Automatic Enumeration and Random Generation for pattern-avoiding Permutation Classes

Adeline Pierrot

Institute of Discrete Mathematics and Geometry, TU Wien (Vienna)

Permutation Patterns 2014

Joint work with F. Bassino, M. Bouvel, C. Pivoteau and D. Rossin
Main Result

Enumeration and Random Generation for Permutation Classes

Permutation class

Algorithm

Enumeration

Random sampler
Main Result

Enumeration and Random Generation for Permutation Classes

Permutation class

Algorithm

Generating function

Random sampler
Main Result

Enumeration and Random Generation for Permutation Classes

Excluded patterns

Algorithm

Generating function
Random sampler
Patterns in permutations

Permutation: $\sigma = 3 \ 1 \ 2 \ 8 \ 5 \ 4 \ 7 \ 9 \ 6$
Patterns in permutations

**Permutation**: $\sigma = 3\,1\,2\,8\,5\,4\,7\,9\,6$

**Pattern**: $1\,3\,2\,4 \preceq 3\,1\,2\,8\,5\,4\,7\,9\,6$ since $2\,5\,4\,9 \equiv 1\,3\,2\,4$. 

![Diagram showing permutation and pattern](image-url)
Patterns in permutations

Permutation: \( \sigma = 3 \ 1 \ 2 \ 8 \ 5 \ 4 \ 7 \ 9 \ 6 \)

Pattern: 1 3 2 4 \( \preceq \) 3 1 2 8 5 4 7 9 6 since 2 5 4 9 \( \equiv \) 1 3 2 4.

\( \sigma \) contains 1 3 2 4 but avoids 4 3 2 1
Patterns in permutations

Permutation: $\sigma = 3 \, 1 \, 2 \, 8 \, 5 \, 4 \, 7 \, 9 \, 6$

Pattern: $1 \, 3 \, 2 \, 4 \preceq 3 \, 1 \, 2 \, 8 \, 5 \, 4 \, 7 \, 9 \, 6$ since $2 \, 5 \, 4 \, 9 \equiv 1 \, 3 \, 2 \, 4$.

$\sigma$ contains $1 \, 3 \, 2 \, 4$

but avoids $4 \, 3 \, 2 \, 1$

Remark: $\sigma, \pi$ as input, deciding whether $\pi \preceq \sigma$ is NP-complete.
Permutation Classes

Class of permutations = set downward closed for $\preceq$:

$\sigma \in C$ and $\pi \preceq \sigma \Rightarrow \pi \in C$

Example : Increasing permutations $\bigcup_{n=1}^{\infty}\{12\ldots n\}$

$Av(B)$ : the set of permutations avoiding all the elements of $B$.

Example : $Av(21) = \bigcup_{n=1}^{\infty}\{12\ldots n\}$
Permutation Classes

Class of permutations = set downward closed for \( \preceq \):
\[
\sigma \in C \text{ and } \pi \preceq \sigma \implies \pi \in C
\]

Example: Increasing permutations \( \bigcup_{n=1}^{\infty} \{12\ldots n\} \)

\( Av(B) \): the set of permutations avoiding all the elements of \( B \).
Example: \( Av(21) = \bigcup_{n=1}^{\infty} \{12\ldots n\} \)

Prop.: Every class \( C \) is characterized by its basis \( B \):

\[
\forall C, \exists \text{ a unique antichain } B \text{ s.t. } C = Av(B)
\]

\( B \) finite or infinite.

Two points of view:
\( C \) given by \( B \) / by a property stable for \( \preceq \)
Issues

Algorithms

- Test the membership to a class

Combinatorics

- Above all *enumeration*
  \(\leftrightarrow\) Lots of existing *ad hoc results* about a given class
- Search for general theory
  \(\leftrightarrow\) Search *structure* in classes

Theorem [Albert, Atkinson, 2005]:
\[ C \text{ contains finitely many simple permutations} \quad \Rightarrow \quad C \text{ is finitely based and has an algebraic generating function.} \]
**Issues**

**Algorithms**
- Test the **membership** to a class

**Combinatorics**
- Above all **enumeration**
  - $\rightarrow$ Lots of existing **ad hoc results** about a given class
- Search for general theory
  - $\rightarrow$ Search **structure** in classes

**Theorem**[Albert, Atkinson, 2005] :

$C$ contains finitely many simple permutations
  $\Rightarrow C$ is finitely based and has an algebraic generating function.
Issues

Algorithms

- Test the membership to a class

Combinatorics

- Above all enumeration
  → Lots of existing ad hoc results about a given class
- Search for general theory
  → Search structure in classes

Theorem [Albert, Atkinson, 2005]:
$C$ contains finitely many simple permutations
  ⇒ $C$ is finitely based and has an algebraic generating function.

