-finite, non D-finite.
- For $Av(231)$ and $Av(321)$: Catalan numbers, **algebraic** GF. But:
 - All proper subclasses of $Av(231)$ are **rational** [Albert, Atkinson 05].
 - $Av(321)$ contains **non D-finite** subclasses.
 - However, every proper subclass of $Av(321)$ which has finite basis or is **wqo** is rational [Albert, Brignall, Ruškuc, Vatter 2017+].
 - (Tight?) connection between **wqo and nice GF**:
 - A class is wqo (well quasi-ordered) if it contains no infinite antichains.
 - If a class \mathcal{C} and all its subclasses are algebraic, then \mathcal{C} is wqo.
 - Vatter conjectures that the converse holds.

Nature of the generating functions of permutation classes

- A **variety of behaviors** can occur: rational, algebraic, D-finite, non D-finite.
- For $Av(231)$ and $Av(321)$: Catalan numbers, **algebraic** GF. But:
 - All proper subclasses of $Av(231)$ are **rational** [Albert, Atkinson 05].
 - $Av(321)$ contains **non D-finite** subclasses.
 - However, every proper subclass of $Av(321)$ which has finite basis or is **wqo** is rational [Albert, Brignall, Ruškuc, Vatter 2017+].
 - (Tight?) connection between **wqo and nice GF**:
 - A class is wqo (well quasi-ordered) if it contains no infinite antichains.
 - If a class \mathcal{C} and all its subclasses are algebraic, then \mathcal{C} is wqo.
 - Vatter conjectures that the converse holds.
 - Sufficient algebraicity condition [Albert, Atkinson 05]:
When a class contains finitely many **simple** permutations.

A probabilistic look at permutation classes

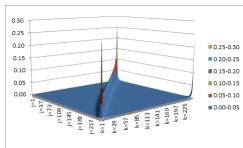
Typical diagrams of [large permutations in classes](#): what do they look like?

A probabilistic look at permutation classes

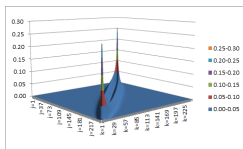
Typical diagrams of **large permutation classes**: what do they look like?

- One excluded pattern of size 3:

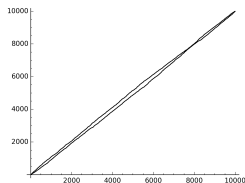
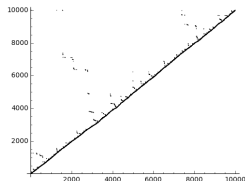
$Av(231)$



$Av(321)$



[Miner, Pak 14]



[Hoffman, Rizzolo, Slivken 16]

A probabilistic look at permutation classes

Typical diagrams of **large permutations in classes**: what do they look like?

- One excluded pattern of size 3:
 - **Precise** local description of the asymptotic shape [Miner, Pak 14] [Madras and collaborators].
 - **Scaling limits** and link with the **Brownian excursion** (for the fluctuations around the main diagonal) [Hoffman, Rizzolo, Slivken 16].
 - For any pattern π , the following quantity converges in distribution to a strictly positive random variable [Janson 16]:

$$\frac{\text{number of occurrences of } \pi \text{ in uniform } \sigma \in Av_n(132)}{n(|\pi| + \text{number of descents of } \pi + 1)/2}.$$

A probabilistic look at permutation classes

Typical diagrams of **large permutations in classes**: what do they look like?

- One excluded pattern of size 3:
 - **Precise** local description of the asymptotic shape [Miner, Pak 14] [Madras and collaborators].
 - **Scaling limits** and link with the **Brownian excursion** (for the fluctuations around the main diagonal) [Hoffman, Rizzolo, Slivken 16].
 - For any pattern π , the following quantity converges in distribution to a strictly positive random variable [Janson 16]:

$$\frac{\text{number of occurrences of } \pi \text{ in uniform } \sigma \in Av_n(132)}{n^{(|\pi| + \text{number of descents of } \pi + 1))/2}}.$$

