Pré-Publicações do Departamento de Matemática
Universidade de Coimbra
Preprint Number 13–31

GRAY CODES AND LEXICOGRAPHICAL COMBINATORIAL GENERATION FOR NONNESTING AND SPARSE NONNESTING SET PARTITIONS

ALESSANDRO CONFLITTI AND RICARDO MAMEDE

Abstract: We present combinatorial Gray codes and explicit designs of efficient algorithms for lexicographical combinatorial generation of the sets of nonnesting and sparse nonnesting set partitions of length n.

AMS Subject Classification (2000): 05A18, 68R05, 68W99, 94B25.

1. Introduction

One of the earliest problem addressed on the topic of *combinatorial algorithms* was to efficiently generate elements in a specific combinatorial class in such a way that each item is generated exactly once, hence producing a listing of all objects in the considered class. Both for theoretical and practical purposes, in order for such listing to be meaningful or useful, even for objects of moderate size, combinatorial generation methods must be extremely efficient, and the order in which objects appear in such list should satisfy some significant conditions, see e.g. [3, 4, 25, 27, 34, 41, 43, 45, 51].

Specifically, very interesting and widely studied orderings for combinatorial generation are the lexicographic order and any ordering that gives rise to a Gray code for the considered combinatorial class.

Some of the advantages of the lexicographic order are that for any combinatorial class it is easy to define, to visualize and work with, and it is surprisingly useful in a variety of contexts. For instance, for many combinatorial objects, the fastest known algorithms for listing, ranking and unranking are with respect to lexicographic order, and many algorithms that are not overtly lexicographic have anyway some underlying lexicographic structure, see e.g. [43].

Received July 19, 2013.
This work was partially supported by the Centro de Matemática da Universidade de Coimbra (CMUC), funded by the European Regional Development Fund through the program COMPETE and by the Portuguese Government through the FCT - Fundação para a Ciência e a Tecnologia under the project PEst-C/MAT/UI0324/2011.

A combinatorial Gray code is any method for generating combinatorial objects such that successive objects differ in a some pre–specified, usually small, way. For instance, given a combinatorial class one can define a *distance* in such class and define a Gray code as a list of all objects in the considered class such that the distance between two consecutive objects is 1. Usually this is done associating a binary numeral system to the combinatorial class and considering the Hamming distance. Such theory of Gray codes has a vast range of applications in several areas such as hardware and software testing, thermodynamic, biology and biochemistry; see [3, 45] and the references therein.

There exist results for (lexicographical) combinatorial generation of several classes of permutations and set partitions, see e.g. [8, 20, 22, 32, 36, 37, 50, 53, 54], but so far there are no direct, explicit efficient algorithms for the lexicographical combinatorial generation of noncrossing and nonnesting partitions. Specifically, all known results up to now for generating (not lexicographically) nonnesting partitions use a generating tree approach which often involve passing through Young tableaux or lattice paths, see [10] and the references therein.

Likewise, several results exist on Gray codes for different families of permutations, set partitions, and Catalan classes of objects such as plane trees, Dyck paths, parallelogram polyominoes, Catalan permutations, etc., see e.g. [3, 5, 6, 7, 9, 14, 15, 16, 21, 26, 29, 30, 44]. Furthermore, Gray codes for noncrossing partitions have recently been obtained in [19], but so far nothing is known for nonnesting partitions.

Noncrossing partitions are a class of combinatorial objects widely studied, see e.g. [1, 2, 11, 12, 13, 17, 23, 28, 29, 31, 40, 42, 46], and among other things they have applications to the theory of free probability, see [35, 47, 48]. In particular, there exist several bijections between noncrossing and nonnesting partitions, see e.g. [12, 13, 17, 23, 28, 42], so it is very natural to wonder if all results which hold for noncrossing partitions hold for nonnesting partitions as well. Unfortunately, nonnesting partitions are much more mysterious and intricate, and plenty of results valid for noncrossing partitions do not translate to nonnesting partitions.

In this paper we present Gray codes for the two classes of nonnesting partitions and sparse partitions, therefore extending the results for noncrossing partitions given in [19], and we explicitly design efficient algorithms for lexicographical combinatorial generation of the two sets of nonnesting and sparse nonnesting set partitions.

2. Preliminaries and notations

A set partition π of $[n] := \{1, \ldots, n\}$, $n \geq 1$, is a collection of nonempty disjoint subsets $B_1, \ldots, B_n$ of $[n]$, called *blocks*, whose union is $[n]$. We call a pair of integers (i, j) an *arc* of a set partition π if i and j occur in the same block and j is the least element of the block greater than i; the first coordinate i of an arc (i, j) is defined to be an *opener* and the second coordinate j is defined to be a *closer* of the set partition π. For example, $\pi = (\{1, 2, 4\}, \{3, 5\}, \{6\})$ is a set partition of [6] with three blocks $B_1 = \{1, 2, 4\}$, $B_2 = \{3, 5\}$ and $B_3 = \{6\}$, and the set of arcs is $\{(1, 2), (2, 4), (3, 5)\}$, $\{1, 2, 3\}$ are the openers and $\{2, 4, 5\}$ the closers of π. So it is possible for an integer to be at the same time both an opener and a closer. The set of all set partitions of $[n]$ will be denoted by $P(n)$. A partition is said to be in *standard form* if it is written as $\pi = B_1/B_2/\cdots/B_t$, where the blocks are listed in ascending order according to their smallest element. Any partition of $[n]$ can be written in sequential form, that is, as a word $\pi = \pi_1\pi_2\cdots\pi_n$ of length n over some alphabet such that $\pi_i = \pi_j$ if and only if i, j lie in the same block. A partition has many sequential forms, one of them being the *canonical sequential form*, in which the alphabet is $[n]$ and where $\pi_i = j$ if $\pi_i \in B_j$, for all j. Such words are known as *restricted growth sequences*, see [49], in the combinatorial literature and they are characterized by the following properties: $\pi_1 = 1$ and

$$\pi_i \leq 1 + \max\{\pi_1, \ldots, \pi_{i-1}\}, \qquad \text{for } i = 2, \ldots, n.$$

The latter condition restricts the growth of the letters in the word by requiring π_i to be at most one more than the maximum of the previous letters. Note that each partition has a unique canonical sequential form: for instance, 12123 is the canonical sequential form, and 13/24/5 is the standard form of the partition $\pi = (\{1, 3\}, \{2, 4\}, \{5\})$. In the following, we shall represent set partitions as words using their (canonical) sequential forms.

