(Un)ranking permutation classes

Nathanaël Hassaler   Vincent Vajnovszki

LIB, Université Bourgogne Europe, Dijon, France

Malosco, Italy
GASCom, June 2026

Ranking and unranking functions

For a **list** of combinatorial objects we can ask :

- which position in the list of objects a given object has
- determine the objet in a given position

$$\mathrm{rank}_C : C \to \{0, 1, \ldots, |C| - 1\}$$

that maps x to the number of elements of C that are before x

$$\mathrm{unrank}_C : \{0, 1, \ldots, |C| - 1\} \to C$$

that maps p to the object x such that $\mathrm{rank}_C(x) = p$

Ranking and unranking functions

For a **list** of combinatorial objects we can ask :

- which position in the list of objects a given object has
- determine the objet in a given position

$$\mathrm{rank}_C : C \to \{0, 1, \ldots, |C| - 1\}$$

that maps x to the number of elements of C that are before x

$$\mathrm{unrank}_C : \{0, 1, \ldots, |C| - 1\} \to C$$

that maps p to the object x such that $\mathrm{rank}_C(x) = p$

Ranking and unranking functions

For a **list** of combinatorial objects we can ask :

- which position in the list of objects a given object has
- determine the objet in a given position

$$\mathrm{rank}_C : C \to \{0, 1, \ldots, |C| - 1\}$$

that maps x to the number of elements of C that are before x

$$\mathrm{unrank}_C : \{0, 1, \ldots, |C| - 1\} \to C$$

that maps p to the object x such that $\mathrm{rank}_C(x) = p$

Bibliography I

D.H. Lehmer, *The machine tools of combinatorics*, 1964

F. Ruskey, *Combinatorial Generations*, 2003

P. Kagey *Ranking and unranking restricted permutations*, 2024

algorithms MDPI

Article

Lexicographic Unranking of Combinations Revisited

Antoine Genitrini * and Martin Pépin

CNRS, Laboratoire de Paris 6—LIP6—UMR 7606, Sorbonne Université, F-75005 Paris, France; martin.pepin@lip6.fr

* Correspondence: antoine.genitrini@lip6.fr

Abstract: In the context of combinatorial sampling, the so-called "unranking method" can be seen as a link between a total order over the objects and an effective way to construct an object of given rank. The most classical order used in this context is the lexicographic order, which corresponds to the familiar word ordering in the dictionary. In this article, we propose a comparative study of four algorithms dedicated to the lexicographic unranking of combinations, including three algorithms that were introduced decades ago. We start the paper with the introduction of our new algorithm using a new strategy of computations based on the classical factorial numeral system (or factoradics). Then, we present, in a high level, the three other algorithms. For each case, we analyze its time complexity on average, within a uniform framework, and describe its strengths and weaknesses. For about 20 years, such algorithms have been implemented using big integer arithmetic rather than bounded integer arithmetic which makes the cost of computing some coefficients higher than previously stated. We propose improvements for all implementations, which take this fact into account, and we give a detailed complexity analysis, which is validated by an experimental analysis. Finally, we show that, even if the algorithms are based on different strategies, all are doing very similar computations. Lastly, we extend our approach to the unranking of other classical combinatorial objects such as families counted by multinomial coefficients and k-permutations.

