December 9, 2003

TRANSFORMATIONS ON WORDS

By

Dominique FOATA
Département de mathématique
Université Louis Pasteur
7, rue René-Descartes,
F-67084 Strasbourg, France
email: foata@math.u-strasbg.fr

and

Guo-Niu HAN
I.R.M.A.
Université Louis Pasteur et C.N.R.S.
7, rue René-Descartes
F-67084 Strasbourg, France
email: guoniu@math.u-strasbg.fr

Abstract

The purpose of this paper is to give the description of a single algorithm that specializes into several classical transformations derived on words, namely the Cartier-Foata transform, its contextual extension due to Han and the two k-extensions proposed by Clarke and Foata.

1. Introduction. — When sorting was systematically studied in the sixties and seventies, in particular for comparing the different methods used in practice (see, e.g. [14]), it was essential to go back to the classics, to the works by MacMahon and especially to his treatise on Combinatory Analysis [19]. He had made an extensive study of the distributions of several *statistics* on permutations, or more generally, on permutations with repeated elements, simply called *words* in the sequel. The most celebrated of those statistics is probably the classical *number of inversions* which stands for a very natural measurement of how far a permutation is from the identity. There are several other statistics relevant to sorting or to statistical theory, such as the *number of descents*, the *number of excedances*, the *major index*, and more recently the *Denert statistic.*

MacMahon [19] had already calculated the distributions of the early statistics and proved that some of them were equally distributed on each class of rearrangements of a given word. Let us state one of his basic results. To this end suppose that X is a finite non-empty set, referred to as an *alphabet.* For convenience, take X to be the subset $[\,r\,] = \{1, 2, \ldots, r\}$ $(r \geq 1)$ of the positive integers, equipped with its standard ordering. Let $\mathbf{c} = (c_1, c_2, \ldots, c_r)$ be a sequence of r nonnegative integers and v be the nondecreasing word $v = 1^{c_1} 2^{c_2} \ldots r^{c_r}$, i.e., $v = y_1 y_2 \ldots y_m$ with $m = c_1 + c_2 + \cdots + c_r$ and $y_1 = \cdots = y_{c_1} = 1$, $y_{c_1+1} = \cdots = y_{c_1+c_2} = 2$, $\ldots$, $y_{c_1+\cdots+c_{r-1}+1} = \cdots = y_m = r$. The class of all rearrangements of v, i.e., the class of all the words w that can be obtained from v by permuting its letters in some order will be denoted by $R(\mathbf{c})$.

If $w = x_1 x_2 \ldots x_m$ is such a word, the number of excedances, $\operatorname{exc} w$, and the number of descents, $\operatorname{des} w$, of w are classically defined as

$$\begin{aligned} \operatorname{exc} w &= \#\{i : 1 \leq i \leq m, x_i > y_i\}, \\ \operatorname{des} w &= \#\{i : 1 \leq i \leq m-1, x_i > x_{i+1}\}. \end{aligned} \tag{1.1}$$

Let $A_{\mathbf{c}}^{\text{exc}}(t)$ (resp. $A_{\mathbf{c}}^{\text{des}}(t)$) be the generating polynomial for the class $R(\mathbf{c})$ by the statistic "exc" (resp. "des"), i.e.,

$$A_{\mathbf{c}}^{\text{exc}}(t) = \sum_{w} t^{\operatorname{exc} w}, \qquad A_{\mathbf{c}}^{\text{des}}(t) = \sum_{w} t^{\operatorname{des} w} \quad (w \in R(\mathbf{c})).$$

MacMahon [17] showed that those two polynomials were equal for every $\mathbf{c}$. Accordingly, the two statistics "exc" and "des" are *equidistributed* on each rearrangement class $R(\mathbf{c})$. MacMahon also derived an explicit formula for the generating function for those polynomials.