Given two partitions π, ω of $[n]$, their distance, see [18], is defined as the minimum number of letters that must be deleted from $[n]$ so that the two residual induced partitions are identical:

$$D(\pi, \omega) = \min\{|A^c| : \emptyset \subset A \subseteq [n], \pi_{|_A} = \omega_{|_A}\},$$

where A^c is the complement of A in $[n]$ and $\pi_{|_A}$ is the partition of A induced by π. In other words, $D(\pi, \omega)$ is equal to the minimum number of letters that must be moved between blocks of π so that the resulting partition is ω. In particular, $D(\pi, \omega) = 1$ if the two partitions differ by taking exactly one element from one block and moving it into another one, possible creating a singleton, i.e. a block with a single element. When a letter moves from block B_i to block B_j it may happen that the relative position of the blocks in the partition are changed; this means that we may have two partitions with distance 1 whose canonical sequential forms differ in more that one letter. For instance, the partitions $\pi = 11123$ and $\omega = 12134$ have distance 1. Note however that if we represent π as $\pi = 11134$, then between this one and the canonical sequential representation of ω only one letter is changed. Therefore, two partitions represented in canonical sequential form have distance 1 if and only if they either differ by a single letter, or there is a sequential representation for one of them such that this representation and the canonical sequential form of the other differ by a single letter.

A *nonnesting* set partition of $[n]$ is a partition of $[n]$ such that if $a < b < c < d$ and a, d are consecutive elements of a block, then b and c cannot both be contained in some other block. We denote by $\mathrm{NN}(n)$ the set of all nonnesting set partitions of $[n]$, and by $\mathrm{NN}(n, k)$ the set of nonnesting set partitions having exactly k blocks, $1 \leq k \leq n$. The number of nonnesting set partitions of $[n]$ is equal to

$$|\,\mathrm{NN}(n)| = \frac{1}{n}\binom{2n}{n-1},$$

and the number of nonnesting set partitions of $[n]$ with exactly k blocks is equal to

$$|\,\mathrm{NN}(n,k)| = \frac{1}{n}\binom{n}{k}\binom{n}{k-1},$$

the Catalan and Narayama number, respectively.

We can endow the sets $\mathrm{NN}(n)$ and $\mathrm{NN}(n, k)$ with a graph structure by declaring two partitions adjacent if their distance is 1. We use the same notation for the set of nonnesting partitions and the corresponding graph. A Hamilton path with distance 1 in $\mathrm{NN}(n)$ or $\mathrm{NN}(n, k)$ corresponds to an exhaustive sequence of the nonnesting partitions of $[n]$ such that the distance between two successive partitions is 1, and thus it gives a Gray code for these objects. If this path is closed then obviously we have a Hamilton cycle.

There are several bijections between noncrossing and nonnesting set partitions (see, for example [2, 13, 17, 28, 42]), and since in [19] a Gray code for noncrossing partitions is presented, it is tempting to try employing these bijections in order to obtain a Gray code for nonnesting partitions. But, as referred in [45], a Gray code for a combinatorial class is intrinsically bound to the representation of objects in the class, and in the present case, the Gray code is not preserved under bijection.

3. The Gray codes

One of the oldest and more important tools used to produce a Gray code for general set partitions of $[n]$ is the recursive algorithm described in [21], which is also one of the main ingredient for some of the constructions in [19]. We show that the ideas behind that recursive algorithm can be used to construct Gray codes for a larger set of partitions, namely sparse partitions and nonnesting partitions. A set partition of $[n]$ is said to be *sparse*, see [33], if for every $i \in [n-1]$ the values i and $i+1$ lie in two distinct blocks. Denote by $\mathrm{SP}(n)$ the set of sparse set partitions of $[n]$; it is easy to check that the cardinality of $\mathrm{SP}(n)$ is equal to the number of set partitions of $[n-1]$. A bijection between these two sets can be construct as follows (see [11, 24, 39]). Take a set partition π of $[n-1]$, and consider each factor of two or more consecutive letters in the canonical sequential form of π. For each one of these factors, say $\pi_i \pi_{i+1} \cdots \pi_j$, replace with n the letters π_{i+2k+1} for k such that $i+1 \leq i+2k+1 \leq j$, and adjoin n at the right end side of π. The resulting partition is clearly in $\mathrm{SP}(n)$, and this is a bijection: to reconstruct the original partition of $[n-1]$ just remove the n–th letter and replace each other letter n by the letter sitting on its left.

Let $\mathrm{T}(n)$ be one of the sets $\mathrm{SP}(n)$ or $\mathrm{NN}(n)$ and let π be a partition in $\mathrm{T}(n)$. Following the terminology of [21], by a *children* of π we mean any partition of $[n+1]$ obtained from π by inserting a letter at the right of the rightmost letter of π such that the resulting partition is in $\mathrm{T}(n+1)$. We denote by $C_{\mathrm{T}}(\pi)$ the set of all children of π and we remark that this set is never empty since the child π^* obtained by adjoining $n+1$ on the right of π, is in $C_{\mathrm{T}}(\pi)$. Moreover, every partition in $\mathrm{T}(n+1)$ has a unique parent in $\mathrm{T}(n)$ obtained by removing the letter in the $(n+1)$-th position.

The following Lemmata are immediate from the definitions, and they are a generalization of those presented in [19].

Lemma 3.1. *The distance between two children of the same partition is* 1.

Lemma 3.2. *If the distance between two partitions π and ω of $[n]$ is 1, then the same holds for the partitions π^* and ω^*.*

These two Lemmata show that all siblings of a partition of $\mathrm{T}(n)$ induce a complete subgraph of $\mathrm{T}(n+1)$, and that the sets $C_{\mathrm{T}}(\pi)$ and $C_{\mathrm{T}}(\omega)$, of siblings of two adjacent partitions $\pi, \omega \in \mathrm{T}(n)$, are linked by at least a pair of adjacent partitions. In fact something stronger is true, since the following Lemma assures the existence of at least one more pair of adjacent partitions linking the two sets.