- Other known cases:
 - Connected monotone grid classes (deterministic limit) [Bevan 15]
 - Separable permutations (non-deterministic limit) [Bassino, B., Féray, Gerin, Pierrot 2017+]

Some general methods

To prove general results on families of permutation classes (e.g. growth rates, nature of GF), some **general methods** are often used, which each capture a notion of **nice structure** of permutations in these classes:

Some general methods

To prove general results on families of permutation classes (e.g. growth rates, nature of GF), some **general methods** are often used, which each capture a notion of **nice structure** of permutations in these classes:

- Generating trees
- **Substitution decomposition**
- Merging and splitting
- (Geometric) grid classes
- Encodings by words over a finite alphabet
- ...

Some general methods

To prove general results on families of permutation classes (e.g. growth rates, nature of GF), some **general methods** are often used, which each capture a notion of **nice structure** of permutations in these classes:

- Generating trees
- **Substitution decomposition**
- Merging and splitting
- (Geometric) grid classes
- Encodings by words over a finite alphabet
- ...

These methods are also sometimes used to prove results about (or enumerate) specific classes.

Substitution decomposition

Substitution for permutations

Substitution is an operation building a permutation from smaller ones.

Notation for **substitution** (or **inflation**): $\sigma = \pi[\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)}]$

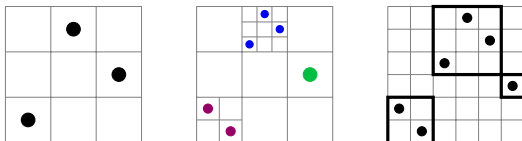
with $k = \text{size of } \pi$.

Substitution for permutations

Substitution is an operation building a permutation from smaller ones.

Notation for **substitution** (or **inflation**): $\sigma = \pi[\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)}]$
with $k = \text{size of } \pi$.

Example: Here, $\pi = 132$, and

$$\left\{ \begin{array}{l} \alpha^{(1)} = 21 = \begin{array}{|c|c|} \hline \bullet & \\ \hline & \bullet \\ \hline \end{array} \\ \alpha^{(2)} = 132 = \begin{array}{|c|c|c|} \hline & \bullet & \\ \hline & & \bullet \\ \hline \bullet & & \\ \hline \end{array} \\ \alpha^{(3)} = 1 = \begin{array}{|c|} \hline \bullet \\ \hline \end{array} \end{array} \right. .$$


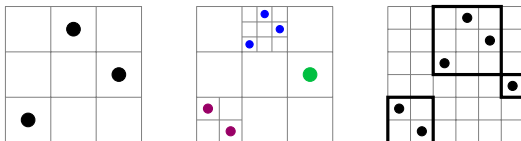
Hence $\sigma = 132[21, 132, 1] = 214653$.

Substitution for permutations

Substitution is an operation building a permutation from smaller ones.

Notation for **substitution** (or **inflation**): $\sigma = \pi[\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)}]$
with $k = \text{size of } \pi$.

Example: Here, $\pi = 132$, and

$$\left\{ \begin{array}{l} \alpha^{(1)} = 21 = \begin{array}{|c|c|} \hline \bullet & \\ \hline & \bullet \\ \hline \end{array} \\ \alpha^{(2)} = 132 = \begin{array}{|c|c|c|} \hline & \bullet & \\ \hline & & \bullet \\ \hline \bullet & & \\ \hline \end{array} \\ \alpha^{(3)} = 1 = \begin{array}{|c|} \hline \bullet \\ \hline \end{array} \end{array} \right. .$$


Hence $\sigma = 132[21, 132, 1] = 214653$.

In general, **many** substitutions give σ , but we will see a canonical one.