This talk: Automatic process for enumeration / generation
Main tool: Recursive decomposition
Main tool : Substitution decomposition

**Substitution decomposition** : recursively describe discrete objects by decomposing them in **core items** (prime structures).

**Existence and unicity** of the decomposition.

**Examples** :
- prime factorization of integers (core items = prime numbers),
- modular decomposition of graphs (core items = prime graphs)...

Core items in the case of permutations : **simple permutations**.
Simple permutations

**Block** = window of elements of $\sigma$ whose values form a range

**Example**: $3 \, 5 \, 4$ is a block of $6 \, 3 \, 5 \, 4 \, 1 \, 7 \, 2$
**Simple permutations**

**Block** = window of elements of $\sigma$ whose values form a range

**Example**: $3 \ 5 \ 4$ is a block of $6 \ 3 \ 5 \ 4 \ 1 \ 7 \ 2$

Simple permutations

**Block** = window of elements of $\sigma$ whose values form a range

Example: $3 \ 5 \ 4$ is a block of $6 \ 3 \ 5 \ 4 \ 1 \ 7 \ 2$

Smallest ones: $1 \ 2$, $2 \ 1$, $2 \ 4 \ 1 \ 3$, $3 \ 1 \ 4 \ 2$
Simple permutations

**Block** = window of elements of $\sigma$ whose values form a range

Example: 3 5 4 is a block of 6 3 5 4 1 7 2

**Simple permutation** = has no block
Simple permutations

**Block** = window of elements of $\sigma$ whose values form a range

**Example**: $3 \ 5 \ 4$ is a block of $6 \ 3 \ 5 \ 4 \ 1 \ 7 \ 2$

---

**Simple permutation** = has no block

**Example**: $3 \ 1 \ 7 \ 4 \ 6 \ 2 \ 5$ is simple, $6 \ 3 \ 5 \ 4 \ 1 \ 7 \ 2$ is not simple.
Simple permutations

Block = window of elements of $\sigma$ whose values form a range
Example: 3 5 4 is a block of 6 3 5 4 1 7 2

Simple permutation = has no block
Example: 3 1 7 4 6 2 5 is simple, 6 3 5 4 1 7 2 is not simple.
Simple permutations

**Block** = window of elements of $\sigma$ whose values form a range

*Example*: $3 \ 5 \ 4$ is a block of $6 \ 3 \ 5 \ 4 \ 1 \ 7 \ 2$

**Simple permutation** = has no block

*Example*: $3 \ 1 \ 7 \ 4 \ 6 \ 2 \ 5$ is simple,
$6 \ 3 \ 5 \ 4 \ 1 \ 7 \ 2$ is not simple.
Simple permutations

**Block** = window of elements of $\sigma$ whose values form a range

**Example**: 3 5 4 is a block of 6 3 5 4 1 7 2

**Simple permutation** = has no block

**Example**: 3 1 7 4 6 2 5 is simple, 6 3 5 4 1 7 2 is not simple.
Simple permutations

Block = window of elements of $\sigma$ whose values form a range

Example: 3 5 4 is a block of 6 3 5 4 1 7 2

Simple permutation = has no block

Example: 3 1 7 4 6 2 5 is simple, 6 3 5 4 1 7 2 is not simple.
Simple permutations

**Block** = window of elements of $\sigma$ whose values form a range

*Example*: $3 \ 5 \ 4$ is a block of $6 \ 3 \ 5 \ 4 \ 1 \ 7 \ 2$

**Simple permutation** = has no block

*Example*: $3 \ 1 \ 7 \ 4 \ 6 \ 2 \ 5$ is simple, $6 \ 3 \ 5 \ 4 \ 1 \ 7 \ 2$ is not simple.

*Smallest ones*: $12, \ 21, \ 2413, \ 3142$
Substitution $\sigma[\pi^1, \ldots, \pi^n]$ : Replace each point $\sigma_i$ by a block $\pi^i$.

Example : $1 \ 3 \ 2[2 \ 1, \ 1 \ 3 \ 2, \ 1] = 2 \ 1 \ 4 \ 6 \ 5 \ 3$. 
Substitution $\sigma[\pi^1, \ldots, \pi^n]$: Replace each point $\sigma_i$ by a block $\pi^i$.

Example: $1 \ 3 \ 2 \ [2 \ 1 \ 3 \ 2 \ 1] = 2 \ 1 \ 4 \ 6 \ 5 \ 3$.

Remark: $\sigma[\pi^1, \ldots, \pi^n] \in C \Rightarrow \sigma, \pi^1, \ldots, \pi^n \in C$.
Substitution $\sigma[\pi^1, \ldots, \pi^n]$: Replace each point $\sigma_i$ by a block $\pi^i$.