Lemma 3.3. *If the distance between partitions $\pi, \omega \in \mathrm{T}(n)$ is 1, then there exist children $\widehat{\pi} \neq \pi^*$ of π and $\widehat{\omega} \neq \omega^*$ of ω such that their distance is also 1.*

Proof: There exist sequential representations of π and ω, not necessarily both in canonical sequential form, such that $\pi = \pi_1 \cdots \pi_n$ and $\omega = \omega_1 \cdots \omega_n$, with $\pi_\ell = \omega_\ell$, for all $\ell \neq i$, and $\pi_i \neq \omega_i$ for some $i \in [n]$. If $\mathrm{T}(n) = \mathrm{NN}(n)$, consider the partitions $\widehat{\pi} = \pi_1 \cdots \pi_n \pi_n$ and $\widehat{\omega} = \omega_1 \cdots \omega_n \pi_n$. If $i \neq n$ then $\pi_n = \omega_n$ and therefore $\widehat{\pi}$ and $\widehat{\omega}$ are nonnesting with distance 1. On the other hand, if $i = n$ then we must have $\pi_{n-1} = \omega_{n-1}$, and again this implies that $\widehat{\pi}$ and $\widehat{\omega}$ are nonnesting and have distance 1.

Consider now the case $\mathrm{T}(n) = \mathrm{SP}(n)$. If $i \neq n-1$, consider the partitions $\widehat{\pi} = \pi_1 \cdots \pi_n \pi_{n-1}$ and $\widehat{\omega} = \omega_1 \cdots \omega_n \pi_{n-1}$, and if $i = n-1$ consider $\widehat{\pi} = \pi_1 \cdots \pi_n \pi_{n-2}$ and $\widehat{\omega} = \omega_1 \cdots \omega_n \pi_{n-2}$. In both cases the partitions $\widehat{\pi}$ and $\widehat{\omega}$ are sparse and their distance is 1. ∎

We give now a recursive construction of a Gray code with distance 1 for the set $\mathrm{T}(n)$. The construction is trivial for $n = 1, 2$ and such a Gray code for $\mathrm{SP}(3)$ and $\mathrm{NN}(3)$ is presented in Figures 1 and 2, respectively.

Suppose, inductively, that we know the list $\mathcal{L}_n$, the Gray code sequence of all partitions in $\mathrm{T}(n)$ with distance 1, for some $n \geq 3$. Construct the list $\mathcal{L}_{n+1}$ as follows: take the first partition π in $\mathcal{L}_n$ and list all its children, starting with π^* and ending with a partition $\widehat{\pi}$ adjacent to some child $\widehat{\omega} \neq \omega^*$ of the second partition ω in $\mathcal{L}_n$. Go to $\widehat{\omega}$ and then list the remaining children of ω ending with ω^*. From here the process repeats itself, starting with the next partition in $\mathcal{L}_n$, until we get to the last partition in $\mathcal{L}_n$.

Theorem 3.4. *The list $\mathcal{L}_n$ is a Gray code with distance 1 for the partitions in $\mathrm{T}\,(n)$, for $n \geq 3$.*

Proof: Note that $\mathcal{L}_n$ is a list of all partitions in $\mathrm{T}(n)$, since every partition in this set has a parent in $\mathrm{T}(n-1)$ and we start the induction with the

$\mathcal{L}_3$	$\mathcal{L}_4$	$\mathcal{L}_5$
121	1213	12134
		12132
		12131
	1212	12121
		12123
123	1232	12324
		12321
		12323
	1231	12313
		12312
		12314
	1234	12345
		12343
		12342
		12341

FIGURE 1. Gray codes for SP(n), $n = 3, 4, 5$

list $\mathcal{L}_3$ given in Figures 1 and 2, which are the complete list of sparse and nonnesting partitions of [3], respectively. Given two consecutive partitions in $\mathcal{L}_n$ two cases can happen: either they are siblings or they are children from adjacent partitions in $\mathcal{L}_{n-1}$. In the first case Lemma 3.1 assures that their distance is 1, and in the second case their construction and the fact that the distance is 1 are guaranteed by Lemmata 3.2 and 3.3. ■

The above construction can be rearranged in order to give Hamilton cycles for SP(n) and NN(n). We start with the set of sparse partitions of [n], for which we need to specify some special partitions.

Definition 3.5. *For $n \geq 4$ and $\ell \in [n-2]$, let γ^n, δ^n and η^n_ℓ be the following sparse partitions, written in canonical sequential form: $\gamma^n = 12\cdots n$, $\delta^n = 12134\cdots(n-1)$ and $\eta^n_\ell = 12\cdots(n-1)\ell$.*

Note that the distance between the partitions γ^n and δ^n, and between γ^n and η^n_ℓ is 1. Also, we have $(\gamma^n)^* = \gamma^{n+1}, (\delta^n)^* = \delta^{n+1}$ and $(\eta^n_\ell)^* = 12\cdots(n-1)\ell n$. The next Lemma highlights the connections between the children of γ^n and η^n_ℓ.

$\mathcal{L}_3$	$\mathcal{L}_4$	$\mathcal{L}_5$		
111	1112	11123	12334	12223
	1111	11122	12333	12222
121	1211	11121	12343	12232
	1212	11111	12342	12234
	1213	11112	12344	12233
123	1233	12113	12345	11233
	1234	12111	12341	11234
	1231	12121	12311	11232
	1232	12122	12312	11231
122	1222	12123	12313	11211
	1223	12133	12314	11213
112	1123	12131	12324	11212
	1121	12132	12322	11222
	1122	12134	12323	11223

FIGURE 2. Gray codes for NN(n), $n = 3, 4, 5$, where the columns are read top to bottom, and left to right.

Lemma 3.6. *There are $n-1$ pairs of partitions $\widehat{\gamma}$ and $\widehat{\eta}$ with distance 1, where $\widehat{\gamma}$ is a child of γ^n and $\widehat{\eta}$ is a child of η^n_ℓ, for some $\ell \in [n-2]$.*

Proof: The children of γ^n are the partitions

$$\eta^{n+1}_j = 12\cdots nj, \qquad \text{for } j \in [n-1] \qquad \text{and } (\gamma^n)^* = 12\cdots n(n+1),$$

and the children of η^n_ℓ are

$$12\cdots(n-1)\ell j, \qquad \text{for } j \in [n] \setminus \{\ell\}.$$

Therefore, we can consider the pair $(\gamma^n)^*$ and $(\eta^n_\ell)^* = 12\cdots(n-1)\ell n$ of children of γ^n and η^n_ℓ, and also the pairs $\widehat{\gamma}$ and $\widehat{\eta}$, where $\widehat{\gamma} = 12\cdots nj$ and $12\cdots(n-1)\ell j$, for all $j \in [n] \setminus \{\ell, n\}$, all of which have distance 1. ■

To construct a Hamilton cycle in SP(n) we need to consider the parity of SP($n-1$). Note that SP(3) is just the cycle $123, 121$ with distance 1.