Simple permutations

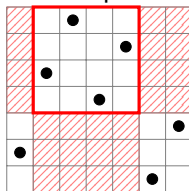
Interval (or **block**) = set of elements of σ whose positions **and** values form intervals of integers

Example: 5746 is an interval of 2 **5746** 13

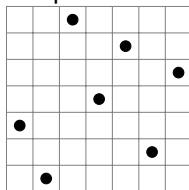
Simple permutation = permutation with no interval, except the trivial ones: $1, 2, \dots, n$ and σ

Example: 3174625 is simple

Not simple:



Simple:



Simple permutations

Interval (or **block**) = set of elements of σ whose positions **and** values form intervals of integers

Example: 5746 is an interval of 2 **5746** 13

Simple permutation = permutation with no interval, except the trivial ones: $1, 2, \dots, n$ and σ

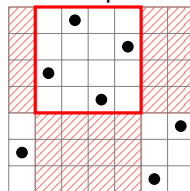
Example: 3174625 is simple

The smallest simple permutations:

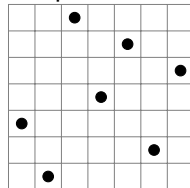
12, 21, 2413, 3142, 6 of size 5, ...

But: For us, it is convenient to consider that 12 and 21 are **not** simple permutations.

Not simple:



Simple:



Simple permutations

Interval (or **block**) = set of elements of σ whose positions **and** values form intervals of integers

Example: 5746 is an interval of 2 **5746** 13

Simple permutation = permutation with no interval, except the trivial ones: $1, 2, \dots, n$ and σ

Example: 3174625 is simple

The smallest simple permutations:

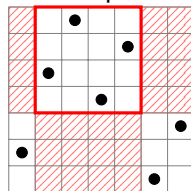
12, 21, 2413, 3142, 6 of size 5, ...

But: For us, it is convenient to consider that 12 and 21 are **not** simple permutations.

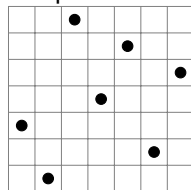
Remark: Enumeration of simple permutations:

- Generating function is not D-finite
- Asymptotically $\frac{n!}{e^2}$ of size n [Albert, Atkinson, Klazar 03]

Not simple:



Simple:



Substitution decomposition theorem for permutations

Notation:

- \oplus represents any permutation $12 \dots k$ for $k \geq 2$
- \ominus represents any permutation $k \dots 21$ for $k \geq 2$
- \oplus -indecomposable: that cannot be written as $\oplus[\beta^{(1)}, \beta^{(2)}]$
- \ominus -indecomposable: that cannot be written as $\ominus[\beta^{(1)}, \beta^{(2)}]$

Substitution decomposition theorem for permutations

Notation:

- \oplus represents any permutation $12 \dots k$ for $k \geq 2$
- \ominus represents any permutation $k \dots 21$ for $k \geq 2$
- \oplus -indecomposable: that cannot be written as $\oplus[\beta^{(1)}, \beta^{(2)}]$
- \ominus -indecomposable: that cannot be written as $\ominus[\beta^{(1)}, \beta^{(2)}]$

Theorem: [Albert, Atkinson, Klazar 03]

Every $\sigma (\neq 1)$ is **uniquely** decomposed as

- $\oplus[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where the $\alpha^{(i)}$ are \oplus -indecomposable
- $\ominus[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where the $\alpha^{(i)}$ are \ominus -indecomposable
- $\pi[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where π is simple of size $k \geq 4$

Proof idea: The $\alpha^{(i)}$ represent the maximal proper intervals of σ .

Substitution decomposition theorem for permutations

Notation:

- \oplus represents any permutation $12 \dots k$ for $k \geq 2$
- \ominus represents any permutation $k \dots 21$ for $k \geq 2$
- \oplus -indecomposable: that cannot be written as $\oplus[\beta^{(1)}, \beta^{(2)}]$
- \ominus -indecomposable: that cannot be written as $\ominus[\beta^{(1)}, \beta^{(2)}]$

Theorem: [Albert, Atkinson, Klazar 03]

Every $\sigma (\neq 1)$ is **uniquely** decomposed as

- $\oplus[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where the $\alpha^{(i)}$ are \oplus -indecomposable
- $\ominus[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where the $\alpha^{(i)}$ are \ominus -indecomposable
- $\pi[\alpha^{(1)}, \dots, \alpha^{(k)}]$, where π is simple of size $k \geq 4$

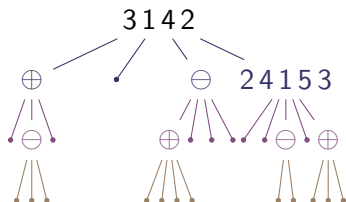
Proof idea: The $\alpha^{(i)}$ represent the maximal proper intervals of σ .