Back in the sixties, when it was natural to prove such equidistribution properties in a *bijective* manner, the late Schützenberger asked whether a bijection Φ could be constructed on each rearrangement class $R(\mathbf{c})$ with the property that

$$\operatorname{exc} w = \operatorname{des} \Phi(w) \tag{1.2}$$

holds for every w. Such a transformation Φ was devised in [10]. A further presentation was made in [14, p. 24–29], a more algebraic version appeared in [4] and also in [16, chap. 10].

In the nineties three extensions of that equidistribution property have been derived. First, by taking pairs of statistics into account, second, by extending the property to the case of an alphabet with two classes of letters; third, by combining those two extensions.

In (1.1) when the *number of the* i's is replaced by the *sum of the* i's we define respectively

$$\begin{aligned} \operatorname{excindex} w &= \sum \{i : 1 \le i \le m, x_i > y_i\}, \\ \operatorname{maj} w &= \sum \{i : 1 \le i \le m-1, x_i > x_{i+1}\}, \end{aligned} \tag{1.3}$$

i.e., the *excedance index* and the *major index* (also called the *greater index* and already studied by MacMahon (*op. cit.*)). The major index has been often associated with the number of descents, for instance in the study of the q-Eulerian polynomials [1, 2, 3]. Moreover, the joint distribution of the pair $(\operatorname{des}, \operatorname{maj})$ was already derived by MacMahon [19, vol. 2, p. 211]. It was then natural to try to introduce a statistic "stat" on words, such that the pair $(\operatorname{des}, \operatorname{maj})$ was equidistributed with the pair $(\operatorname{exc}, \operatorname{stat})$ on each rearrangement class $R(\mathbf{c})$. We could imagine that "excindex" would be the appropriate statistic "stat." However a careful study of the generating polynomials for small values of $\mathbf{c}$ showed that that assumption was not correct. However by adding a *correcting term* to "excindex" a new statistic, "den," called the *Denert statistic*, could be formed and fitted the requirements of "stat."

Surprisingly, "den" was discovered in a very different context, by Denert [9] for the calculation of the genus zeta function of hereditary orders in some simple algebras. Its explicit definition is given in (8.4). The construction of a bijection Φ^q that maps every rearrangement class $R(\mathbf{c})$ onto itself and satisfies

$$(\operatorname{exc}, \operatorname{den})\, w = (\operatorname{des}, \operatorname{maj})\, \Phi^q(w) \tag{1.4}$$

was derived in [12].

The previous results were further extended to the case of an alphabet with two kinds of letters. Keep the same alphabet $\{1, \ldots, r\}$ as above, but suppose given two nonnegative integers s and ℓ such that $s + \ell = r$. A letter x in $[\,r\,]$ is said to be *small* (resp. *large*) if $x \le s$ (resp. if $x \ge s+1$). Whenever such an alphabet is considered we will use the notation ${}_sX_\ell$.

Let $w = x_1x_2\dots x_m$ be a word in the alphabet ${}_sX_\ell$ whose nondecreasing rearrangement is $v = y_1y_2\dots y_m$. An integer i such that $1 \le i \le m$ is said to be a *ℓ-excedance* of w if either $x_i > y_i$, or $x_i = y_i$ and x_i large; it is said to be a *ℓ-descent* of w, if either $x_i > x_{i+1}$, or $x_i = x_{i+1}$ and x_i large (by convention, $x_{m+1} = s + \frac{1}{2}$.) The number of ℓ-excedances (resp. of ℓ-descents) of the word w will be denoted by $\operatorname{exc}_\ell w$ (resp. by $\operatorname{des}_\ell w$).

Again when we *add* the integers i which are ℓ-excedances (resp. ℓ-descents) of w, we define the *ℓ-excedance index*, $\operatorname{excindex}_\ell w$ (resp. the *ℓ-major index*, $\operatorname{maj}_\ell w$), of w.

The extension of the equidistribution property (1.2) consists of constructing a bijection ${}_s\Phi_\ell$ of each rearrangement class onto itself satisfying

$$\operatorname{exc}_\ell w = \operatorname{des}_\ell\ {}_s\Phi_\ell(w). \tag{1.5}$$

Such a bijection was derived in [6].

