Sort-Invariant Non-Messing-Up

Bridget Eileen Tenner

Department of Mathematical Sciences
DePaul University
Chicago, IL, USA

bridget@math.depaul.edu

Submitted: Nov 17, 2010; Accepted: Nov 8, 2011; Published: Nov 21, 2011
Mathematics Subject Classification: 06A07, 68P10, 06A05

Abstract

A poset has the non-messing-up property if it has two covering sets of disjoint saturated chains so that for any labeling of the poset, sorting the labels along one set of chains and then sorting the labels along the other set yields a linear extension of the poset. The linear extension yielded by thus twice sorting a labeled non-messing-up poset may be independent of which sort was performed first. Here we characterize such sort-invariant labelings for convex subposets of a cylinder. They are completely determined by avoidance of a particular subpattern: a diamond of four elements whose smallest two labels appear at opposite points.

1 Introduction

The non-messing-up property of matrices is a well-known result about sorting the entries in a matrix. It appears in work by Boerner [1], Gale and Karp [2], Knuth [3], and Winkler [6], and a detailed discussion of its provenance occurs in [5]. Given a matrix M, let $\mathcal{C}(M)$ be the matrix obtained by ordering the entries within each column of M, and likewise let $\mathcal{R}(M)$ be the matrix obtained by ordering the entries within each row of M.

Example 1.1. Let

$$M = \begin{bmatrix} 4 & 9 & 7 & 8 \\ 12 & 5 & 1 & 10 \\ 2 & 6 & 11 & 3 \end{bmatrix}.$$

Then the matrices $\mathcal{C}(M)$ and $\mathcal{R}(M)$ are

$$\mathcal{C}(M) = \begin{bmatrix} 2 & 5 & 1 & 3 \\ 4 & 6 & 7 & 8 \\ 12 & 9 & 11 & 10 \end{bmatrix} \quad \text{and} \quad \mathcal{R}(M) = \begin{bmatrix} 4 & 7 & 8 & 9 \\ 1 & 5 & 10 & 12 \\ 2 & 3 & 6 & 11 \end{bmatrix}.$$

Moreover,

$$\mathcal{RC}(M) = \begin{bmatrix} 1 & 2 & 3 & 5 \\ 4 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \quad \text{and} \quad \mathcal{CR}(M) = \begin{bmatrix} 1 & 3 & 6 & 9 \\ 2 & 5 & 8 & 11 \\ 4 & 7 & 10 & 12 \end{bmatrix}. \tag{1}$$

Theorem 1.2 (Non-messing-up matrices)**.** *Consider a matrix M. The entries in each column in the matrix $\mathcal{RC}(M)$ are in non-decreasing order; that is, $\mathcal{RC}(M) = \mathcal{CRC}(M)$.*

Less formally, Theorem 1.2 says that after sorting the data within each column of a matrix, sorting the data within each row of the subsequent matrix does not "mess up" the fact that each column's data are sorted. Note that by symmetry, the entries in the rows of the matrix $\mathcal{CR}(M)$ must also be sorted, so $\mathcal{CR}(M) = \mathcal{RCR}(M)$ as well.

The expressions of (1) provide an example of the non-messing-up property of matrices. Note from these that it is not necessarily the case that the matrices $\mathcal{RC}(M)$ and $\mathcal{CR}(M)$ coincide; that is, the sorting operations do not necessarily commute. It is this issue which we address here. Moreover, we examine it for more general non-messing-up posets, which we previously treated in [5].

In Section 2 we review the results of [5] and define the central object of this paper: a transverse non-messing-up poset. Our main result, stated precisely in Theorem 3.6, is that the sorting operations commute on a labeling of a transverse non-messing-up poset if and only if the labeling avoids a particular forbidden subpattern (subject to some degeneracies) consisting of a diamond of four elements whose smallest labels appear at opposite points. Unfortunately this result does not hold for non-transverse non-messing-up posets, as shown in Section 4. The paper concludes with a brief discussion of sorted matrices and when they have a preferred sorting procedure; that is, when it is more likely that a sorted matrix arises from first column-sorting and then row-sorting, or vice versa.

2 Non-messing-up posets

The goal of this paper is to describe exactly when the sorting operations $\mathcal{R}$ and $\mathcal{C}$ commute on a labeling, and we do this in the context of non-messing-up posets. Previously, in [5], we defined and characterized a generalization of the non-messing-up phenomenon to the setting of posets. That work is summarized here, and more details can be found in [5].