Lexicographic Unranking Algorithms for the Twelvefold Way

Amaury Curiel
Sorbonne Université, CNRS, LIP6 - UMR 7606, Paris, France

Antoine Genitrini
Sorbonne Université, CNRS, LIP6 - UMR 7606, Paris, France

Abstract

The Twelvefold Way represents Rota's classification, addressing the most fundamental enumeration problems and their associated combinatorial counting formulas. These distinct problems are connected to enumerating functions defined from a set of elements denoted by $\mathcal{N}$ into another one $\mathcal{K}$. The counting solutions for the twelve problems are well known. We are interested in unranking algorithms. Such an algorithm is based on an underlying total order on the set of structures we aim at constructing. By taking the rank of an object, i.e. its number according to the total order, the algorithm outputs the structure itself after having built it. One famous total order is the lexicographic order: it is probably the one that is the most used by people when one wants to order things. While the counting solutions for Rota's classification have been known for years it is interesting to note that three among the problems have yet no lexicographic unranking algorithm. In this paper we aim at providing algorithms for the last three cases that remain without such algorithms. After presenting in detail the solution for set partitions associated with the famous Stirling numbers of the second kind, we explicitly explain how to adapt the algorithm for the two remaining cases. Additionally, we propose a detailed and fine-grained complexity analysis based on the number of bitwise arithmetic operations.

Interest

- the ordered list C is

$$\mathrm{unrank}(0), \mathrm{unrank}(1), \mathrm{unrank}(2), \ldots$$

- $\mathrm{unrank}(p)$ yields a random object as long as p is a random integer
- the successor of a given object x can be computed as $\mathrm{unrank}(\mathrm{rank}(x) + 1)$
- to describe bijection between equinumerous classes

Even though these techniques are not efficient in general, they can provide hints for solutions to the underlying problems and enhance understanding of the concerned objects.

We are interested in ranking and unranking methods for simple yet non-trivial combinatorial classes : **permutations avoiding a pattern of length three**

The order we consider is the **lexicographic order** and some of its variations

If C is a set of length-n words over an ordered alphabet, then the rank of $w = w_1 w_2 \ldots w_n \in C$, in lexicographic order, is

$$\mathrm{rank}(w) = \sum_{i=1}^{n} |E_i|,$$

where for each i, E_i is the set of words $u_1 u_2 \ldots u_n \in C$ with $u_j = w_j$ for $j < i$ and $u_i < w_i$

$$\pi = 2341$$

$$\text{rank}(\pi) = \text{rank}(2341)$$

$$\pi = 2341$$

$$\text{rank}(\pi) = \text{rank}(2341) = 6$$

1234
1243
1342
1324
1423
1432

$$\pi = 2341$$

$$\text{rank}(\pi) = \text{rank}(2341) = 6 + 2$$

1234
1243
1342
1324
1423
1432
2134
2143

$$\pi = 2341$$

$$\text{rank}(\pi) = \text{rank}(2341) = 6 + 2 + 1$$

1234
1243
1342
1324
1423
1432
2134
2143
2314

$$\pi = 2341$$

$$\mathrm{rank}(\pi) = \mathrm{rank}(2341) = 6 + 2 + 1 = 9$$

1234
1243
1342
1324
1423
1432
2134
2143
2314
2341
...
...

Patterns in permutations

$\sigma = \sigma(1)\sigma(2)\cdots\sigma(k)$

$\pi = \pi(1)\pi(2)\cdots\pi(n)$

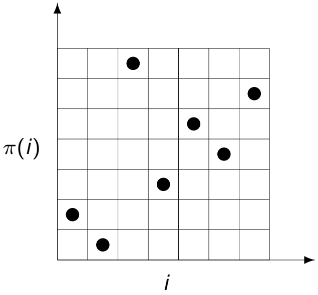
be permutations, $k \leq n$

Patterns in permutations

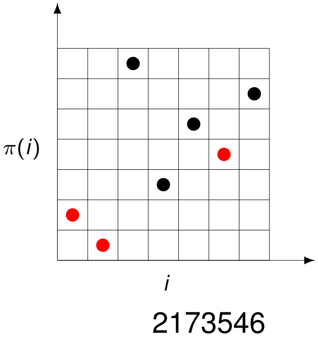
$$\sigma = \sigma(1)\sigma(2)\cdots\sigma(k)$$
$$\pi = \pi(1)\pi(2)\cdots\pi(n)$$

be permutations, $k \leq n$

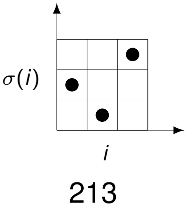
$$\pi \text{ contains } \sigma$$
$$\Updownarrow$$
$$\pi \text{ contains } \pi(i_1)\pi(i_2)\cdots\pi(i_k) \text{ order isomorphic to } \sigma$$