Example: $1\ 3\ 2[2\ 1,\ 1\ 3\ 2,\ 1] = 2\ 1\ 4\ 6\ 5\ 3$.

Remark: $\sigma[\pi^1, \ldots, \pi^n] \in C \implies \sigma, \pi^1, \ldots, \pi^n \in C$

Substitution-closed class:
$\sigma, \pi^1, \ldots, \pi^n \in C \implies \sigma[\pi^1, \ldots, \pi^n] \in C$. 
Substitution decomposition

**Theorem** [Albert Atkinson 05] : \( \forall \sigma \neq 1, \exists \) a unique simple permutation \( \pi \) and unique \( \alpha_i \) such that \( \sigma = \pi[\alpha_1, \ldots, \alpha_k] \).

Example:
\[
\sigma = 10 12 14 11 13 1 21 19 16 18 20 17 15 4 8 3 2 9 5 6 7
= 3142 [13524, 1, 7524631, 37218456] = 3142 [12[1, 2413], 1, 1, 21[1, 524631, 24153[1, 1, 123]]]
= ...
\]
Substitution decomposition

**Theorem** [Albert Atkinson 05] : ∀σ ≠ 1, ∃ a unique simple permutation π and unique α_i such that σ = π[α_1, ..., α_k]. If π = 12 (or 21), for unicity, α_1 is 12 (resp. 21) -indecomposable.
**Substitution decomposition**

**Theorem**[Albert Atkinson 05] : \( \forall \sigma \neq 1, \exists \) a unique simple permutation \( \pi \) and unique \( \alpha_i \) such that \( \sigma = \pi[\alpha_1, \ldots, \alpha_k] \).

If \( \pi = 12 \) (or 21), for unicity, \( \alpha_1 \) is 12 (resp. 21) -indecomposable.

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

\[ = 3142 \ [13524, 1, 7524631, 37218456] \]

\[ = 3142 \ [12[1, 2413], 1, 21[1, 524631], 24153[1, 1, 21, 1, 123]] \]

\[ = \ldots \]
**Theorem**[Albert Atkinson 05] : \( \forall \sigma \neq 1, \exists \) a unique simple permutation \( \pi \) and unique \( \alpha_i \) such that \( \sigma = \pi[\alpha_1, \ldots, \alpha_k] \)

If \( \pi = 12 \) (or 21), for unicity, \( \alpha_1 \) is 12 (resp. 21) -indecomposable.

**Consequence** : For any permutation class \( C \),

\[
C \subset \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

Where \( S_C = \{\pi \in C \mid \pi \text{ simple } \neq 12, 21\} \)

\( C^+ = \{\alpha \in C \mid \alpha \text{ is 12-indecomposable}\} \)

\( C^- = \{\alpha \in C \mid \alpha \text{ is 21-indecomposable}\} \)
Decomposition $\Rightarrow$ Equation?

**Theorem** [Albert Atkinson 05]: $\forall \sigma \neq 1$, $\exists$ a unique simple permutation $\pi$ and unique $\alpha_i$ such that $\sigma = \pi[\alpha_1, \ldots, \alpha_k]$

If $\pi = 12$ (or 21), for unicity, $\alpha_1$ is 12 (resp. 21) -indecomposable.

**Consequence**: If $C$ is a substitution-closed class (containing 12, 21)

$$C \subset \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]$$

Where $S_C = \{\pi \in C \mid \pi$ simple $\neq 12, 21\}$

$C^+ = \{\alpha \in C \mid \alpha$ is 12-indecomposable $\}$

$C^- = \{\alpha \in C \mid \alpha$ is 21-indecomposable $\}$
Decomposition ⇒ Equation?

**Theorem** [Albert Atkinson 05]: \( \forall \sigma \neq 1, \exists \) a unique simple permutation \( \pi \) and unique \( \alpha_i \) such that \( \sigma = \pi[\alpha_1, \ldots, \alpha_k] \)

If \( \pi = 12 \) (or 21), for unicity, \( \alpha_1 \) is 12 (resp. 21) -indecomposable.

**Consequence**: If \( C \) is a substitution-closed class (containing 12, 21)

\[
C = \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

Where \( S_C = \{\pi \in C \mid \pi \text{ simple } \neq 12, 21\} \)

\( C^+ = \{\alpha \in C \mid \alpha \text{ is 12-indecomposable}\} \)

\( C^- = \{\alpha \in C \mid \alpha \text{ is 21-indecomposable}\} \)
Theorem [Albert Atkinson 05] : \( \forall \sigma \neq 1, \exists \) a unique simple permutation \( \pi \) and unique \( \alpha_i \) such that \( \sigma = \pi[\alpha_1, \ldots, \alpha_k] \)

If \( \pi = 12 \) (or 21), for unicity, \( \alpha_1 \) is 12 (resp. 21) -indecomposable.