Finally, an extension of the previous result to bivariate statistics (or, equivalently, an extension of (1.4) to the alphabet with two classes of letters) was proposed in [7] in the following way. When adding a "correcting term" (see section 6) to the statistic "$\operatorname{excindex}_\ell$" we can form the *$\ell$-Denert statistic* "$\operatorname{den}_\ell$." We can then show that the two pairs $(\operatorname{des}_\ell, \operatorname{maj}_\ell)$ and $(\operatorname{exc}_\ell, \operatorname{den}_\ell)$ are equidistributed on each rearrangement class, by constructing a bijection ${}_s\Phi_\ell^q$ of each rearrangement class onto itself satisfying

$$(\operatorname{exc}_\ell, \operatorname{den}_\ell)\, w = (\operatorname{des}_\ell, \operatorname{maj}_\ell)\ {}_s\Phi_\ell^q(w). \tag{1.6}$$

Such a transformation ${}_s\Phi_\ell^q$ was constructed in [7].

The constructions of bijections having properties (1.2), (1.4), (1.5), (1.6) have been given in four papers, namely [4, 12, 6, 7], respectively. Each time the notations used have been different. It was also unclear whether there was a general principle presiding over all those constructions. The purpose of this paper is to give the description of a *single algorithm* (the algorithm $\mathbf{T}$ in section 5) which, once a commutation rule on pairs of letters is given and once a linear ordering on the alphabet X is defined, gives the description of each of the four transformations Φ, Φ^q, ${}_s\Phi_\ell$, ${}_s\Phi_\ell^q$.

There is no special device to introduce for the constructions of ${}_s\Phi_\ell$ and ${}_s\Phi_\ell^q$, as it was done in [6] and [7], where a subsidiary letter "$\star$" was added to the initial alphabet X to make the algorithms work. Also, as was noticed by Knuth [15] in a private communication, there is no need to derive the cancellation law [12, Theorem 3.1] for the contextual multiplication in the construction of Φ^q.

First, we introduce the notion of *biword* and give the description of the *straightening* algorithm of a biword based upon a given sequence of commutations (section 3). Although any commutation rule can be integrated into our main algorithm, the two commutations that yield the constructions of our four bijections are the so-called *Cartier-Foata* commutation and the *contextual* commutation. Both are described in section 4. Our main algorithm $\mathbf{T}$ and its reverse $\mathbf{U} = \mathbf{T}^{-1}$ are presented in sections 5 and 6. In sections 7 and 8 we have gathered the definitions of all the statistics involved. We conclude the paper by giving the main properties of the bijections constructed by means of algorithm $\mathbf{T}$.

2. Words and biwords. — In the sequel the alphabet X will be equipped either with its standard ordering, or with a well-defined ordering. A *biword* is an ordered pair of words of the same length, written as $\alpha = (h, b)$ ("h" stands for "high" and "b" for "bottom") or as

$$\alpha = \binom{h}{b} = \binom{h_1 h_2 \ldots h_m}{b_1 b_2 \ldots b_m}.$$

For easy reference we shall sometimes indicate the *places* 1, 2, ... , m of the letters on the top of the biword:

$$\begin{bmatrix} \mathrm{id} \\ h \\ b \end{bmatrix} = \begin{bmatrix} 1 & 2 & \ldots & m \\ h_1 & h_2 & \ldots & h_m \\ b_1 & b_2 & \ldots & b_m \end{bmatrix}.$$

The word h (resp. b) is the *top* (resp. *bottom*) word of the biword (h, b). Each biword $\binom{h}{b}$ can also be seen as a word whose letters are the *biletters* $\binom{h_1}{b_1}, \ldots, \binom{h_m}{b_m}$. The integer m is the *length* of the biword w. A triple $(h, b; i)$ where i is an integer satisfying $1 \le i \le m-1$ is called a *pointed biword*. When h and b are rearrangements of each other, the biword (h, b) is said to be a *circuit*.