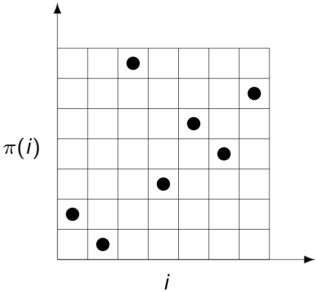
2173546



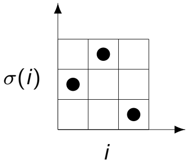
$\pi(i)$
i
2173546



$\sigma(i)$
i
213



2173546



231

123, 132, 213, 231, 312, 321
$\mathrm{Av}_n(\tau)$

123, 132, 213, 231, 312, 321

$\mathrm{Av}_n(\tau)$

123, 132, 213, 231, 312, 321

$\mathrm{Av}_n(\tau)$

If τ is a pattern of length 3, then $\mathrm{card}(\mathrm{Av}_n(\tau)) = c_n$

σ a length-k permutation

τ a length-ℓ

$$\pi = \sigma \oplus \tau$$

$\pi(1)\pi(2)\cdots\pi(k) = \sigma$

$\pi(k+1)\pi(k+2)\cdots\pi(k+\ell)$ is a permutation of $\{k+1, k+2, \ldots, k+\ell\}$ order isomorphic with τ

Similarly

$$\sigma \ominus \tau$$

$\oplus$ and $\ominus$

σ a length-k permutation

τ a length-ℓ

$$\pi = \sigma \oplus \tau$$

$\pi(1)\pi(2)\cdots\pi(k) = \sigma$

$\pi(k+1)\pi(k+2)\cdots\pi(k+\ell)$ is a permutation of $\{k+1, k+2, \ldots, k+\ell\}$ order isomorphic with τ

Similarly

$$\sigma \ominus \tau$$

σ a length-k permutation

τ a length-ℓ

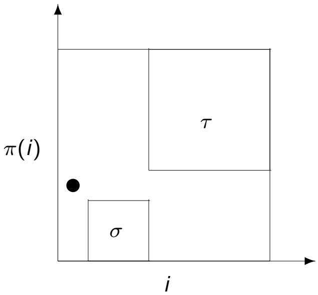
$$\pi = \sigma \oplus \tau$$

$\pi(1)\pi(2)\cdots\pi(k) = \sigma$

$\pi(k+1)\pi(k+2)\cdots\pi(k+\ell)$ is a permutation of $\{k+1, k+2, \ldots, k+\ell\}$ order isomorphic with τ

Similarly

$$\sigma \ominus \tau$$



$$\pi = (1 \ominus \sigma) \oplus \tau$$

231-avoiding permutations

$$\pi \text{ avoids } 231$$
$$\Updownarrow$$
$$\pi = (1 \ominus \sigma) \oplus \tau$$

where
σ and τ are 231-avoiding permutations

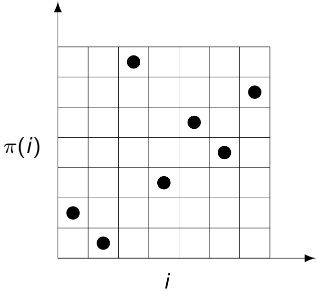
Reifegerste, On the diagram of 132-avoiding permutations, 2003

231-avoiding permutations

$$\pi \text{ avoids } 231$$
$$\Updownarrow$$
$$\pi = (1 \ominus \sigma) \oplus \tau$$

where
σ and τ are 231-avoiding permutations

Reifegerste, On the diagram of 132-avoiding permutations, 2003



The 231-avoiding permutation
$2173546 = (1 \ominus 1) \oplus (1 \ominus (1 \oplus ((1 \ominus 1) \oplus 1)))$