Consequence : If \( C \) is a substitution-closed class (containing 12, 21)

\[
C = \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

\[
C^+ = \{1\} \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

\[
C^- = \{1\} \uplus 12[C^+, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

Where \( S_C = \{\pi \in C \mid \pi \text{ simple } \neq 12, 21\} \)

\( C^+ = \{\alpha \in C \mid \alpha \text{ is 12-indecomposable} \} \)

\( C^- = \{\alpha \in C \mid \alpha \text{ is 21-indecomposable} \} \)
Combinatorial Specification (closed classes)

If $C$ is a substitution-closed class (containing 12, 21)

$$C = \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]$$

$$C^+ = \{1\} \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]$$

$$C^- = \{1\} \uplus 12[C^+, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]$$

Where $S_C = \{\pi \in C \mid \pi \text{ simple } \neq 12, 21\}$

$C^+ = \{\alpha \in C \mid \alpha \text{ is 12-indecomposable} \}$

$C^- = \{\alpha \in C \mid \alpha \text{ is 21-indecomposable} \}$
Combinatorial Specification (closed classes)

If $C$ is a substitution-closed class (containing 12, 21)

$$
C = \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \bigcup_{\pi \in S_C} \pi[C \ldots C]
$$

$$
C^+ = \{1\} \uplus 21[C^-, C] \uplus \bigcup_{\pi \in S_C} \pi[C \ldots C]
$$

$$
C^- = \{1\} \uplus 12[C^+, C] \uplus \bigcup_{\pi \in S_C} \pi[C \ldots C]
$$

Where $S_C = \{\pi \in C \mid \pi \text{ simple } \neq 12, 21\}$

$C^+ = \{\alpha \in C \mid \alpha \text{ is 12-indecomposable}\}$

$C^- = \{\alpha \in C \mid \alpha \text{ is 21-indecomposable}\}$

$\hookrightarrow$ Combinatorial Specification if $S_C$ is finite and known
(Recursive description using combinatorial constructors)
$c_n =$ Number of permutations of size $n$ in $\mathcal{C}$

Generating function $C(x) = \sum_{n=1}^{\infty} c_n x^n$
Combinatorial Specification \(\Rightarrow\) Equation

\[ c_n = \text{Number of permutations of size } n \text{ in } C \]

Generating function  
\[ C(x) = \sum_{n=1}^{\infty} c_n \cdot x^n \]

Symbolic method = a systematic translation mechanism:  
combinatorial constructions \(\rightarrow\) operations on generating functions.
Combinatorial Specification $\Rightarrow$ Equation

c_n = \text{Number of permutations of size } n \text{ in } \mathcal{C}

Generating function $C(x) = \sum_{n=1}^{\infty} c_n x^n$

**Symbolic method** = a systematic translation mechanism: combinatorial constructions $\rightarrow$ operations on generating functions. (see *Analytic Combinatorics* [Philippe Flajolet, Robert Sedgewick 2009])
Combinatorial Specification ⇒ Equation

c_n = Number of permutations of size n in C
Generating function \( C(x) = \sum_{n=1}^{\infty} c_n \cdot x^n \)

Symbolic method = a systematic translation mechanism: combinatorial constructions → operations on generating functions. (see Analytic Combinatorics [Philippe Flajolet, Robert Sedgewick 2009])

\[ C = \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C] \]
Combinatorial Specification $\Rightarrow$ Equation

\[ c_n = \text{Number of permutations of size } n \text{ in } \mathcal{C} \]

Generating function
\[ C(x) = \sum_{n=1}^{\infty} c_n x^n \]

**Symbolic method** = a systematic translation mechanism: combinatorial constructions $\rightarrow$ operations on generating functions.

(see *Analytic Combinatorics* [Philippe Flajolet, Robert Sedgewick 2009])

\[
\mathcal{C} = \{1\} \cup 12[\mathcal{C}^+, \mathcal{C}] \cup 21[\mathcal{C}^-, \mathcal{C}] \cup \biguplus_{\pi \in \mathcal{S}_{\mathcal{C}}} \pi[\mathcal{C} \ldots \mathcal{C}]
\]

\[
\rightarrow C(z) = z + \mathcal{C}^+(z).C(z) + \mathcal{C}^-(z).C(z) + S_{\mathcal{C}}(C(z))
\]
Combinatorial Specification ⇒ Equation

\( c_n = \text{Number of permutations of size } n \text{ in } C \)

Generating function \( C(x) = \sum_{n=1}^{\infty} c_n \cdot x^n \)