Theorem 3.7. *For $n \geq 4$ the graph SP (n) has a Hamilton cycle with distance 1, where γ^n is adjacent to δ^n and η^n_ℓ, for some $\ell \in [n-2]$.*

Proof: The proof is by induction on $n \geq 4$. The case $n = 4$ is displayed in Figure 1. Assume the result for $n \geq 4$ and let

$$\mathcal{S}_n = \gamma^n, \delta^n, \pi, \ldots, \omega, \eta^n_\ell$$

be a Hamilton cycle in the graph $\mathrm{SP}(n)$, for some $\ell \in [n-2]$. Start the Hamilton cycle $\mathcal{S}_{n+1}$ in $\mathrm{SP}(n+1)$ with the child $(\gamma^n)^*$ of γ^n, and then go to $(\delta_n)^*$. Go through all the remaining children of δ^n (the order does not matter because each one has distance 1 from all others, see Lemma 3.1; the important point is only to choose wisely the first and the last child to visit) ending with a partition $\widehat{\delta}^n$ adjacent to some child $\widehat{\pi} \neq \pi^*$ of π. Next, follow the construction given for $\mathcal{L}_n$ till we get to ω^*.

If the number of elements of $\mathcal{S}_n$ is odd, then ω^* is the first child in $C_{\mathrm{SP}(n)}(\omega)$ to be linked. As before, go through all the remaining elements of this set (again, the order does not matter because each one has distance 1 from all others, see Lemma 3.1, so it is just important to choose correctly the first and the last child to visit) ending with a partition $\widehat{\omega}$ adjacent to some child $\widehat{\eta} = 12\cdots(n-1)\ell j$ of η^n_ℓ, for some $j \in [n] \setminus \{\ell, n\}$. Go through the remaining siblings of $\widehat{\eta}$ ending on $12\cdots(n-1)\ell k$, for some $k \in [n] \setminus \{\ell, j, n\}$. By Lemma 3.6, this partition exists and can be linked to the child $12\cdots nk$ of γ^n.

Go through all the remaining children of γ^n (again, the order does not matter because each one has distance 1 from all others, see Lemma 3.1, so it is important only to choose correctly the first and the last child to visit) ending with $(\gamma^n)^*$ to close the cycle.

Otherwise, if $\mathcal{S}_n$ has an even number of elements, follow the construction for the even case till we get to partition ω^*, which is now the last child of ω to be inserted in the cycle $\mathcal{S}_{n+1}$. Next, go to $(\eta^n_\ell)^*$ and go through all the remaining children of η^n_ℓ (in any order since each one has distance 1 from all others, see Lemma 3.1; the only important point is the choose wisely the first and the last child to visit) ending with some $\widehat{\eta} = 12\cdots(n-1)\ell k$, for some $k \in [n-1] \setminus \{\ell\}$. Again Lemma 3.6 guarantees that this is feasible and that this partition can be followed by the child $12\cdots nk$ of γ^n. Go through the remaining children of γ^n, ending on $(\gamma^n)^*$ to close the cycle.

As in the proof of Theorem 2, it is easy to check that in both cases the cycle $\mathcal{C}_{n+1}$ contains all partitions in $\mathrm{NN}(n+1)$ and that the distance between any two consecutive partitions is 1. ■

Consider next the problem of constructing a Hamilton cycle for the set of nonnesting partitions of $[n]$. In order to solve it, we need to consider the parity of the cardinality of set $\mathrm{NN}(n-1)$, and two special partitions.

Definition 3.8. *For $n \geq 3$, define the partitions $\lambda^n = 1^n$ and $\nu^n = 121^{n-2}$.*

The partitions λ^n and ν^n are nonnesting and their children are

$$C_{\mathrm{NN}(n)}(\lambda^n) = \{\lambda^{n+1}, (\lambda^n)^*\} \quad \text{and} \quad C_{\mathrm{NN}(n)}(\nu^3) = \{1211, 1212, 1213\},$$

with $C_{\mathrm{NN}(n)}(\nu^n) = \{\nu^{n+1}, (\nu^n)^*\}$ for $n > 4$. Note also that

$$C_{\mathrm{NN}(n)}((\lambda^{n-1})^*) = \{1^{n-1}23, 1^{n-1}22, 1^{n-1}21\}.$$

Theorem 3.9. *For $n \geq 3$ there exists a Hamilton cycle with distance 1 in $\mathrm{NN}(n)$, where λ^n is adjacent to ν^n and $(\lambda^{n-1})^*$.*

Proof: The construction is done by induction on $n \geq 3$, following closely the construction of $\mathcal{L}_n$. The case $n = 3$ is shown in Figure 2. Assume that the result holds for $n \geq 3$ and let

$$\mathcal{C}_n = \lambda^n, \nu^n, \pi, \ldots, \omega, (\lambda^{n-1})^*$$

be a Hamilton cycle with distance 1 in $\mathrm{NN}(n)$. We can now construct the Hamilton cycle $\mathcal{C}_{n+1}$, which depends on the parity of the number of elements of $\mathrm{NN}(n)$.

If $|\,\mathrm{NN}(n)|$ is even, start the cycle with the child λ^{n+1} of λ^n. The next partition in $\mathcal{C}_{n+1}$ is ν^{n+1} followed by its only sibling $(\nu^n)^*$. From here go to π^* and follow the construction given for $\mathcal{L}_n$ till we get to ω^*. The condition of being even guarantees that this partition is the first child of $C(\omega)$ in the list $\mathcal{C}_{n+1}$. Go through all the remaining children of $C(\omega)$ (the order does not matter because each one has distance 1 from all others, see Lemma 3.1; only it is crucial to choose correctly the first and the last child to visit) ending with a partition adjacent to one of the two children of $(\lambda^{n-1})^*$, other than $((\lambda^{n-1})^*)^*$. Go to the next child in the set $C((\lambda^{n-1})^*) \setminus \{((\lambda^{n-1})^*)^*\}$ and then to $((\lambda^{n-1})^*)^*$. Next, go to $(\lambda^n)^*$ and finally to λ^{n+1}, thus completing the Hamilton cycle.

If $|\,\mathrm{NN}(n)|$ is odd follow the construction of the even case from λ^{n+1} till the partition ω^*. Now, the odd condition guarantees that this is the last child of $C(\omega)$ in the list. Next, place the children $((\lambda^{n-1})^*)^*, 1^{n-1}21, 1^{n-1}22$ of $C(\nu^n)$, by this order, and then go to $(\lambda^n)^*$ followed by λ^{n+1} to complete the cycle.