Generalizing to the realm of posets is natural because an r-by-c matrix resembles the poset $\boldsymbol{r} \times \boldsymbol{c}$, a product of two chains. The rows and columns of the matrix correspond to two sets of disjoint saturated chains covering the elements of $\boldsymbol{r} \times \boldsymbol{c}$. Sorting a chain means putting that chain's labels in order so that the minimum element in the chain has the minimum label. Thus the non-messing-up property indicates that sorting the labels along both the rows and the columns gives a linear extension of $\boldsymbol{r} \times \boldsymbol{c}$.

Definition 2.1. Let P be a finite poset. A *gridwork* for P is a pair (R, C) where R and C are each sets of disjoint saturated chains covering the elements of P, and where each covering relation in P is contained in an element of $R \cup C$. We call the elements of R the *rows* of P, and the elements of C the *columns* of P.

Definition 2.2. Let P be a finite poset with gridwork (R, C). The gridwork is *transverse* if each row and each column intersect at most once.

Definition 2.3. Let P be a finite poset with a transverse gridwork (R, C). Given a labeling of the elements of P, let $\mathcal{R}$ be the operation which sorts the labels within each row of the poset, so that the minimum element in a row has the minimum label of that row. Likewise, let $\mathcal{C}$ be the operation which sorts the labels within each column of the poset.

We are now able to carry the notion of non-messing-up to the setting of posets.

Definition 2.4. A poset P has the *transverse non-messing-up* property, or is *transverse non-messing-up*, if there exists a transverse gridwork such that, for any labeling λ of the elements of P, both $\mathcal{RC}(\lambda)$ and $\mathcal{CR}(\lambda)$ yield linear extensions of P; that is, $\mathcal{RC}(\lambda) = \mathcal{CRC}(\lambda)$ and $\mathcal{CR}(\lambda) = \mathcal{RCR}(\lambda)$.

Before stating the characterization of transverse non-messing-up posets found in [5], we must state one more definition.

Definition 2.5. Fix positive integers k and n so that $k < n$. The *cylinder poset* $Cyl_{k,n}$ is $\mathbb{Z}^2/(-k, n-k)\mathbb{Z}$. The partial order on $Cyl_{k,n}$ is induced by the componentwise partial order on $\mathbb{Z}^2$.

Theorem 2.6 ([5]). *A poset is transverse non-messing-up if and only if each of its connected components P is a convex subposet of a cylinder poset.*

The gridwork for a non-messing-up poset is specified in [5], and the sets R and C are analogous to the rows and columns of the motivating matrix example. The reader is encouraged to find the details in [5].

Note that one can also consider non-transverse non-messing-up posets, and those are also classified in [5]. Such posets differ from transverse non-messing-up posets by allowing a row and a column to intersect more than once. Each non-transverse non-messing-up poset is obtained from a transverse one by replacing one or more elements by chains, subject to length restrictions. Not only does a non-transverse non-messing-up poset look quite different from the motivating matrix situation, but there is some redundancy in the sorting operations since, for example, chains of elements in the columns will already have been sorted by $\mathcal{R}$.

3 Transverse sort-invariance

Throughout this section, let P be a transverse non-messing-up poset with transverse gridwork (R, C). For the sake of clarity, suppose that in any labeling of P, the elements are labeled by $\{1, 2, \ldots, |P|\}$.

Definition 3.1. A labeling λ of the poset P is *sort-invariant* if $\mathcal{RC}(\lambda) = \mathcal{CR}(\lambda)$.

In other words, a labeling λ is sort-invariant if the operations $\mathcal{R}$ and $\mathcal{C}$ commute on λ. Our goal is to characterize sort-invariant labelings of transverse non-messing-up posets.

Definition 3.2. Let x and y be elements of non-messing-up poset, that are in neither the same row nor the same column. Suppose that x is in row $\boldsymbol{r_x}$ and column $\boldsymbol{c_x}$, while y is in row $\boldsymbol{r_y}$ and column $\boldsymbol{c_y}$ (thus $\boldsymbol{r_x} \neq \boldsymbol{r_y}$ and $\boldsymbol{c_x} \neq \boldsymbol{c_y}$). The *corner-set of* x *and* y is the subset

$$\begin{aligned} \Diamond(x,y) &= (\boldsymbol{r_x} \cup \boldsymbol{r_y}) \cap (\boldsymbol{c_x} \cup \boldsymbol{c_y}) \\ &= (\boldsymbol{r_x} \cap \boldsymbol{c_x}) \cup (\boldsymbol{r_y} \cap \boldsymbol{c_y}) \cup (\boldsymbol{r_x} \cap \boldsymbol{c_y}) \cup (\boldsymbol{c_x} \cap \boldsymbol{r_y}). \end{aligned} \tag{2}$$

If x and y are in either the same row or the same column, then set $\Diamond(x,y) = \emptyset$.