**Symbolic method** = a systematic translation mechanism:
combinatorial constructions \( \rightarrow \) operations on generating functions.
(see *Analytic Combinatorics* [Philippe Flajolet, Robert Sedgewick 2009])

\[
C = \{1\} \uplus 12[C^+ , C] \uplus 21[C^- , C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

\[
\rightarrow C(z) = z + C^+(z) \cdot C(z) + C^-(z) \cdot C(z) + S_C(C(z))
\]

**Same way**

\[
C^+(z) = z + C^-(z) \cdot C(z) + S_C(C(z))
\]

**and**

\[
C^-(z) = z + C^+(z) \cdot C(z) + S_C(C(z))
\]
Combinatorial Specification $\Rightarrow$ Equation

\[ c_n = \text{Number of permutations of size } n \text{ in } C \]

**Generating function** \( C(x) = \sum_{n=1}^{\infty} c_n x^n \)

**Symbolic method** = a systematic translation mechanism: combinatorial constructions $\rightarrow$ operations on generating functions. (see *Analytic Combinatorics* [Philippe Flajolet, Robert Sedgewick 2009])

\[ C = \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C] \]

$\rightarrow$ \[ C(z) = z + C^+(z).C(z) + C^-(z).C(z) + S_C(C(z)) \]

**Combinatorial specification** $\rightarrow$ **generating function** and **uniform random sampler** (recursive method or Boltzmann’s method)
$B$: finite basis of excluded patterns

**General case**

- Finite number of simple permutations in $Av(B)$?
  - $O(n \log n + p^{2k})$ [BBPR]
  - Yes
  - Computation of simple permutations
    - $O(N.k.\ell^{p+2})$ [PR]
    - Iterative computation [BBPPR]

**Substitution-closed case**

- $O(n \log n)$ [BBPR]
- No
  - Exit

**Specification for $Av(B)$**

- Generating function
- Random sampler

$n = \sum_{\beta \in B} |\beta|$, $k = |B|$, $N = |S_C|$, $\ell = \max\{|\pi| : \pi \in S_C\}$ et $p = \max\{|\beta| : \beta \in B\}$

BBPPR = Bassino, Bouvel, Pierrot, Pivoteau, Rossin
Combinatorial Specification (closed classes)

If \( C \) is a substitution-closed class (containing 12, 21)

\[
C = \{1\} \uplus 12[C^+, C] \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

\[
C^+ = \{1\} \uplus 21[C^-, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

\[
C^- = \{1\} \uplus 12[C^+, C] \uplus \biguplus_{\pi \in S_C} \pi[C \ldots C]
\]

Where \( S_C = \{\pi \in C \mid \pi \text{ simple } \neq 12, 21\} \)
\( C^+ = \{\alpha \in C \mid \alpha \text{ is 12-indecomposable} \} \)
\( C^- = \{\alpha \in C \mid \alpha \text{ is 21-indecomposable} \} \)

\(\iff\) **Combinatorial Specification** if \( S_C \) is finite and known
(Recursive description using combinatorial constructors)
$B$: finite basis of excluded patterns

General case

Substitution-closed case

Finite number of simple permutations in $Av(B)$?

$O(n \log n + p^{2k})$ [BBPR]

$O(n \log n)$ [BBPR]

Computation of simple permutations

$O(N.k.\ell^{p+2})$ [PR]

$O(N.\ell^4)$ [PR]

Iterative computation

[BBPPR]

Specification for $Av(B)$

Generating function

Random sampler

$n = \sum_{\beta \in B} |\beta|$, $k = |B|$, $N = |S_C|$, $\ell = \max\{|\pi| : \pi \in S_C\}$ et $p = \max\{|\beta| : \beta \in B\}$

BBPPR = Bassino, Bouvel, Pierrot, Pivoteau, Rossin
Finite number of simple permutations?

**Thm** [Brignall and al.]: Input = (finite) basis $B$. We can decide whether $C = \text{Av}(B)$ contains finitely many simple permutations.

**Procedure**: Check if $C$ contains finitely many

1. parallel alternations
2. wedge simple permutations
3. proper pin-permutations
Finite number of simple permutations?

**Thm** [Brignall and al.] : Input = (finite) basis $B$. We can decide whether $C = Av(B)$ contains finitely many simple permutations.