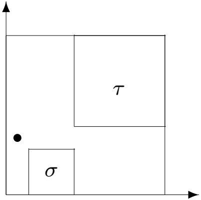
Proposition

Let $\pi = (1 \ominus \sigma) \oplus \tau$ *be 231-avoiding permutation, with*

$$\sigma \in \mathrm{Av}_\ell(231)$$
$$\tau \in \mathrm{Av}_k(231)$$

The rank of π*, in lexicographical order, is given recursively by*

$$\mathrm{rank}_{\mathrm{Av}_n(231)}(\pi) = \sum_{i=1}^{\pi(1)-1} c_i \cdot c_{n-i-1} + \mathrm{rank}_{\mathrm{Av}_\ell(231)}(\sigma) \cdot c_k + \mathrm{rank}_{\mathrm{Av}_k}(\tau)$$



- For $1 \leq i < \pi(1)$,

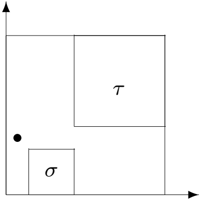
$$\sum_{i=1}^{\pi(1)-1} c_i \cdot c_{n-i-1} =$$

# of length-n 231-avoiders ρ with $\rho(1) < \pi(1)$

-

$$\mathrm{rank}_{\mathrm{Av}_\ell(231)}(\sigma) \cdot c_k =$$

# of 231-avoiders $\rho = (1 \ominus \sigma') \oplus \tau'$ with $\rho(1) = \pi(1)$ but $\sigma' < \sigma$



- For $1 \leq i < \pi(1)$,

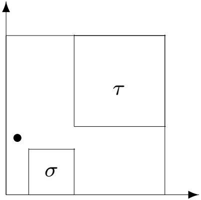
$$\sum_{i=1}^{\pi(1)-1} c_i \cdot c_{n-i-1} =$$

# of length-n 231-avoiders ρ with $\rho(1) < \pi(1)$

-

$$\mathrm{rank}_{\mathrm{Av}_\ell(231)}(\sigma) \cdot c_k =$$

# of 231-avoiders $\rho = (1 \ominus \sigma') \oplus \tau'$ with $\rho(1) = \pi(1)$ but $\sigma' < \sigma$



- For $1 \leq i < \pi(1)$,

$$\sum_{i=1}^{\pi(1)-1} c_i \cdot c_{n-i-1} =$$

# of length-n 231-avoiders ρ with $\rho(1) < \pi(1)$

- $$\mathrm{rank}_{\mathrm{Av}_\ell(231)}(\sigma) \cdot c_k =$$

# of 231-avoiders $\rho = (1 \ominus \sigma') \oplus \tau'$ with $\rho(1) = \pi(1)$ but $\sigma' < \sigma$

Proposition

Let n, p *be integers,* $0 \leq p < c_n$
$\mathrm{unrank}_{\mathrm{Av}_n(231)}(p)$ *is obtained recursively as follows*
Determine the smallest k *such that*

$$p < \sum_{i=0}^{k} c_i c_{n-i-1}$$

and denote

$s := \sum_{i=0}^{k-1} c_i c_{n-i-1}$,
$u := \lfloor \frac{p-s}{c_{n-k}} \rfloor$, *and*
$v := p - s - u \cdot c_{n-k}$

Then $\mathrm{unrank}_{\mathrm{Av}_n(231)}(p) = (1 \ominus \sigma) \oplus \tau$
where

$\sigma = \mathrm{unrank}_{\mathrm{Av}_{k-1}(231)}(u)$, *and*
$\tau = \mathrm{unrank}_{\mathrm{Av}_{n-k}(231)}(v)$

Proposition

Let n, p be integers, $0 \leq p < c_n$
$\mathrm{unrank}_{\mathrm{Av}_n(231)}(p)$ *is obtained recursively as follows*
Determine the smallest k such that ($k = \pi(1)$)

$$p < \sum_{i=0}^{k} c_i c_{n-i-1}$$

and denote

$s := \sum_{i=0}^{k-1} c_i c_{n-i-1}$,
$u := \lfloor \frac{p-s}{c_{n-k}} \rfloor$, *and*
$v := p - s - u \cdot c_{n-k}$