As in the proof of Theorem 2, it is easy to check that in both cases the cycle $\mathcal{C}_{n+1}$ contains all partitions in $\mathrm{NN}(n+1)$ and that the distance between any two consecutive partitions is 1. ■

4. Lexicographical combinatorial generation

Usually the study of set partitions of $[n]$ having a specific property, e.g. being noncrossing , nonnesting, sparse, etc., is refined by considering all such partitions which have exactly k blocks. In this sections we consider arcs instead of blocks, investigating nonnesting and sparse nonnesting partitions of $[n]$ which have exactly k arcs, since in this way all results are much easier to state and prove. However we remark that we can safely choose arcs instead of blocks without losing in generality because of the following result.

Proposition 4.1. *Consider $\pi \in P(n)$, where $n \geq 1$ is an integer, and let* $\mathrm{block}(\pi)$ *be the set of its blocks and* $\mathrm{arc}(\pi)$ *be the set of its arcs; then the equality*

$$\#\,\mathrm{block}(\pi) + \#\,\mathrm{arc}(\pi) = n$$

holds.

Proof: By definition,

$$\sum_{B \in \mathrm{block}(\pi)} \#B = n,$$

and obviously in any $B \in \mathrm{block}(\pi)$ there are exactly $\#B - 1$ arcs of π, viz. B contributes to $\mathrm{arc}(\pi)$ with $\#B - 1$ arcs. For any $B \in \mathrm{block}(\pi)$ let $\mathrm{arc}_B(\pi)$ be the set of arcs in π which are inside the block B; then by definition of arc we have

$$\#\,\mathrm{arc}(\pi) = \sum_{B \in \mathrm{block}(\pi)} \#\,\mathrm{arc}_B(\pi),$$

therefore

$$\begin{aligned} n &= \sum_{B \in \mathrm{block}(\pi)} \#B = \sum_{B \in \mathrm{block}(\pi)} (1 + \#\,\mathrm{arc}_B(\pi)) \\ &= \#\,\mathrm{block}(\pi) + \sum_{B \in \mathrm{block}(\pi)} \#\,\mathrm{arc}_B(\pi) = \#\,\mathrm{block}(\pi) + \#\,\mathrm{arc}(\pi)\,. \end{aligned}$$

■

Hence defining $\mathrm{SNN}\,(n)$ to be the set of sparse nonnesting partitions of $[n]$, $\mathrm{NN}\,(n,k)$ and $\mathrm{SNN}\,(n,k)$ to be the sets of nonnesting, respectively sparse nonnesting, partitions of $[n]$ with exactly k blocks, and $\mathrm{NN}\,[n,k]$ and $\mathrm{SNN}\,[n,k]$ to be the sets of nonnesting, respectively sparse nonnesting, partitions of $[n]$ with exactly k arcs, we get the two following set equality: $\mathrm{NN}\,[n,k] = \mathrm{NN}\,(n,n-k)$ and $\mathrm{SNN}\,[n,k] = \mathrm{SNN}\,(n,n-k)$, for $k = 0,\dots,n-1$.

Now we present a characterization of the set of nonnesting partitions which, albeit very simple to prove, will demonstrate itself to be the key ingredient in designing an efficient algorithm for the combinatorial generation of such set.

Lemma 4.2. *Let $n \geq 2$ be an integer number and π a set partition of $[n]$ with*
$(o_1, c_1), \dots, (o_k, c_k)$ the couples of all its openers and corresponding closers, i.e. all the arcs of π.

Then the two following statements are equivalent:

(1) $\pi \in \mathrm{NN}\,(n)$,
(2) *if* $o_1 < \ldots < o_k$ *then* $c_1 < \ldots < c_k$.

Proof: Obviously without loss of generality up the a rearrangement we can always suppose that $o_1 < \ldots < o_k$. From the definition it is straightforward that there is a nesting if and only if $c_i \geq c_j$ for $i < j$. ■

In general, given $\pi \in P(n)$ there is no univocal natural ordering of the arcs because it depends whether openers or closers are considered: for instance, if $\pi = \{\{1,4\},\{2,3\}\}$, then $(1,4),(2,3)$ is the natural arrangement according to openers, but according to closers the natural one is $(2,3),(1,4)$. However, because of the previous Lemma, we see that there is no such ambiguity when $\pi \in \mathrm{NN}\,(n)$.

Given $\pi \in \mathrm{NN}\,(n)$, let $(o_1, c_1), \dots, (o_k, c_k)$ be all its arcs ordered according to openers (or closers, because of Lemma 4.2), i.e. if $i < j$ then $o_i < o_j$; obviously we can univocally represent π with the data structure $\mathrm{Arch}\,(\pi)$ given by the following integer sequence: if π has no arcs, i.e. $\pi = \{\{1\},\dots,\{n\}\}$, the partitions made by all singleton blocks, then $\mathrm{Arch}\,(\pi) := (\emptyset)$, whereas if π admits some arcs then $\mathrm{Arch}\,(\pi) := (o_1, c_1, \dots, o_k, c_k)$.

On the other hand, given an integer $n \geq 1$ and an integer sequence $\mathcal{S} = (a_1, \ldots, a_{2k})$ for some $1 \leq k \leq n-1$ such that

$$\begin{cases} 1 \leq a_{2t-1} < a_{2t} \leq n & \text{for all } t = 1, \ldots, k \\ a_{2t-1} < a_{2t+1} & \text{for all } t = 1, \ldots, k-1 \\ a_{2t} < a_{2(t+1)} & \text{for all } t = 1, \ldots, k-1 \end{cases}$$

we can univocally associate them a partition $\pi \in \mathrm{NN}\,[n,k]$, since clearly $\mathcal{S} = \mathrm{Arch}\,(\pi)$.

We explicitly present now an algorithm which generates $\mathrm{NN}\,[n,k]$ and $\mathrm{SNN}\,[n,k]$ starting from $\mathrm{NN}\,[n,k-1]$ and $\mathrm{SNN}\,[n,k-1]$, respectively.
In order to show the extremely close similarities between the generation of these two sets, $\mathrm{NN}\,[n,k]$ and $\mathrm{SNN}\,[n,k]$, we consider a flag variable b such that

$$b = \begin{cases} 0 & \text{if we are generating NN}\,[n,k] \text{ from NN}\,[n,k-1] \\ 1 & \text{if we are generating SNN}\,[n,k] \text{ from SNN}\,[n,k-1] \end{cases}$$

therefore we can just present a unified design of such algorithm.