**Procedure** : Check if $C$ contains finitely many
   1. parallel alternations
   2. wedge simple permutations
   3. proper pin-permutations

**Complexity** : (with $n = \sum_{\pi \in B} |\pi|$, $k = |B|$ and $s \leq m = \max |\pi|$)
   1. and 2. : pattern matching for patterns of size $\leq 4 \rightarrow O(n \log n)$
   3. : use words and automata theory
      Brignall’s procedure : $O(n.8^m + 2^{k \cdot s \cdot 2^s})$
      Our algorithm [BBPR] : $O(n + s^{2k})$
      Substitution-closed class : $O(n)$. 
Prop[Schmerl, Trotter 93]: Any simple permutation of size $n$ has a simple pattern of size $n - 1$ or $n - 2$. 
Compute simple permutations

**Prop [Schmerl, Trotter 93]**: Any simple permutation of size $n$ has a simple pattern of size $n - 1$ or $n - 2$.

**Naive algorithm**: Generate all simple permutations $\sigma$ of size $n$ and test if $\sigma \in C$ until $n$ s.t. no simple of size $n$ or $n - 1$ belongs to $C$. 

**Improvement [PR]**: Restrict the number of tests: Build candidates from simples of size $n - 1$ of $C \rightarrow < n$.

$c_n$ tests instead of $n!$.

**Thm [PR]**: $\sigma, \pi$ simple, $\pi \prec \sigma \Rightarrow \exists \tau$ simple s.t. $\pi \preceq \tau \prec \sigma$.

**Improvement [PR]**: If $C$ substitution-closed, no need of pattern matching: just check permutations obtained by deleting a point to the candidate.
Compute simple permutations

Prop[Schmerl, Trotter 93] : Any simple permutation of size $n$ has a simple pattern of size $n - 1$ or $n - 2$.

Naive algorithm : Generate all simple permutations $\sigma$ of size $n$ and test if $\sigma \in C$ until $n$ s.t. no simple of size $n$ or $n - 1$ belongs to $C$.

Improvement[PR] : Restrict the number of tests : Build candidates from simples of size $n - 1$ of $C$ $\rightarrow < n.c^n$ tests instead of $n!$. 
Compute simple permutations

**Prop**[Schmerl, Trotter 93] : Any simple permutation of size $n$ has a simple pattern of size $n - 1$ or $n - 2$.

**Naive algorithm** : Generate all simple permutations $\sigma$ of size $n$ and test if $\sigma \in C$ until $n$ s.t. no simple of size $n$ or $n - 1$ belongs to $C$.

**Improvement**[PR] : Restrict the number of tests : Build candidates from simples of size $n - 1$ of $C$

$\rightarrow < n.c^n$ tests instead of $n!$.

**Thm**[PR] : $\sigma$, $\pi$ simple, $\pi \prec \sigma$ $\Rightarrow \exists \tau$ simple s.t. $\pi \preceq \tau \prec \sigma$

\[
\begin{array}{c}
\end{array}
\]
Compute simple permutations

Prop[Schmerl, Trotter 93] : Any simple permutation of size $n$ has a simple pattern of size $n - 1$ or $n - 2$.

Naive algorithm : Generate all simple permutations $\sigma$ of size $n$ and test if $\sigma \in C$ until $n$ s.t. no simple of size $n$ or $n - 1$ belongs to $C$.

Improvement[PR] : Restrict the number of tests : Build candidates from simples of size $n - 1$ of $C$ $\rightarrow < n.c^n$ tests instead of $n!$.

Thm[PR] : $\sigma, \pi$ simple, $\pi \prec \sigma \Rightarrow \exists \tau$ simple s.t. $\pi \preceq \tau \prec \sigma$

Improvement[PR] : If $C$ substitution-closed, no need of pattern matching : just check permutations obtained by deleting a point to the candidate.
Recall: If $C$ substitution-closed (and $12, 21 \in C$)

\[
C = 1 \uplus 12 \uplus 21 \uplus \biguplus_{\pi \in S_C} \pi \\
\left( C^+ \cap C \right) \uplus \left( C^- \cap C \right) \uplus \biguplus_{\pi \in S_C} \pi \cap C
\]

With $S_C = \{ \pi \in C \mid \pi \text{ simple } \neq 12, 21 \}$

$C^+ = \{ \alpha \in C \mid \alpha \text{ is 12-indecomposable} \}$

$C^- = \{ \alpha \in C \mid \alpha \text{ is 21-indecomposable} \}$
Consequence: If $\mathcal{C}$ not closed of closure $\hat{\mathcal{C}}$ (and $12, 21 \in \mathcal{C}$)

$$
\hat{\mathcal{C}} = 1 \cup \hat{\mathcal{C}}^+ \cup \hat{\mathcal{C}}^- \cup \bigcup_{\pi \in \mathcal{S}_\mathcal{C}} \pi \hat{\mathcal{C}}
$$

With $\mathcal{S}_\mathcal{C} = \{ \pi \in \mathcal{C} \mid \pi \text{ simple } \neq 12, 21 \}$