Two classes of *circuits* will play a special role in the paper. First, we introduce the *standard circuits* $\Gamma(b)$ which are circuits of the form $\binom{\overline{b}}{b}$, where $\overline{b}$ is the nondecreasing rearrangement of the word b with respect to the standard ordering. Clearly Γ maps each word onto a standard circuit in a bijective manner.

For the second class of circuits we fix a total order "$\preceq$" on X, not necessarily identical with the standard order and make use of the obvious notations: "$\prec$," "$\succ$," "$\succeq$." A nonempty word $h = h_m h_1 \ldots h_{m-2} h_{m-1}$ is said to be *$\preceq$-dominated*, if $h_m \succ h_1$, $h_m \succ h_2$, ... , $h_m \succ h_{m-1}$. The *right to left cyclic shift* of h is defined to be the word $\delta h = h_1 h_2 \ldots h_{m-1} h_m$. A biword of the form $\binom{\delta h}{h}$ with h $\preceq$-dominated is called a *$\preceq$-dominated cycle.*

As is known (see, e.g. [16, Lemma 10.2.1]) or easily verified, each word b is the juxtaposition product $u^1 u^2 \ldots$ of $\preceq$-dominated words whose first letters $\mathrm{pre}(u^1)$, $\mathrm{pre}(u^2)$, ... are in nondecreasing order with respect to "$\preceq$" :

$$\mathrm{pre}(u^1) \preceq \mathrm{pre}(u^2) \preceq \cdots \tag{2.1}$$

This factorization, called the *increasing factorization* of b, is unique.

Given the increasing factorization $u^1u^2\ldots$ of a word b with respect of an order "$\preceq$," we can form the juxtaposition product

$$\Delta(b) = \begin{pmatrix} \delta u^1 & \delta u^2 & \ldots \\ u^1 & u^2 & \ldots \end{pmatrix} \tag{2.2}$$

of the $\preceq$-dominated cycles. Clearly Δ maps each word onto a product of $\preceq$-dominated cycles satisfying inequalities (2.1), in a bijective manner. Such a product, written as a biword (2.2), will be called a *well-factorized circuit.*

For instance, $4\,|\,4\,5\,|\,1\,|\,3\,1\,2\,|\,3\,5$ is such an increasing factorization of a word b with respect to the ordering $5 \prec 4 \prec 1 \prec 2 \prec 3$. The corresponding well-factorized circuit reads:

$$\Delta(b) = \left(\begin{array}{c|cc|c|ccc|cc} 4 & 5 & 4 & 1 & 1 & 2 & 3 & 5 & 3 \\ 4 & 4 & 5 & 1 & 3 & 1 & 2 & 3 & 5 \end{array}\right).$$

3. Commutations. — Suppose given a four-variable Boolean function $Q(x,y;z,t)$ (also written as $Q\begin{pmatrix} x,y \\ z,t \end{pmatrix}$) defined on quadruples of letters in X. The *commutation* "Com" induced by the Boolean function $Q(x,y;z,t)$ is defined to be a mapping that maps each pointed biword $(h,b;i)$ onto a biword $(h',b') = \mathrm{Com}(h,b;i)$ with the following properties: if

$$\begin{pmatrix} h \\ b \end{pmatrix} = \begin{pmatrix} h_1h_2\ldots h_m \\ b_1b_2\ldots b_m \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} h' \\ b' \end{pmatrix} = \begin{pmatrix} h'_1h'_2\ldots h'_{m'} \\ b'_1b'_2\ldots b'_{m'} \end{pmatrix},$$

then

(C0) $m' = m$;

(C1) $h'_j = h_j$, $b'_j = b_j$ for every $j \neq i, i+1$;

(C2) $h'_{i+1} = h_i$, $h'_i = h_{i+1}$ (the i-th and $(i+1)$-st letters of the top word are transposed);

(C3) $b'_i = b_i$ and $b'_{i+1} = b_{i+1}$ if $Q(h_i, h_{i+1}; b_i, b_{i+1})$ true ; $b'_i = b_{i+1}$ and $b'_{i+1} = b_i$ if $Q(h_i, h_{i+1}; b_i, b_{i+1})$ false.