Because P is transverse, if x and y are in neither the same row nor the same column in P, then we have

$$\Diamond(x,y) = \{x,y\} \cup (\boldsymbol{r_x} \cap \boldsymbol{c_y}) \cup (\boldsymbol{c_x} \cap \boldsymbol{r_y}), \tag{3}$$

and this $\Diamond(x,y)$ contains at most four elements.

The nomenclature here refers to the fact that the set $\Diamond(x,y)$ looks like the four corners of a diamond containing x and y in P, when the edges of the diamond must be either rows or columns in the poset. Note that because non-messing-up posets (and their rows and columns) arise from convex subposets of cylinder posets, if x and y are comparable in P, then both $\boldsymbol{r_x} \cap \boldsymbol{c_y}$ and $\boldsymbol{c_x} \cap \boldsymbol{r_y}$ are nonempty.

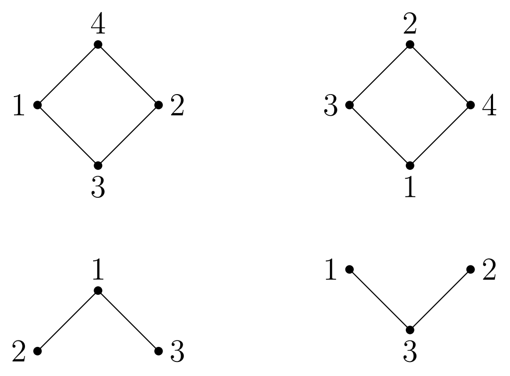
In the case of the product of two chains, we will show that a labeling is sort-invariant if and only if corner-sets with extreme values on opposite sides are avoided. In general, a corner-set may have fewer than four elements, and we must avoid more particular patterns with these smaller sets, as stated in the second and third points of Definition 3.3.

Definition 3.3. A nonempty corner-set $\Diamond(x,y)$ is *bad for* λ (or simply *bad* if λ is clear from the context) if $|\Diamond(x,y)| > 2$ and one of the following situations is satisfied:

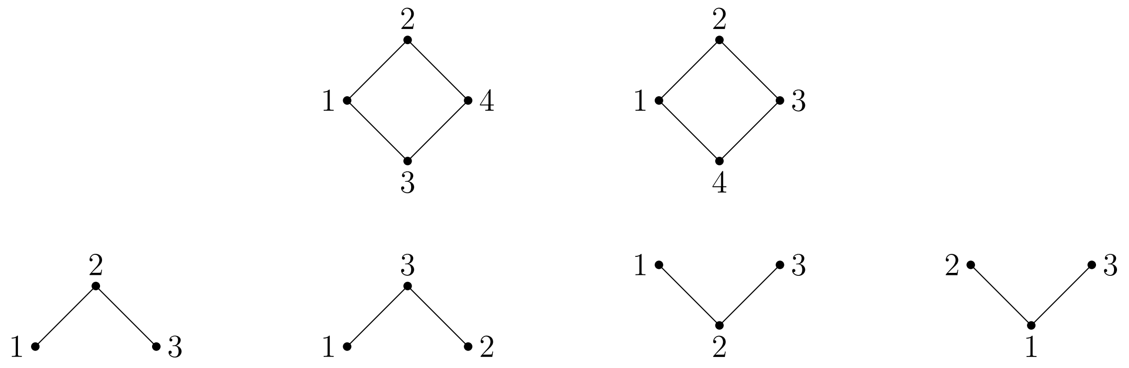
- $\Diamond(x,y) = \{x,y,w,z\}$ and
$$\lambda(x), \lambda(y) > \lambda(w), \lambda(z) \quad \text{or} \quad \lambda(x), \lambda(y) < \lambda(w), \lambda(z),$$
- $\Diamond(x,y) = \{x,y,z\}$ with $x, y < z$ and $\lambda(x), \lambda(y) > \lambda(z)$, or
- $\Diamond(x,y) = \{x,y,z\}$ with $x, y > z$ and $\lambda(x), \lambda(y) < \lambda(z)$.

If $\Diamond(x,y)$ is not bad for λ, then it is *good for* λ (or simply *good* if λ is clear from the context).

Example 3.4. The following four posets and labelings have only bad corner-sets.