$\mathcal{C}^+ = \{ \alpha \in \mathcal{C} \mid \alpha \text{ is 12-indecomposable } \}$

$\mathcal{C}^- = \{ \alpha \in \mathcal{C} \mid \alpha \text{ is 21-indecomposable } \}$
Consequence: If $\mathcal{C}$ not closed of closure $\hat{\mathcal{C}}$ (and $12, 21 \in \mathcal{C}$)

\[
\hat{\mathcal{C}} = 1 \cup 12 \cup 21 \cup \bigcup_{\pi \in \mathcal{S}_\mathcal{C}} \pi
\]

\[
\hat{\mathcal{C}}^+ \quad \hat{\mathcal{C}}^- \quad \hat{\mathcal{C}} \quad \cdots \quad \hat{\mathcal{C}}
\]

With $\mathcal{S}_\mathcal{C} = \{\pi \in \mathcal{C} \mid \pi \text{ simple } \neq 12, 21\}$

$\mathcal{C}^+ = \{\alpha \in \mathcal{C} \mid \alpha \text{ is 12-indecomposable} \}$

$\mathcal{C}^- = \{\alpha \in \mathcal{C} \mid \alpha \text{ is 21-indecomposable} \}$

$\mathcal{C} = \hat{\mathcal{C}} \cap \operatorname{Av}(B) = \hat{\mathcal{C}}\langle B \rangle$
Consequence: If $\mathcal{C}$ not closed of closure $\hat{\mathcal{C}}$ (and $12, 21 \in \mathcal{C}$)

$$
\hat{\mathcal{C}}(B) = 1 \biguplus 12 \langle B \rangle \biguplus 21 \langle B \rangle \biguplus \biguplus_{\pi \in \mathcal{S}_\mathcal{C}} \pi \langle B \rangle
$$

$$
\hat{\mathcal{C}}^+ \hat{\mathcal{C}} \hat{\mathcal{C}}^- \hat{\mathcal{C}} \hat{\mathcal{C}} \cdots \hat{\mathcal{C}}
$$

With $\mathcal{S}_\mathcal{C} = \{\pi \in \mathcal{C} | \pi \text{ simple} \neq 12, 21\}$

$\mathcal{C}^+ = \{\alpha \in \mathcal{C} | \alpha \text{ is 12-indecomposable}\}$

$\mathcal{C}^- = \{\alpha \in \mathcal{C} | \alpha \text{ is 21-indecomposable}\}$

$\mathcal{C} = \hat{\mathcal{C}} \cap \text{Av}(B) = \hat{\mathcal{C}}(B)$
Combinatorial Specification

Consequence: If $\mathcal{C}$ not closed of closure $\hat{\mathcal{C}}$ (and $12, 21 \in \mathcal{C}$)

$$\hat{\mathcal{C}}\langle B \rangle = 1 \uplus \hat{\mathcal{C}}^{12} \langle B \rangle \uplus \hat{\mathcal{C}}^{21} \langle B \rangle \uplus \bigcup_{\pi \in S_{\mathcal{C}}} \hat{\mathcal{C}}^\pi \langle B \rangle$$

With $S_{\mathcal{C}} = \{\pi \in \mathcal{C} \mid \pi \text{ simple } \neq 12, 21\}$

$\mathcal{C}^+ = \{\alpha \in \mathcal{C} \mid \alpha \text{ is 12-indecomposable}\}$

$\mathcal{C}^- = \{\alpha \in \mathcal{C} \mid \alpha \text{ is 21-indecomposable}\}$

$$\mathcal{C} = \hat{\mathcal{C}} \cap Av(B) = \hat{\mathcal{C}}\langle B \rangle$$
Combinatorial Specification

Consequence: If $\mathcal{C}$ not closed of closure $\hat{\mathcal{C}}$ (and $12, 21 \in \mathcal{C}$)

$$\hat{\mathcal{C}}\langle B \rangle = 1 \uplus 12 \langle B \rangle \uplus 21 \langle B \rangle \uplus \biguplus_{\pi \in S_c} \pi \hat{\mathcal{C}}\langle B \rangle \uplus \hat{\mathcal{C}}^+ \hat{\mathcal{C}} \hat{\mathcal{C}}^- \hat{\mathcal{C}} \uplus \hat{\mathcal{C}}\langle B_1 \rangle \cdots \hat{\mathcal{C}}\langle B_k \rangle$$

Constraint Propagation $\rightarrow$ System of equations like:

$$\mathcal{C}_1 = 1 \uplus 12[\mathcal{C}_2, \mathcal{C}_3] \uplus 21[\mathcal{C}_4, \mathcal{C}_5] \uplus \biguplus_{\pi \in S_c} \pi[\mathcal{C}_6, \ldots, \mathcal{C}_k]$$
Combinatorial Specification