Then $\mathrm{unrank}_{\mathrm{Av}_n(231)}(p) = (1 \ominus \sigma) \oplus \tau$
where

$\sigma = \mathrm{unrank}_{\mathrm{Av}_{k-1}(231)}(u)$, *and*
$\tau = \mathrm{unrank}_{\mathrm{Av}_{n-k}(231)}(v)$

321-avoiding permutations

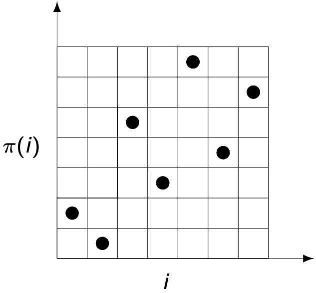
π avoids 321

$$\Updownarrow$$

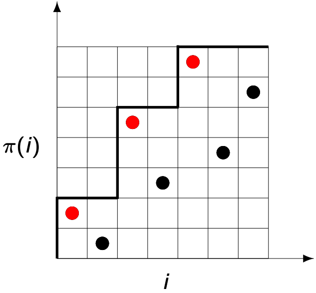
the subsequence of its left-to-right maxima

and

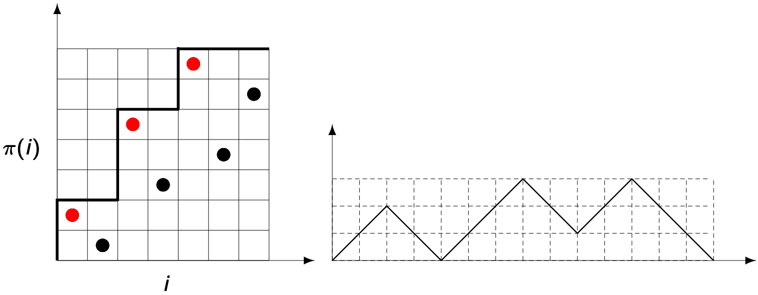
the subsequence of its remaining elements are increasing



$\pi = 2153746$

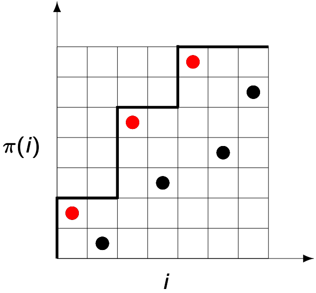


$\pi = 2153746$

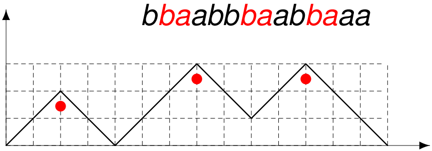


$\pi = 2153746$

Knuth, The Art of Computer Programming, III



$\pi = 2153746$



Knuth, The Art of Computer Programming, III

A 321-avoiding permutation is uniquely determined by its

- one-line representation
- left-to-right maxima
- corresponding Dyck path, or
- Dyck word

Proposition

$$\mathrm{Av}_n(321) \xrightarrow{\Psi} D_n$$

$$D_n \xrightarrow{\Psi^{-1}} \mathrm{Av}_n(321)$$

preserve the lexicographical order

Ballot sequences are prefixes of Dyck words

$B(i,j)$ the set of ballot sequences with i occurrences of b and j occurrences of a

$\mathrm{card}(B(i,j)) = t(i,j) = \binom{i+j}{i}\frac{i-j+1}{i+1}, 0 \leq j \leq i$

Catalan triangle `A009766`,

For $s(i,j) = \sum_{k=0}^{j} t(i,k)$ we have

$$s(i,j) = t(i+1,j)$$

Ballot sequences are prefixes of Dyck words

$B(i,j)$ the set of ballot sequences with i occurrences of b and j occurrences of a