Example 3.5. The following six posets and labelings have only good corner-sets.



We are now able to state the main result of this paper, which characterizes exactly when a labeling of a transverse non-messing-up poset is sort-invariant in terms of its corner-sets.

Theorem 3.6. *Let P be a transverse non-messing-up poset and λ a labeling of P. The labeling λ is sort-invariant if and only if each nonempty corner-set in P is good.*

In other words, sort-invariance of a transverse non-messing-up poset (that is, a convex subposet of a cylinder poset) is characterized by a forbidden subpattern: a bad corner-set.

Theorem 3.6 is proved in three steps, two of which we highlight as propositions. The first step is to analyze sort-invariance for the product of two chains (the poset version of the matrix), the second step is to consider convex subposets of the product of two chains, and the final step is to look at the cylinder.

In the next proposition, we describe sort-invariance for posets which are the product of two chains.

Proposition 3.7. *Let the poset P be the product of two chains, and let λ be a labeling of P. The labeling λ is sort-invariant if and only if each nonempty corner-set in P is good.*

Proof. First, note that P is a transverse non-messing-up poset. Let R and C be the rows and columns of P, which are analogous to the rows and columns in a matrix. The row which intersects each column in the minimum position of the column will be called the "first" row, the row intersecting in the position above that will be the "second" row, and so on. We similarly name the columns.

We first note that sort-invariance is equivalent to saying that for all $k \in \{1, \ldots, |P|\}$, the shape formed by $\{1, \ldots, k\}$ in $\mathcal{RC}(\lambda)$ is the same as the shape formed by $\{1, \ldots, k\}$ in $\mathcal{CR}(\lambda)$. Thus we can replace the labels in λ by just $\{1_k, 2_k\}$, where 1_k replaces each element of $\{1, \ldots, k\}$ and 2_k replaces each element of $\{k+1, \ldots, |P|\}$, and look at the resulting placements of the 1_ks after $\mathcal{RC}$ and $\mathcal{CR}$, with the understanding that $2_k > 1_k$.

For the remainder of the proof we will drop the subscript k and just look at a labeling by 1s and 2s, and decide when $\mathcal{R}$ and $\mathcal{C}$ commute on this filling.

The operation $\mathcal{C}$ pushes all the 1s to the bottom of each column. One interpretation of $\mathcal{RC}$ is that it sorts the set of columns so that the column in which the most 1s appears is now the first column, the column in which the next most 1s appears is now the second column, and so on. The shape of 1s resulting from $\mathcal{CR}$ can be described analogously.

For these shapes to be equal, we need the ith most 1s appearing in any row to equal the number of columns containing at least i 1s. This is equivalent to the following criterion: let $S_{\boldsymbol{c}} \subseteq R$ be the set of rows in which column $\boldsymbol{c}$ has a 1; index the columns $\boldsymbol{c_1}, \boldsymbol{c_2}, \ldots$ so that $\boldsymbol{c_{i+1}}$ has at least as many 1s as $\boldsymbol{c_i}$ does; then we must have

$$S_{\boldsymbol{c_1}} \subseteq S_{\boldsymbol{c_2}} \subseteq \ldots . \tag{4}$$

(Equivalently, the roles of "row" and "column" can be swapped in the previous discussion.)

For example, the following is a labeling satisfying (4), where the poset has been rotated to highlight the rows and the columns.

$$\begin{array}{cccccc}
2 & 1 & 2 & 2 & 1 & 1 \\
2 & 1 & 1 & 2 & 1 & 1 \\
2 & 1 & 2 & 2 & 2 & 2 \\
2 & 1 & 2 & 2 & 2 & 1 \\
2 & 2 & 2 & 2 & 2 & 2 \\
2 & 1 & 1 & 2 & 1 & 1 \\
\uparrow & \uparrow & \uparrow & \uparrow & \uparrow & \uparrow \\
\boldsymbol{c_1} & \boldsymbol{c_6} & \boldsymbol{c_3} & \boldsymbol{c_2} & \boldsymbol{c_4} & \boldsymbol{c_5}
\end{array}$$

The $\boldsymbol{c_i}$ designations are given below each column. Note that $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ can be swapped.