We can also describe the commutation by the following pair of mappings

$$h = h_1\ldots h_{i-1}h_ih_{i+1}h_{i+2}\ldots h_m \mapsto h' = h_1\ldots h_{i-1}h_{i+1}h_ih_{i+2}\ldots h_m;$$
$$b = b_1\ldots b_{i-1}b_ib_{i+1}b_{i+2}\ldots b_m \mapsto b' = b_1\ldots b_{i-1}\,z\,t\,b_{i+2}\ldots b_m;$$

where either $z = b_i$, $t = b_{i+1}$ if $Q(h_i, h_{i+1}; b_i, b_{i+1})$ true, or $z = b_{i+1}$, $t = b_i$ if $Q(h_i, h_{i+1}; b_i, b_{i+1})$ false.

Definition. — A Boolean function $Q(x, y; z, t)$ is said to be *bi-symmetric* if it is symmetric in the two sets of parameters $\{x, y\}$, $\{z, t\}$.

LEMMA 3.1. — *Let "Com" be the commutation induced by a bi-symmetric Boolean function $Q(x, y; z, t)$. Then the mapping $(h, b; i) \mapsto \mathrm{Com}(h, b; i)$ is involutive.*

The proof of the lemma is a simple verification and will be omitted.

In the rest of the paper we will assume that all the four-variable Boolean functions $Q(x, y; z, t)$ are bi-symmetric.

Two extreme cases are worth being mentioned, when Q is the Boolean function Q_{true} "always true" (resp. Q_{false} "always false"). The commutation $\mathrm{Com}_{\mathrm{true}}$, associated with Q_{true}, permutes only the i-th and $(i+1)$-st *letters* of the *top word* b, while $\mathrm{Com}_{\mathrm{false}}$, associated with Q_{false}, permutes the i-th and $(i+1)$-st *biletters* of the *biword* (h, b).

Sorting a biword is defined as follows. Again consider a biword (h, b) of length m and let $(h', b') = \mathrm{Com}(h, b; i)$ with $1 \le i \le m-1$. If $1 \le j \le m-1$, we can form the pointed biword $(h', b'; j)$ and further apply the commutation "Com" to $(h', b'; j)$. We obtain the biword $\mathrm{Com}(h', b'; j) = \mathrm{Com}(\mathrm{Com}(h, b; i); j)$, which we shall denote by $\mathrm{Com}(h, b; i, j)$. By induction we can define $\mathrm{Com}(h, b; i_1, \ldots, i_n)$, where $(i_1, \ldots, i_n)$ is a given sequence of integers less than m.

As each commutation always permutes two adjacent letters within the *top* word (condition (C2)), we can transform each biword (h, b) into a biword (h', b') whose top word h' is *non-decreasing* by applying a sequence of commutations. We can also say that for each biword (h, b) there exists a sequence $(i_1, \ldots, i_n)$ of integers such that the top word in the resulting biword $\mathrm{Com}(h, b; i_1, \ldots, i_n)$ is non-decreasing. Such a biword is called a *minimal* biword and the sequence $(i_1, \ldots, i_n)$ a *commutation sequence.*

When using the commutations $\mathrm{Com}_{\mathrm{true}}$ or $\mathrm{Com}_{\mathrm{false}}$ we always reach the same minimal biword, but the commutation sequence is not unique. With an arbitrary commutation "Com" neither the minimal biword, nor the commutation sequence are necessarily unique. We then define a particular commutation sequence $(i_1, \ldots, i_n)$ called the *minimal sequence* by the following two conditions:

(i) it is of minimum length;

(ii) it is minimal with respect to the lexicographic order.