$\mathrm{card}(B(i,j)) = t(i,j) = \binom{i+j}{i}\frac{i-j+1}{i+1}, 0 \le j \le i$

Catalan triangle `A009766`,

Similarly, $\bar{B}(i,j)$ is the set of *suffixes of Dyck words*

For $s(i,j) = \sum_{k=0}^{j} t(i,k)$ we have

$$s(i,j) = t(i+1,j)$$

$i \backslash j$	0	1	2	3	4	5	6	7
0	1							
1	1	1						
2	1	2	2					
3	1	3	5	5				
4	1	4	9	14	14			
5	1	5	14	28	42	42		
6	1	6	20	48	90	132	132	
7	1	7	27	75	165	297	429	429

The Catalan triangle $\left(t(i,j)\right)_{0 \le j \le i}$

$i \backslash j$	0	1	2	3	4	5	6	7
0	1							
1	1	1						
2	1	2	2					
3	1	3	5	5				
	1	4	9	14	14			
5	1	5	14	28	42	42		
6	1	6	20	48	90	132	132	
7	1	7	27	75	165	297	429	429

The Catalan triangle $\left(t(i,j)\right)_{0 \leq j \leq i}$

Proposition

For π a length-n 321-avoiding permutation with $(m_i, \ell_i)_{i=1}^k$

- *m_i's are the lrmp's of π*
- *$\ell_i = \pi(m_i)$ are the lrmv's*

$$\mathrm{rank}_{\mathrm{Av}_n(321)}(\pi) = \sum_{i=1}^{k} \big(s(n - m_i + 1, n - \max\{\ell_{i-1}, m_i\}) - s(n - m_i + 1, n - \ell_i + 1)\big)$$

$(\ell_0 = 0)$

Idea of proof

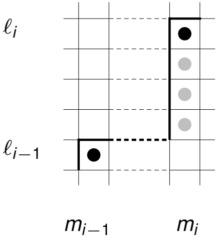
- π a 321-avoiding permutation and $(m_1, \ell_1), (m_2, \ell_2), (m_{i-1}, \ell_{i-1}), (m_i, \ell_i), \cdots$ its left-to-right maxima (position, value)

Idea of proof

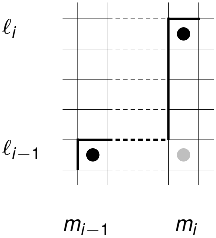
- π a 321-avoiding permutation and $(m_1, \ell_1), (m_2, \ell_2), (m_{i-1}, \ell_{i-1}), (m_i, \ell_i), \cdots$ its left-to-right maxima (position, value)
- σ a 321-avoiding permutation and $(m_1, \ell_1), (m_2, \ell_2), (m_{i-1}, \ell_{i-1}), \cdots$ with $\sigma(m_i) < \ell_i = \pi(m_i)$

Idea of proof

- π a 321-avoiding permutation and $(m_1, \ell_1), (m_2, \ell_2), (m_{i-1}, \ell_{i-1}), (m_i, \ell_i), \cdots$ its left-to-right maxima (position, value)
- σ a 321-avoiding permutation and $(m_1, \ell_1), (m_2, \ell_2), (m_{i-1}, \ell_{i-1}), \cdots$ with $\sigma(m_i) < \ell_i = \pi(m_i)$



keeping $m_1 \cdots m_i$



m_i is not longer a *ltrm*

Procedure for computing the Dyck word of semilength n and rank p

procedure UNRANKDYCK(n, p : integer)
    global d : array
    $i \leftarrow n, j \leftarrow n$
    for k **from** 1 **to** $2n$ **do**
        if $i = j$ **or** $p \geq \overline{t}(i-1, j)$ **then**
            $d_k \leftarrow \mathrm{b}$; $p \leftarrow p - \overline{t}(i-1, j)$; $i \leftarrow i - 1$
        else $d_k \leftarrow \mathrm{a}$; $j \leftarrow j - 1$
        end if
end procedure