We claim that the inclusions of (4) are equivalent to having no bad corner-set. Certainly if the inclusions of (4) hold then there is no such bad corner-set. Conversely, if there is no bad corner-set then consider any pair of columns $\boldsymbol{c}$ and $\boldsymbol{c'}$, where $\boldsymbol{c}$ has at least as many 1s as $\boldsymbol{c'}$. If there is a row in which $\boldsymbol{c}$ has a 1 but $\boldsymbol{c'}$ does not, then every row in which $\boldsymbol{c'}$ has a 1 must be a row in which $\boldsymbol{c}$ does as well. This implies (4), and completes the proof. $\square$

Proposition 3.8. *Let the poset P be a convex subposet of the product of two chains, and let λ be a labeling of P. The labeling λ is sort-invariant if and only if each nonempty corner-set in P is good.*

Proof. Suppose P is a convex subposet of P', where P' is a product of two chains. Extend the labeling λ of P to a labeling λ' of P' by giving the label 1 to every element of $P' \setminus P$ less than an element of P, and the label $|P|$ to all other elements of $P' \setminus P$. Clearly λ is sort-invariant if and only if λ' is sort-invariant. Similarly, λ has a bad corner-set if and only if λ' has a bad corner-set. Proposition 3.7 links these two concepts for λ', and thus we conclude that λ is sort-invariant if and only if all of its non-empty corner-sets are good. $\square$

Proof of Theorem 3.6. A labeling of a poset is sort-invariant if and only if the labeling of each connected component of the poset is sort-invariant. Thus we need only prove the result for a connected poset. The remainder of the proof resembles that of [5, Theorem 6].

Let P be a finite convex subposet of $Cyl_{k,n}$. Cut $Cyl_{k,n}$ to get a preimage of P in the plane. Glue together copies of this poset via the identifications on the cylinder. After perhaps removing elements at the sides of the planar poset, this is a convex subposet of a product of two chains. For the labeling λ of P, label every preimage of x in the plane by $\lambda(x)$. Glue enough copies of the poset so that after the two sorting procedures, sufficiently many of the centermost copies in the plane have the labels they would have had on the cylinder. This is possible because only finitely many elements cross over a line of identification. Call this glued-together poset P', and its labeling λ'.

The labeling λ has a bad corner-set if and only if λ' does. Thus, by Proposition 3.8, λ has a bad corner-set if and only if λ' is not sort-invariant. If λ' is sort-invariant then certainly λ is as well. Therefore, if λ has no bad corner-sets, then λ is sort-invariant.

On the other hand, suppose that λ has a bad corner-set. Thus λ' has a bad corner-set, and is not sort-invariant. We must be sure that the sort-invariance fails somewhere in one of the centermost copies of the cut poset. This will ensure that λ is also not sort-invariant, which will conclude the proof. Since Proposition 3.7 characterizes sort-invariance in the plane by forbidding bad corner-sets, interacting with the labels appearing in a bad corner-set during $\mathcal{R}$ or $\mathcal{C}$ must lead to the failure of the sort-invariance. By construction of P', such failure must appear among the centermost copies of the cut poset. Since these centermost copies have the same labels that they would have had on the cylinder, this means that λ is not sort-invariant. $\square$

4 Non-split sort-invariance

One might hope that sort-invariance for non-transverse non-messing-up posets can be characterized by an analogous statement about bad corner-sets. Unfortunately the obvious generalization of this concept is not the answer. More precisely, recall the definition of a corner-set, and in particular equation (2). In a non-transverse poset, the subsequent equation (3) does not necessarily hold, and the corner-set might well have more than four elements. We can extend the definition of "bad" in the obvious way, to say that a corner-set is bad if there exist elements

$$\begin{aligned} x' &\in \boldsymbol{r_x} \cap \boldsymbol{c_x}, \\ y' &\in \boldsymbol{r_y} \cap \boldsymbol{c_y}, \\ u &\in \boldsymbol{r_x} \cap \boldsymbol{c_y}, \text{ and} \\ v &\in \boldsymbol{r_y} \cap \boldsymbol{c_x}, \end{aligned}$$

so that the labels of these elements either satisfy

$$\lambda(x'), \lambda(y') > \lambda(u), \lambda(v)$$

or

$$\lambda(x'), \lambda(y') < \lambda(u), \lambda(v).$$

However, avoidance of such a bad corner-set is not enough to guarantee sort-invariance in a general (non-transverse) non-messing-up poset, as demonstrated in the following example.

Example 4.1. Figure 1 gives a non-transverse non-messing-up poset P (as described in [5]) and a labeling λ of P.