Decomposing recursively inside the $\alpha^{(i)} \Rightarrow$ **decomposition tree**

Decomposition tree

Example: Decomposition tree of

$\sigma = 10\ 13\ 12\ 11\ 14\ 1\ 18\ 19\ 20\ 21\ 17\ 16\ 15\ 4\ 8\ 3\ 2\ 9\ 5\ 6\ 7$



$\sigma = 3\ 1\ 4\ 2[\oplus[1, \ominus[1, 1, 1], 1], 1, \ominus[\oplus[1, 1, 1, 1], 1, 1, 1], 2\ 4\ 1\ 5\ 3[1, 1, \ominus[1, 1], 1, \oplus[1, 1, 1]]]$

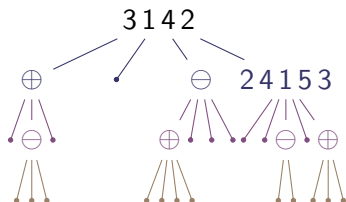
Notation and properties:

- Nodes labeled by \oplus , \ominus or π simple of size ≥ 4 .
- No edge $\oplus - \oplus$ nor $\ominus - \ominus$.
- **Rooted ordered** trees.
- These conditions **characterize** decomposition trees.

Decomposition tree

Example: Decomposition tree of

$$\sigma = 10\ 13\ 12\ 11\ 14\ 1\ 18\ 19\ 20\ 21\ 17\ 16\ 15\ 4\ 8\ 3\ 2\ 9\ 5\ 6\ 7$$



$$\sigma = 3\ 1\ 4\ 2[\oplus[1, \ominus[1, 1, 1], 1], 1, \ominus[\oplus[1, 1, 1, 1], 1, 1, 1], 2\ 4\ 1\ 5\ 3[1, 1, \ominus[1, 1], 1, \oplus[1, 1, 1]]]$$

The substitution decomposition theorem provides a **bijection** between permutations of size n and decomposition trees with n leaves.

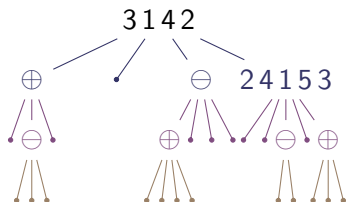
Notation and properties:

- Nodes labeled by \oplus , \ominus or π simple of size ≥ 4 .
- No edge $\oplus - \oplus$ nor $\ominus - \ominus$.
- **Rooted ordered** trees.
- These conditions **characterize** decomposition trees.

Decomposition tree

Example: Decomposition tree of

$\sigma = 10\ 13\ 12\ 11\ 14\ 1\ 18\ 19\ 20\ 21\ 17\ 16\ 15\ 4\ 8\ 3\ 2\ 9\ 5\ 6\ 7$



$\sigma = 3142[\oplus[1, \ominus[1, 1, 1], 1], 1, \ominus[\oplus[1, 1, 1, 1], 1, 1, 1], 24153[1, 1, \ominus[1, 1], 1, \oplus[1, 1, 1]]]$

The substitution decomposition theorem provides a **bijection** between permutations of size n and decomposition trees with n leaves.

Very convenient, since “trees are the prototypical recursive structure” [Flajolet, Sedgewick 09]

Notation and properties:

- Nodes labeled by \oplus , \ominus or π simple of size ≥ 4 .
- No edge $\oplus - \oplus$ nor $\ominus - \ominus$.
- Rooted ordered trees.
- These conditions characterize decomposition trees.