Here $\overline{t}(i, j) = \mathrm{card}(\overline{B}(i, j))$, so $\overline{t}(i, j) = t(j, i)$

Unranking Dyck words

Procedure for computing the Dyck word of semilength n and rank p

procedure UNRANKDYCK(n, p : integer)
  global d : array
  $i \leftarrow n, j \leftarrow n$
  for k **from** 1 **to** $2n$ **do**
    if $i = j$ **or** $p \geq \bar{t}(i-1, j)$ **then**
      $d_k \leftarrow \mathrm{b}$; $p \leftarrow p - \bar{t}(i-1, j)$; $i \leftarrow i - 1$
    else $d_k \leftarrow \mathrm{a}$; $j \leftarrow j - 1$
    end if
end procedure

Here $\bar{t}(i,j) = \mathrm{card}(\overline{B}(i,j))$, so $\bar{t}(i,j) = t(j,i)$

Kása, Generating and ranking of Dyck words, 2009

Proposition

For n, p with $0 \leq p < c_n$ $\mathrm{unrank}_{\mathrm{Av}_n(321)}(p)$ is obtained as follows :

- *run the* UNRANKDYCK *procedure to construct the Dyck word w of semilength n and rank p*
- *construct the set $\{(m_i, \pi(m_i))\}_{i=1}^{k}$ of left-to-right maxima (values, positions) of the permutation $\Psi^{-1}(w)$*
- *Finally, construct the permutation $\mathrm{unrank}_{\mathrm{Av}_n(321)}(p)$*

Other length-3 patterns ?

321 231

123 132 213 312

Other length-3 patterns ?

321 231

123 132 213 312

$$321 \xleftrightarrow[\mathfrak{c}]{\mathfrak{r}} 123$$

Other length-3 patterns ?

321 231

123 132 213 312

$$321 \xleftrightarrow[\mathfrak{c}]{\mathfrak{r}} 123$$

$$\begin{array}{ccc} 231 & \xleftrightarrow{\mathfrak{r}} & 132 \\ \mathfrak{c} \updownarrow & & \updownarrow \mathfrak{c} \\ 213 & \xleftrightarrow{\mathfrak{r}} & 312 \end{array}$$

Proposition

Let $\tau \in \{213, 123\}$ *be a pattern* ($\mathfrak{c}(\tau) \in \{231, 321\}$). *Then,*

- *for* $\pi \in \mathrm{Av}_n(\tau)$

$$\mathrm{rank}_{\mathrm{Av}_n(\tau)}(\pi) = c_n - \mathrm{rank}_{\mathrm{Av}_n(\mathfrak{c}(\tau))}(\mathfrak{c}(\pi)) - 1$$

- *for* p, $0 \leq p < c_n$

$$\mathrm{unrank}_{\mathrm{Av}_n(\tau)}(p) = \mathrm{unrank}_{\mathrm{Av}_n(\mathfrak{c}(\tau))}(c_n - p - 1)$$

Proposition

- *for* $\pi \in \mathrm{Av}_n(\tau)$

$$\mathrm{rank}^*_{\mathrm{Av}_n(\tau)}(\pi) = \mathrm{rank}_{\,\mathrm{Av}_n(\mathfrak{r}(\tau))}(\mathfrak{r}(\pi))$$

- *for* p, $0 \le p < c_n$

$$\mathrm{unrank}^*_{\mathrm{Av}_n(\tau)}(p) = \mathfrak{r}(\mathrm{unrank}_{\,\mathrm{Av}_n(\mathfrak{r}(\tau))}(p))$$

Open problems

- extend these results to multiple patterns of length 3
- to avoidance of patterns of size greater than 3
- the efficient (time and space) implementation of these methods
- other order relations (Gray code order ?)

Bibliography II

Knuth, The Art of Computer Programming, III

Reifegerste, On the diagram of 132-avoiding permutations, 2003

Kása, Generating and ranking of Dyck words, 2009

Grazie per la vostra attenzione !