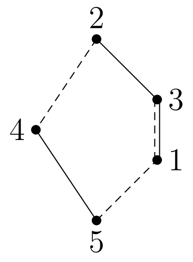


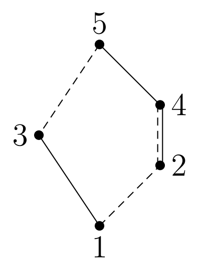
Figure 1: A non-messing-up poset P whose rows are drawn with dashed lines and columns with solid. A labeling λ of P is also given.

Notice that, in the sense described at the beginning of this section, the labeling λ is good. However, it is not sort-invariant, as demonstrated in Figure 2.

It is disappointing that the equivalence between sort-invariance and avoidance of a bad corner-set does not carry over to this more general setting, and it remains open to characterize sort-invariance of these non-transverse non-messing-up posets.

5 Preferred sorting procedures for matrices

We conclude with a brief discussion of a different but related question regarding sorted matrices; namely, whether a given sorted matrix A is more likely to have arisen from $\mathcal{RC}$ or from $\mathcal{CR}$. Certainly there is a matrix M which can yield A under both of these



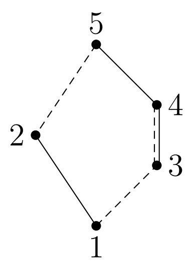


Figure 2: The left-hand poset is $\mathcal{RC}(\lambda)$, and the right-hand poset is $\mathcal{CR}(\lambda)$. They are unequal, and thus λ is not sort-invariant.

operations, namely $M = A$, but there may be a significant preference for $\mathcal{RC}$, say, in general.

Say that the matrix is *sorted* if the data in each row are non-decreasing from left to right, and if the data in each column are non-decreasing from top to bottom. Note that the transpose of a sorted matrix is also necessary sorted. Given a sorted matrix A, there are many possible original matrices M with $\mathcal{RC}(M) = A$ or with $\mathcal{CR}(M) = A$. Furthermore, the set of such M depends on whether the rows were sorted first and then the columns, or vice versa. For a given sorted matrix, it may be that one of these options is preferred to the other.

Without loss of generality, suppose that the data in the matrix is the set $\{1, \ldots, rc\}$, where the matrix has r rows and c columns. Using [4, Exercise 7.20b], we can compute the number of matrices M for which $\mathcal{RC}(M) = A$, and likewise for which $\mathcal{CR}(M) = A$. First we define the function h_A on the set $\{1, \ldots, rc\}$ of entries in A: if i is in the first row of A then $h_A(i) = 1$; otherwise,

$$h_A(i) = \#\{j < i : j \text{ is in the row immediately above } i \text{ in } A \text{ and not to the left of } i\}.$$

Set

$$h_A(A) = \prod h_A(i).$$

It follows from [4, Exercise 7.20b] that the number of matrices M for which $\mathcal{RC}(M) = A$ is

$$h_A(A) \cdot (r!)^c \cdot c!.$$

Likewise, the number of M for which $\mathcal{CR}(M) = A$ is

$$h_{A^T}(A) \cdot (c!)^r \cdot r!,$$

where A^T is the (sorted) transpose of the sorted matrix A.

Therefore we can make the following conclusion about preferred sorting procedures.

Proposition 5.1. *Suppose that A is a sorted r-by-c matrix. Let M be an r-by-c matrix chosen at random, and flip a fair coin to determine whether to sort M by performing $\mathcal{RC}$ or $\mathcal{CR}$. Then the probability that $A = \mathcal{RC}(M)$ is*

$$\frac{h_A(A) \cdot (r!)^{c-1}}{h_A(A) \cdot (r!)^{c-1} + h_{A^T}(A) \cdot (c!)^{r-1}}.$$

Acknowledgements

Sincere gratitude is due to Peter Winkler for very helpful and entertaining discussions.

References

[1] H. Boerner, *Darstellungen von Gruppen*, Springer-Verlag, Berlin, 1955, second edition 1967.

[2] D. Gale and R. M. Karp, A phenomenon in the theory of sorting, *J. Comput. System Sci.* **6** (1972), 104–115.

[3] D. E. Knuth, *The Art of Computer Programming, vol. 3, Sorting and Searching*, Addison-Wesley, Reading, MA, 1973, second edition 1998.

[4] R. P. Stanley, *Enumerative Combinatorics, vol. 1*, Cambridge Studies in Advanced Mathematics, no. 49, Cambridge University Press, Cambridge, 1997.

[5] B. E. Tenner, A non-messing-up phenomenon for posets, *Ann. Comb.* **11** (2007), 101–114.

[6] P. Winkler, *Mathematical Mind-Benders*, A. K. Peters, Ltd., Wellesley, MA, 2007.