A tree grammar for permutations

With \mathcal{S} the set of simple permutations, the substitution decomposition theorem says:

$$\left\{ \begin{array}{l} \mathfrak{S} = \bullet + \begin{array}{c} \oplus \\ \diagup \quad \diagdown \\ \mathfrak{S}^+ \quad \dots \quad \mathfrak{S}^+ \end{array} + \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \mathfrak{S}^- \quad \dots \quad \mathfrak{S}^- \end{array} + \sum_{\pi \in \mathcal{S}} \begin{array}{c} \pi \\ \diagup \quad \diagdown \quad \diagup \quad \diagdown \\ \mathfrak{S} \quad \mathfrak{S} \quad \dots \quad \mathfrak{S} \end{array} \\ \mathfrak{S}^+ = \bullet + \begin{array}{c} \ominus \\ \diagup \quad \diagdown \\ \mathfrak{S}^- \quad \dots \quad \mathfrak{S}^- \end{array} + \sum_{\pi \in \mathcal{S}} \begin{array}{c} \pi \\ \diagup \quad \diagdown \quad \diagup \quad \diagdown \\ \mathfrak{S} \quad \mathfrak{S} \quad \dots \quad \mathfrak{S} \end{array} \\ \mathfrak{S}^- = \bullet + \begin{array}{c} \oplus \\ \diagup \quad \diagdown \\ \mathfrak{S}^+ \quad \dots \quad \mathfrak{S}^+ \end{array} + \sum_{\pi \in \mathcal{S}} \begin{array}{c} \pi \\ \diagup \quad \diagdown \quad \diagup \quad \diagdown \\ \mathfrak{S} \quad \mathfrak{S} \quad \dots \quad \mathfrak{S} \end{array} \end{array} \right.$$

A tree grammar for permutations

With \mathcal{S} the set of simple permutations, the substitution decomposition theorem says:

$$\left\{ \begin{array}{l} \mathfrak{S} = \bullet + \mathfrak{S}^+ \oplus \mathfrak{S}^+ + \mathfrak{S}^- \ominus \mathfrak{S}^- + \sum_{\pi \in \mathcal{S}} \pi \mathfrak{S} \mathfrak{S} \dots \mathfrak{S} \\ \mathfrak{S}^+ = \bullet + \mathfrak{S}^- \ominus \mathfrak{S}^- + \sum_{\pi \in \mathcal{S}} \pi \mathfrak{S} \mathfrak{S} \dots \mathfrak{S} \\ \mathfrak{S}^- = \bullet + \mathfrak{S}^+ \oplus \mathfrak{S}^+ + \sum_{\pi \in \mathcal{S}} \pi \mathfrak{S} \mathfrak{S} \dots \mathfrak{S} \end{array} \right.$$

Can we specialize this tree grammar to subsets of \mathfrak{S} , and in particular to permutation classes $\mathcal{C} = Av(B)$?

Can we do it automatically? even algorithmically?

What kind of results can be obtained from such a tree grammar describing a permutation class \mathcal{C} ?

**Some (general) results obtained
using substitution decomposition**

How it all started

- **Theorem [Albert, Atkinson 05]:** For any permutation class \mathcal{C} , if \mathcal{C} contains finitely many simple permutations, then \mathcal{C} has a finite basis and an algebraic generating function $C(z)$.

How it all started

- **Theorem [Albert, Atkinson 05]:** For any permutation class \mathcal{C} , if \mathcal{C} contains finitely many simple permutations, then \mathcal{C} has a finite basis and an algebraic generating function $C(z)$.
- **Constructive proof** (of the GF part of the theorem):
 - Propagate avoidance constraints in

$$\begin{cases} \mathfrak{S} = \bullet + \mathfrak{S}^+ \overset{\oplus}{\dots} \mathfrak{S}^+ + \mathfrak{S}^- \overset{\ominus}{\dots} \mathfrak{S}^- + \sum_{\pi \in \mathcal{S}} \mathfrak{S} \overset{\pi}{\dots} \mathfrak{S} \\ \mathfrak{S}^+ = \bullet + \mathfrak{S}^- \overset{\ominus}{\dots} \mathfrak{S}^- + \sum_{\pi \in \mathcal{S}} \mathfrak{S} \overset{\pi}{\dots} \mathfrak{S} \\ \mathfrak{S}^- = \bullet + \mathfrak{S}^+ \overset{\oplus}{\dots} \mathfrak{S}^+ + \sum_{\pi \in \mathcal{S}} \mathfrak{S} \overset{\pi}{\dots} \mathfrak{S} \end{cases}$$