Algorithm 4.3 ($\mathrm{Gen}(n,k,b)$)**.**

```
If $k = 0$ then Output: $\pi$ such that $\mathrm{Arch}\,(\pi) = (\emptyset)$;
    else
      Begin
        Input: $\pi_1, \ldots, \pi_t$, output of $\mathrm{Gen}(n, k-1, b)$ in the order in which they are generated;
        For $i := 1$ to $t$ do
          Begin
            If $k = 1$ then
              Begin
                $lo_i := 0$;
                $lc_i := 1$;
              End
              else      /* i.e. If $k \geq 2$ */
              Begin
                $lo_i :=$ largest opener of $\pi_i$;
  /* i.e. $lo_i := o_{k-1}$, the $2k-3$ coordinate in $\mathrm{Arch}\,(\pi_i)$ */
                $lc_i :=$ largest closer of $\pi_i$;
  /* i.e. $lc_i := c_{k-1}$, the $2\,(k-1)$ coordinate in $\mathrm{Arch}\,(\pi_i)$ */
```

```
          End
        For $o := lo_i + 1$ to $n - 1$ do
        Begin
          for $c := \max\{o+1+b, lc_i+1\}$ to $n$ do Output: $\pi$ such that
Arch$(\pi)$ = (Arch$(\pi_i)$, $o, c$);
        End
      End
    End
```

Before proving the validity of this algorithm we describe one important property.
This algorithm endows $\mathrm{NN}\,[n,k]$, and therefore all its subsets, like for instance $\mathrm{SNN}\,[n,k]$, with a total order $\preceq$, denoted as the *lexicographical order*, given by the order of output generation. This order is the following:

$$\pi \preceq \tau \text{ if and only if } \mathrm{Arch}\,(\pi) \leq \mathrm{Arch}\,(\tau) \text{ lexicographically.}$$

Proposition 4.4. *For all integers* $0 \leq k \leq n-1$*,* $b = 0$*, respectively* $b = 1$*, the algorithm* $\mathrm{Gen}(n,k,b)$ *combinatorially generates* $\mathrm{NN}\,[n,k]$*, respectively* $\mathrm{SNN}\,[n,k]$*, in the lexicographical order.*
Furthermore, $\mathrm{Gen}(n,k,b)$ *has optimal computational complexity.*

Proof: Consider first the case $b = 0$; we have to show that all and only $\pi \in \mathrm{NN}\,[n,k]$ are generated, that each such partition is generated exactly once, and that this generation is in lexicographical order. If $k = 0$ then this is obvious since there is only one partition in $\pi \in \mathrm{NN}\,[n,0]$, namely π such that $\mathrm{Arch}\,(\pi) = (\emptyset)$.
If $k > 0$ then given $\pi \in \mathrm{NN}\,[n,k]$ consider $\mathrm{Arch}\,(\pi) := (o_1, c_1, \dots, o_k, c_k)$; because of Lemma 4.2, $(o_1, c_1, \dots, o_{k-1}, c_{k-1})$ equals $\mathrm{Arch}\,(\pi')$ for some $\pi' \in \mathrm{NN}\,[n,k-1]$, and since in our algorithm we consider all such π' and all possible ways to add the last arc (o_k, c_k) in π we are clearly generating all $\pi \in \mathrm{NN}\,[n,k]$, and nothing else, since we are adding to a nonnesting partition only one arc whose closer is larger than all others. Moreover, it is clear that we cannot generate twice the same partition $\pi \in \mathrm{NN}\,[n,k]$ because given $\pi' \in \mathrm{NN}\,[n,k-1]$ the last arc (o_k, c_k) that we are adding in order to obtain π is always "greater" (according to both openers and closers) of all arcs in π'.
Finally, this generation is in lexicographical order. In fact, consider two partitions π and τ such that τ is generated (immediately) after π, and

the corresponding arrays $\mathrm{Arch}\left(\pi\right) = \left(o_1\left(\pi\right), c_1\left(\pi\right), \ldots, o_k\left(\pi\right), c_k\left(\pi\right)\right)$ and $\mathrm{Arch}\left(\tau\right) = \left(o_1\left(\tau\right), c_1\left(\tau\right), \ldots, o_k\left(\tau\right), c_k\left(\tau\right)\right)$. By induction

$$\left(o_1\left(\pi\right), c_1\left(\pi\right), \ldots, o_{k-1}\left(\pi\right), c_{k-1}\left(\pi\right)\right) \le \left(o_1\left(\tau\right), c_1\left(\tau\right), \ldots, o_{k-1}\left(\tau\right), c_{k-1}\left(\tau\right)\right)$$

lexicographically, and if the equality holds then $\left(o_k\left(\pi\right), c_k\left(\pi\right)\right) \le \left(o_k\left(\tau\right), c_k\left(\tau\right)\right)$ lexicographically, because of the design of the two for loops.

If $b = 1$ all the above reasonings apply, and the only thing one has to check is that the generated partitions are actually sparse, i.e. the distance between every opener and its corresponding closer is at least 2, but this is true because of induction for the first $k-1$ arcs and the flag variable set to 1 in the internal for loop for the last arc.

Lastly, $\mathrm{Gen}(n, k, b)$ has optimal computational complexity because it is clearly a linear output–sensitive combinatorial generation algorithm, see [38].

■

It is known that $\#\,\mathrm{SP}\left(n+1\right) = \#P\left(n\right)$, see e.g. [24, 39, 52], and combining the main result of [11] with one of the known bijections linking noncrossing and nonnesting partitions, we get that $\#\,\mathrm{NN}\left(n,k\right) = \#\,\mathrm{SNN}\left(n+1,k\right)$, but the resulting bijection which exhibits such equality is extremely complicated and “ugly”.

Using Algorithm $\mathrm{Gen}(n, k, b)$ we show the equality

$$\#\,\mathrm{NN}\left[n,k\right] = \#\,\mathrm{SNN}\left[n+1,k\right]$$

for all integers $n \ge 1$ and $0 \le k \le n-1$, and we explicitly exhibit a bijection between the two sets which is extremely easy to present and compute.

In fact, the two algorithms $\mathrm{Gen}(n, k, 0)$ and $\mathrm{Gen}(n+1, k, 1)$ give a (possibly partial) injective map $\varPhi$ between $\mathrm{NN}\left[n,k\right]$ and $\mathrm{SNN}\left[n+1,k\right]$, matching the i–th generated partition of $\mathrm{Gen}(n, k, 0)$ with the i–th generated partition of $\mathrm{Gen}(n+1, k, 1)$.