Consequence: If $\mathcal{C}$ not closed of closure $\hat{\mathcal{C}}$ (and $12, 21 \in \mathcal{C}$)

$$\hat{\mathcal{C}}\langle B \rangle = 1 \uplus \begin{array}{c} 12 \langle B \rangle \uplus 21 \langle B \rangle \uplus \biguplus_{\pi \in S_\mathcal{C}} \uplus \end{array} \begin{array}{c} \hat{\mathcal{C}}^+ \hat{\mathcal{C}} \hat{\mathcal{C}}^- \hat{\mathcal{C}} \end{array} \begin{array}{c} \hat{\mathcal{C}}\langle B_1 \rangle \cdots \hat{\mathcal{C}}\langle B_k \rangle \end{array}$$

Constraint Propagation $\rightarrow$ System of equations like:

$$\mathcal{C}_1 = 1 \uplus 12[\mathcal{C}_2, \mathcal{C}_3] \uplus 21[\mathcal{C}_4, \mathcal{C}_5] \uplus \biguplus_{\pi \in S_\mathcal{C}} \uplus \pi[\mathcal{C}_6, \ldots, \mathcal{C}_k]$$

Iterative computation $\rightarrow$ Ambiguous system.
Combinatorial Specification

**Consequence**: If $C$ not closed of closure $\hat{C}$ (and $12, 21 \in C$)

$$\hat{C}\langle B \rangle = 1 \cup 12\langle B \rangle \cup 21\langle B \rangle \cup \bigcup_{\pi \in S_C} \pi \hat{C}\langle B \rangle \cup \hat{C}^+ \hat{C} \cup \hat{C}^- \hat{C} \cup \hat{C}\langle B_1 \rangle \cdots \hat{C}\langle B_k \rangle$$

Constraint Propagation $\rightarrow$ System of equations like:

$$C_1 = 1 \cup 12[C_2, C_3] \cup 21[C_4, C_5] \cup \bigcup_{\pi \in S_C} \pi [C_6, \ldots, C_k]$$

Iterative computation $\rightarrow$ Ambiguous system.

$\rightarrow$ Disambiguation $[AA05]$ inclusion-exclusion

$[BHV08]$ “query-complete sets”

$[BBPPR]$ mandatory patterns

$$A \cup B = A \cap \bar{B} \cup \bar{A} \cap B \cup A \cap B$$
$B$: finite basis of excluded patterns

**General case**

Finite number of simple permutations in $Av(B)$?

- Yes
  - Computation of simple permutations
    - $O(N.k.\ell^{p+2})$ [PR]
  - Iterative computation [BBPPR]

- No
  - Exit

**Substitution-closed case**

Finite number of simple permutations in $Av(B)$?

- Yes
  - Computation of simple permutations
    - $O(N.\ell^4)$ [PR]

- No
  - Exit

Specification for $Av(B)$

Generating function

Random sampler

$n = \sum_{\beta \in B} |\beta|$, $k = |B|$, $N = |\mathcal{S}_c|$, $\ell = \max\{|\pi| : \pi \in \mathcal{S}_c\}$ et $p = \max\{|\beta| : \beta \in B\}$

BBPPR = Bassino, Bouvel, Pierrot, Pivoteau, Rossin
Perspectives

- Maple Library for the permutation patterns community
Perspectives

- Maple Library for the permutation patterns community
- Random generation → tests
Perspectives

- **Maple Library** for the permutation patterns community
- Random generation → **tests**
- **Limit shape** of permutations?

**Figure:** 30,000 random permutations of size 500 of $Av(2413, 1243, 2341, 531642, 41352)$. 
Perspectives

- **Maple Library** for the permutation patterns community
- Random generation $\rightarrow$ tests
- Limit shape of permutations?
- Study the specifications obtained $\rightarrow$ asymptotics?
Perspectives

- **Maple Library** for the permutation patterns community
- Random generation → tests
- **Limit shape** of permutations?
- Study the specifications obtained → asymptotics?
- Finite number of simples: Representative behaviour of all permutation classes?
Perspectives

• Maple Library for the permutation patterns community

• Random generation → tests

• Limit shape of permutations?

• Study the specifications obtained → asymptotics?

• Finite number of simples: Representative behaviour of all permutation classes?

• Generalization (infinite number of simple permutations...)
Perspectives

- **Maple Library** for the permutation patterns community
- Random generation $\rightarrow$ tests
- Limit shape of permutations?
- Study the specifications obtained $\rightarrow$ asymptotics?
- Finite number of simples: Representative behaviour of all permutation classes?
- Generalization (infinite number of simple permutations...)

Thank you!