- Obtain a (possibly ambiguous) context-free tree grammar for \mathcal{C} .
- Inclusion-exclusion gives a polynomial system for $C(z)$.

How it all started

- **Theorem [Albert, Atkinson 05]:** For any permutation class \mathcal{C} , if \mathcal{C} contains finitely many simple permutations, then \mathcal{C} has a finite basis and an algebraic generating function $C(z)$.

- **Constructive proof** (of the *GF* part of the theorem):

- Propagate avoidance constraints in

$$\left\{ \begin{array}{l} \mathfrak{S} = \bullet + \mathfrak{S}^+ \oplus \mathfrak{S}^+ + \mathfrak{S}^- \ominus \mathfrak{S}^- + \sum_{\pi \in \mathcal{S}} \mathfrak{S} \begin{array}{c} \pi \\ \diagdown \quad \diagup \\ \mathfrak{S} \quad \mathfrak{S} \quad \dots \quad \mathfrak{S} \end{array} \\ \mathfrak{S}^+ = \bullet + \mathfrak{S}^- \ominus \mathfrak{S}^- + \sum_{\pi \in \mathcal{S}} \mathfrak{S} \begin{array}{c} \pi \\ \diagdown \quad \diagup \\ \mathfrak{S} \quad \mathfrak{S} \quad \dots \quad \mathfrak{S} \end{array} \\ \mathfrak{S}^- = \bullet + \mathfrak{S}^+ \oplus \mathfrak{S}^+ + \sum_{\pi \in \mathcal{S}} \mathfrak{S} \begin{array}{c} \pi \\ \diagdown \quad \diagup \\ \mathfrak{S} \quad \mathfrak{S} \quad \dots \quad \mathfrak{S} \end{array} \end{array} \right.$$

- Obtain a (possibly ambiguous) context-free tree grammar for \mathcal{C} .
- Inclusion-exclusion gives a polynomial system for $C(z)$.
- **Next steps:** Automatic computation of a tree grammar for \mathcal{C} , possibly unambiguous (=combinatorial specification).

Algorithmization

- **Input:** a finite basis B defining $\mathcal{C} = Av(B)$

Algorithmization

- **Input:** a finite basis B defining $\mathcal{C} = Av(B)$
- Decide whether \mathcal{C} contains **finitely many simples**:
 - Naive semi-decision procedure [Schmerl, Trotter 93]
 - Decision procedure [Brignall, Ruškuc, Vatter 08]
 - “Much more practical” algorithm [Bassino, B., Pierrot, Rossin 15]

Algorithmization

- **Input:** a finite basis B defining $\mathcal{C} = Av(B)$
- Decide whether \mathcal{C} contains **finitely many simples**:
 - Naive semi-decision procedure [Schmerl, Trotter 93]
 - Decision procedure [Brignall, Ruškuc, Vatter 08]
 - “Much more practical” algorithm [Bassino, B., Pierrot, Rossin 15]
- Compute the **set of simples** in \mathcal{C} :
 - In a naive way [Albert, Atkinson 05] using [Schmerl, Trotter 93]
 - Using the structure of the poset of simples [Pierrot, Rossin 2017+]