From an analysis of the two algorithms we see that actually $\varPhi$ has an extremely simple and nice form, and it is indeed a bijection. In fact, if $k = 0$ then obviously $\#\,\mathrm{NN}\left[n,0\right] = \#\,\mathrm{SNN}\left[n+1,0\right] = 1$, so everything is trivial, whereas if $k \ge 1$ for any $\pi \in \mathrm{NN}\left[n,k\right]$ consider $\mathrm{Arch}\left(\pi\right) = \left(o_1, c_1, \ldots, o_k, c_k\right)$; then $\varPhi\left(\pi\right)$ is such that
$\mathrm{Arch}\left(\varPhi\left(\pi\right)\right) = \left(o_1, c_1+1, \ldots, o_k, c_k+1\right)$.

We conclude this section with some instances of partitions generated by running

$\mathrm{Gen}(n,k,b)$. For the sake of conciseness and lucidity of exposition, we consider

Algorithm 4.5 ($\mathrm{GenTot}(n,b)$)**.**
For $k := 0$ *to* $n-1$ *do* $\mathrm{Gen}(n,k,b)$.

and present the output of $\mathrm{GenTot}(4,0)$ and $\mathrm{GenTot}(5,1)$ in the correct order of generation, representing partitions both according their blocks and using Arch notation.

Example 4.6.

Output of $\mathrm{GenTot}(4,0)$

Block notation	Arch
{{*1*},{*2*},{*3*},{*4*}}	$(\emptyset)$
{{*1,2*},{*3*},{*4*}}	$(1,2)$
{{*1,3*},{*2*},{*4*}}	$(1,3)$
{{*1,4*},{*2*},{*3*}}	$(1,4)$
{{*1*},{*2,3*},{*4*}}	$(2,3)$
{{*1*},{*2,4*},{*3*}}	$(2,4)$
{{*1*},{*2*},{*3,4*}}	$(3,4)$
{{*1,2,3*},{*4*}}	$(1,2,2,3)$
{{*1,2,4*},{*3*}}	$(1,2,2,4)$
{{*1,2*},{*3,4*}}	$(1,2,3,4)$
{{*1,3*},{*2,4*}}	$(1,3,2,4)$
{{*1,3,4*},{*2*}}	$(1,3,3,4)$
{{*1*},{*2,3,4*}}	$(2,3,3,4)$
{{*1,2,3,4*}}	$(1,2,2,3,3,4)$

Output of $\mathrm{GenTot}(5,1)$

Block notation	Arch
{{*1*},{*2*},{*3*},{*4*},{*5*}}	$(\emptyset)$
{{*1,3*},{*2*},{*4*},{*5*}}	$(1,3)$
{{*1,4*},{*2*},{*3*},{*5*}}	$(1,4)$
{{*1,5*},{*2*},{*3*},{*4*}}	$(1,5)$
{{*1*},{*2,4*},{*3*},{*5*}}	$(2,4)$
{{*1*},{*2,5*},{*3*},{*4*}}	$(2,5)$
{{*1*},{*2*},{*3,5*},{*4*}}	$(3,5)$
{{*1,3*},{*2,4*},{*5*}}	$(1,3,2,4)$
{{*1,3*},{*2,5*},{*4*}}	$(1,3,2,5)$
{{*1,3,5*},{*2*},{*4*}}	$(1,3,3,5)$
{{*1,4*},{*2,5*},{*3*}}	$(1,4,2,5)$
{{*1,4*},{*2*},{*3,5*}}	$(1,4,3,5)$
{{*1*},{*2,4*},{*3,5*}}	$(2,4,3,5)$
{{*1,3,5*},{*2,4*}}	$(1,3,2,4,3,5)$

5. Open problems

The notion of nonnesting set partition exists in all classical reflection groups, see e.g. [2, 12, 13, 40], so it is quite natural to ask for a generalization of our results to Weyl groups of type B and D.

Open problem 5.1. *Obtain Gray codes and efficient combinatorial generation algorithms for the sets of nonnesting partitions in Weyl groups of type B and D.*

To the best of our knowledge, so far the only results in this direction are the design of Gray codes for set partitions in classical Weyl groups, see [14, 26], but nothing at all is known when the nonnesting condition is added.

References

[1] D. Armstrong, 'Generalized Noncrossing Partitions and Combinatorics of Coxeter Groups', *Mem. Amer. Math. Soc.* **202** (2009), no. 949.

[2] C. A. Athanasiadis, 'On noncrossing and nonnesting partitions for classical reflection groups', *Electron. J. Comb.* **5** (1998), Research Paper 42, 16pp (electronic).

[3] S. Bacchelli, E. Barcucci, E. Grazzini and E. Pergola, 'Exhaustive generation of combinatorial objects by ECO', *Acta Inform.* **40** (2004), 585–602.

[4] E. Barcucci , A. Del Lungo, E. Pergola and R. Pinzani, 'ECO: a methodology for the enumeration of combinatorial objects', *J. Differ. Equations Appl.* **5** (1999), 435-490.

[5] J.–L. Baril, 'Gray code for permutations with a fixed number of cycles', *Discrete Math.* **307** (2007), 1559-1571.

[6] J.–L. Baril, 'More restrictive Gray codes for some classes of pattern avoiding permutations', *Inform. Process. Lett.* **109** (2009), 799-804.

[7] J.–L. Baril and V. Vajnovszki, 'Gray code for derangements', *Discrete Appl. Math.* **140** (2004), 207-221.

[8] B. Bauslaugh and F. Ruskey, 'Generating alternating permutations lexicographically', *BIT* **30** (1990), 17-26.

[9] A. Bernini, I. Fanti and E. Grazzini, 'An exhaustive generation algorithm for Catalan objects and others', *Pure Math. Appl. (PU.M.A.)* **17** (2006), 39-53.

[10] S. Burrill, S. Elizalde, M. Mishna and L. Yen, 'A generating tree approach to k–nonnesting partitions and permutations', *preprint*, available at `http://front.math.ucdavis.edu/1108.5615`

[11] W. Y. Chen, E. Y. Deng and R. R. X. Du, 'Reduction of m–regular noncrossing partitions', *European J. Combin.* **26** (2005), 237-243.

[12] A. Conflitti and R. Mamede, 'Bijecções entre partições que não cruzam e partições que não encaixam', *Bol. Soc. Port. Mat.* Número Especial, Actas do Encontro Nacional da SPM, Leiria 2010, 20–27.

[13] A. Conflitti and R. Mamede, 'On noncrossing and nonnesting partitions of type D', *Ann. Comb.* **15** (2011), 637–654.

[14] J. H. Conway, N. J. A. Sloane and A. R. Wilks, 'Gray codes for reflection groups', *Graphs Combin.* **5** (1989), 315-325.

[15] W. M. B. Dukes, M. F. Flanagan, T. Mansour and V. Vajnovszki, 'Combinatorial Gray codes for classes of pattern avoiding permutations', *Theoret. Comput. Sci.* **396** (2008), 35-49.

[16] G. Ehrlich, 'Loopless algorithms for generating permutations, combinations, and other combinatorial configurations', *J. Assoc. Comput. Mach.* **20** (1973), 500-513.

[17] A. Fink and B. I. Giraldo, 'Bijections between noncrossing and nonnesting partitions for classical reflection groups, *Port. Math.* **67** (2010), 369-401.

[18] D. Gusfield. 'Partition-distance: a problem and class of perfect graphs arising in clustering', *Inform. Process. Lett.* **82** (2002), 159–164.

[19] C. Huemer, F. Hurtado, M. Noy and E. Omaña–Pulido, 'Gray codes for non–crossing partitions and dissections of a convex polygon', *Discrete Appl. Math.* **157** (2009), 1509–1520.

[20] A. Itai, 'Generating permutations and combinations in lexicographical order' *J. Braz. Comp. Soc.* **7** (2001), 65–68.

[21] R. Kaye, 'A Gray code for set partitions', *Information Processing Lett.* **5** (1976), 171–173.

[22] R. Kemp, 'Generating words lexicographically: an average-case analysis', *Acta Inform.* **35** (1998), 17-89.

[23] J. S. Kim, 'New interpretations for noncrossing partitions of classical types', *J. Combin. Theory Ser. A* **118** (2011), 1168-1189.

[24] M. Klazar, 'Counting set systems by weight', *Electron. J. Comb.* **12** (2005), Research Paper 11, 8pp (electronic).

[25] D. E. Knuth, *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms, Part 1*, Addison–Wesley Professional, 2011

[26] J. Korsh and S. Lipschutz, 'Gray code and loopless algorithm for the reflection group D_n', *Pure Math. Appl. (PU.M.A.)* **17** (2006), 135-146.

[27] D. L. Kreher and D. R. Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*, CRC Press Series on Discrete Mathematics and Its Applications, CRC Press, Boca Raton, Florida, 1998.

[28] R. Mamede, 'A bijection between noncrossing and nonnesting partitions of types A, B and C', *Contrib. Discrete Math.* 6 (2011), 70–90.

[29] T. Mansour, *Combinatorics of set partitions*, Discrete Mathematics and its Applications (Boca Raton), CRC Press, Boca Raton, FL, 2013.

[30] T. Mansour and G. Nassar, 'Gray codes, loopless algorithm and partitions', *J. Math. Model. Algorithms* **7** (2008), 291-310.

[31] J. McCammond, 'Noncrossing partitions in surprising locations', *Amer. Math. Monthly* **113** (2006), 598-610.

[32] M. E. Nebel, 'On the lexicographical generation of compressed codes', *Inform. Process. Lett.* **104** (2007), 95-100.

[33] J. Němeček and M. Klazar, 'A bijection between nonnegative words and sparse *abba*–free partitions', *Discrete Math.* **265** (2003), 411-416.

[34] A. Nijenhuis and H. W. Wilf, *Combinatorial algorithms*, 2nd edition, Academic Press, New York–London, 1978.

[35] A. Nica and R. Speicher, *Lectures on the combinatorics of free probability*, London Mathematical Society Lecture Note Series 335, Cambridge University Press, Cambridge 2006.

[36] R. J. Ord-Smith, 'Algorithms: Algorithm 323: Generation of permutations in lexicographic order', *Comm. ACM* **11** (1968), 117.

[37] M. Poneti and V. Vajnovszki, 'Generating restricted classes of involutions, Bell and Stirling permutations', *European J. Combin.* **31** (2010), 553-564.

[38] F. P. Preparata and M. I. Shamos, *Computational geometry. An introduction*, Springer–Verlag, New York, 1985.

[39] H. Prodinger, 'On the number of Fibonacci partitions of a set', *Fibonacci Quart.* **19** (1981), 463-465.

[40] V. Reiner, 'Non–crossing partitions for classical reflection groups', *Discrete Math.* **177** (1997), 195–222.

[41] E. M. Reingold, J. Nievergelt and N. Deo, *Combinatorial algorithms: theory and practice*, Prentice–Hall, Inc., Englewood Cliffs, N.J., 1977.

[42] M. Rubey and C. Stump, 'Crossing and nestings in set partitions of classical types', *Electron. J. Comb.* **17** (2010), Research Paper 120, 19pp (electronic).

[43] F. Ruskey, *Combinatorial Generation*, available on `http://www.1stworks.com/ref/RuskeyCombGen.pdf`
[44] F. Ruskey and C. Savage, 'Gray codes for set partitions and restricted growth tails', *Australas. J. Combin.* **10** (1994), 85-96.
[45] C. Savage, 'A survey of combinatorial Gray codes', *SIAM Rev.* **39** (1997), 605-629.
[46] R. Simion, 'Noncrossing partitions', in Formal power series and algebraic combinatorics (Vienna, 1997), *Discrete Math.* **217** (2000), 367–409.
[47] R. Speicher, 'Free probability theory and non-crossing partitions', *Sém. Lothar. Combin.* **39** (1997), Art. B39c, 38 pp. (electronic).
[48] R. Speicher, 'Combinatorial theory of the free product with amalgamation and operator–valued free probability theory', *Mem. Amer. Math. Soc.* **132** (1998).
[49] D. Stanton and D. White, *Constructive Combinatorics*, Springer–Verlag, New York 1986.
[50] V. Vajnovszki, 'Generating involutions, derangements, and relatives by ECO', *Discrete Math. Theor. Comput. Sci.* **12** (2010), 109-122.
[51] H. S. Wilf, *Combinatorial algorithms: an update*, CBMS-NSF Regional Conference Series in Applied Mathematics 55, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1989.
[52] W. Yang, 'Bell numbers and k–trees', *Discrete Math.* **156** (1996), 247-252.
[53] S. Zaks, 'Lexicographic generation of ordered trees', *Theoret. Comput. Sci.* **10** (1980), 63-82.
[54] S. Zaks and D. Richards, 'Generating trees and other combinatorial objects lexicographically', *SIAM J. Comput.* **8** (1979), 73-81.

Alessandro Conflitti
CMUC, Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal
E-mail address: `conflitt@mat.uc.pt`

Ricardo Mamede
CMUC, Department of Mathematics, University of Coimbra, 3001-501 Coimbra, Portugal
E-mail address: `mamede@mat.uc.pt`