Algorithmization

- **Input:** a finite basis B defining $\mathcal{C} = Av(B)$
- Decide whether \mathcal{C} contains **finitely many simples**:
 - Naive semi-decision procedure [Schmerl, Trotter 93]
 - Decision procedure [Brignall, Ruškuc, Vatter 08]
 - “Much more practical” algorithm [Bassino, B., Pierrot, Rossin 15]
- Compute the **set of simples** in \mathcal{C} :
 - In a naive way [Albert, Atkinson 05] using [Schmerl, Trotter 93]
 - Using the structure of the poset of simples [Pierrot, Rossin 2017+]
- Compute an **unambiguous tree grammar** for \mathcal{C} :
 - With query-complete sets (not effective) [Brignall, Huczynska, Vatter 08]
 - Algorithm propagating pattern avoidance and containment constraints in the tree grammar [Bassino, B., Pierrot, Pivoteau, Rossin 2017+]

Experimenting with the results of this algorithm

The algorithm produces a combinatorial specification for \mathcal{C} .
From it, we automatically derive a [Boltzmann sampler](#) of permutations in \mathcal{C} [Flajolet, Fusy, Pivoteau 07].

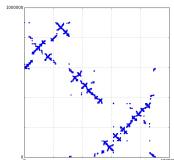
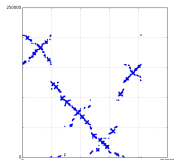
Experimenting with the results of this algorithm

The algorithm produces a combinatorial specification for \mathcal{C} .

From it, we automatically derive a [Boltzmann sampler](#) of permutations in \mathcal{C} [Flajolet, Fusy, Pivoteau 07].

Example: $\mathcal{C} = Av(2413, 3142)$ the class of [separable permutations](#):

Two separable permutations of size 204523 and 903073, drawn uniformly at random among those of the same size:



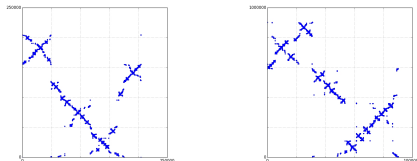
Experimenting with the results of this algorithm

The algorithm produces a combinatorial specification for \mathcal{C} .

From it, we automatically derive a **Boltzmann sampler** of permutations in \mathcal{C} [Flajolet, Fusy, Pivoteau 07].

Example: $\mathcal{C} = Av(2413, 3142)$ the class of **separable permutations**:

Two separable permutations of size 204523 and 903073, drawn uniformly at random among those of the same size:



Goal: Explain these diagrams, by describing the “limit shape” of random separable permutations of size $n \rightarrow +\infty$.

Proportion of patterns in separable permutations

- Notation:

- $\widetilde{\text{occ}}(\pi, \sigma) = \frac{\text{number of occurrences of } \pi \text{ in } \sigma}{\binom{n}{k}}$ for $n = |\sigma|$ and $k = |\pi|$

- \mathcal{S}_n = a uniform random separable permutation of size n

- Theorem [Bassino, B., Féray, Gerin, Pierrot 2017+]:

There exist random variables (Λ_π) , π ranging over all permutations, such that for all π , $0 \leq \Lambda_\pi \leq 1$ and when $n \rightarrow +\infty$, $\widetilde{\text{occ}}(\pi, \mathcal{S}_n)$ converges in distribution to Λ_π .

Substitution decomposition is essential to the proof.

Proportion of patterns in separable permutations

- Notation:

- $\widetilde{\text{occ}}(\pi, \sigma) = \frac{\text{number of occurrences of } \pi \text{ in } \sigma}{\binom{n}{k}}$ for $n = |\sigma|$ and $k = |\pi|$

- \mathcal{O}_n = a uniform random separable permutation of size n

- Theorem [Bassino, B., Féray, Gerin, Pierrot 2017+]:

There exist random variables (Λ_π) , π ranging over all permutations, such that for all π , $0 \leq \Lambda_\pi \leq 1$ and when $n \rightarrow +\infty$, $\widetilde{\text{occ}}(\pi, \mathcal{O}_n)$ converges in distribution to Λ_π .

Substitution decomposition is essential to the proof.

Moreover,

- We describe a construction of Λ_π .
- This holds jointly for patterns π_1, \dots, π_r .
- If π is separable of size at least 2, Λ_π is non-deterministic.
- Combinatorial formula for all moments of Λ_π .

What does this say about limit shapes of diagrams?

- Permutons and permuton convergence:
 - Permuton = measure on $[0, 1]^2$ with uniform marginals
 \approx diagram of a finite or infinite permutation.
 - The convergence of $\widetilde{\text{occ}}(\pi, \sigma)$ for all π characterizes the convergence of permutons [Hoppen, Kohayakawa, Moreira, Rath, Sampaio 13; brought to a probabilistic setting].
 - Hence, denoting μ_σ the permuton associated with σ , there exists a random permuton μ such that μ_{σ_n} tends to μ in distribution (in the weak convergence topology).

What does this say about limit shapes of diagrams?

- **Permutons and permuton convergence:**
 - **Permuton** = measure on $[0, 1]^2$ with uniform marginals
 \approx diagram of a **finite or infinite** permutation.
 - The convergence of $\widetilde{\text{occ}}(\pi, \sigma)$ for all π characterizes the convergence of permutons [Hoppen, Kohayakawa, Moreira, Rath, Sampaio 13; brought to a probabilistic setting].
 - Hence, denoting μ_σ the permuton associated with σ , there exists a random permuton μ such that μ_{σ_n} tends to μ in distribution (in the weak convergence topology).
- **Properties of μ :**
 - μ is not deterministic [Bassino, B., Féray, Gerin, Pierrot 2017+].
 - Construction of μ directly in the continuum [Maazoun 2017+].
 - μ has Hausdorff dimension 1 [Maazoun 2017+].

Extension to substitution-closed classes

A permutation class \mathcal{C} is **substitution-closed** when:

- $\pi[\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)}]$ belongs to \mathcal{C} as soon as π and all $\alpha^{(i)}$ do;
- equivalently, the decomposition trees of permutations in \mathcal{C} are all decomposition trees built using simple permutations in \mathcal{C} .

Remark: The class of separable permutations is the smallest (non-trivial) substitution-closed class (it contains no simples).

Extension to substitution-closed classes

A permutation class \mathcal{C} is **substitution-closed** when:

- $\pi[\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)}]$ belongs to \mathcal{C} as soon as π and all $\alpha^{(i)}$ do;
- equivalently, the decomposition trees of permutations in \mathcal{C} are all decomposition trees built using simple permutations in \mathcal{C} .

Remark: The class of separable permutations is the smallest (non-trivial) substitution-closed class (it contains no simples).

Theorem [Bassino, B., Féray, Gerin, Maazoun, Pierrot 2017+]:

Let \mathcal{C} be a substitution-closed class, whose set S of simple permutations satisfies (mild?) enumeration conditions.

(e.g. S finite, or $|S_n|$ uniformly bounded, or GF of S rational or of radius of convergence $> \sqrt{2} - 1$, ... are sufficient conditions)

There exists a **random permuton** $\mu^{\mathcal{C}}$ (a one-parameter deformation of μ) which is the **limit of** permutons of uniform random permutations in \mathcal{C} .

Extension to substitution-closed classes

A permutation class \mathcal{C} is **substitution-closed** when:

- $\pi[\alpha^{(1)}, \alpha^{(2)}, \dots, \alpha^{(k)}]$ belongs to \mathcal{C} as soon as π and all $\alpha^{(i)}$ do;
- equivalently, the decomposition trees of permutations in \mathcal{C} are all decomposition trees built using simple permutations in \mathcal{C} .

Remark: The class of separable permutations is the smallest (non-trivial) substitution-closed class (it contains no simples).

Theorem [Bassino, B., Féray, Gerin, Maazoun, Pierrot 2017+]:

Let \mathcal{C} be a substitution-closed class, whose set S of simple permutations satisfies (mild?) enumeration conditions.

(e.g. S finite, or $|S_n|$ uniformly bounded, or GF of S rational or of radius of convergence $> \sqrt{2} - 1$, ... are sufficient conditions)

There exists a **random permuton** $\mu^{\mathcal{C}}$ (a one-parameter deformation of μ) which is the **limit of** permutons of uniform random permutations in \mathcal{C} .

Thank you for listening!