Clearly the minimal sequence is uniquely defined by those two conditions and depends only on the top word h in (h, b). The minimal biword derived from (h, b) by using the minimal sequence is called the *straightening* of the biword (h, b). The derivation is described in the following algorithm **SORTB**.

Algorithm SORTB: sorting a biword. — Given a biword (h, b) and a commutation "Com" the following algorithm transforms (h, b) into its straightening (h', b').

Prototype $(h', b') :=$ **SORTB**(h, b, Com).

(1) Let $(h', b') := (h, b)$.
(2) If h' is non-decreasing, **RETURN** (h', b').
(3) Else, let j be the smallest integer such that $h'(j) > h'(j+1)$. Then let $(h', b') := \mathrm{Com}(h', b'; j)$. Go to (2).

4. The two commutations. — We shall introduce two commutations associated with two specific Boolean functions Q.

4.1. *The Cartier-Foata commutation.* — We denote by Com_{CF} the commutation induced by the following Boolean function Q_{CF}:

$$Q_{CF}\begin{pmatrix} x, y \\ z, t \end{pmatrix} \text{ true} \quad \text{if and only if} \quad x = y. \tag{4.1}$$

4.2. *The contextual commutation.* — We now take the alphabet ${}_sX_\ell$ (see the Introduction) and denote by Com_H the commutation induced by the following Boolean function Q_H: let $x^\pm = x + \frac{1}{2}$ if x is small and $x^\pm = x - \frac{1}{2}$ if x is large; then

$$Q_H\begin{pmatrix} x, y \\ z, t \end{pmatrix} \text{ true}$$
$$\text{if and only if} \quad (z - x^\pm)(z - y^\pm)(t - x^\pm)(t - y^\pm) > 0. \tag{4.2}$$

Remarks.

a) Notice that those two commutations are bi-symmetric, so that the mapping $(h, b; i) \mapsto \mathrm{Com}(h, b; i)$ is involutive when Com is equal either to Com_{CF}, or Com_H.

b) *Cyclic intervals.* — The second commutation was introduced in the case $\ell = 0$ (no large letters) in [12] and for an arbitrary ℓ $(0 \le \ell \le r)$ in [7]. In both papers the commutations were defined by means of so-called "cyclic intervals" that can be defined as follows. Place the r elements $1, 2, \ldots, s, s+1, \ldots, (r-1), r$ on a circle counterclockwise and place a bracket on each of those elements as shown in Fig. 1. For $x, y \in X$ $(x \ne y)$ the cyclic interval $]\!]x, y]\!]$ is the subset of all the elements that lie between x and y when the circle is read counterclockwise. The brackets (in the French notation) indicate if the ends of the interval are to be included or not.

For instance, suppose $1 < s < s+1 < r$. Then $]\!]1, s]\!] = \{2, \ldots, s\}$ (the origin 1 excluded, but the end s included), while $]\!]1, s+1]\!] = \{2, \ldots, s\}$

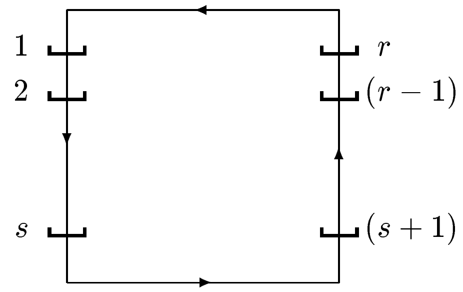

Fig. 1

(both ends are excluded); also $\rrbracket r, s \rrbracket = \{r, 1, \ldots, s\}$ (both ends are included) and $\rrbracket r, s+1 \rrbracket = \{r, 1, \ldots, s\}$ (the origin r included, but the end $(s+1)$ excluded). Finally, define $\rrbracket x, x \rrbracket = \emptyset$ or X depending on whether x is small or large.

Then $Q_H\begin{pmatrix} x, y \\ z, t \end{pmatrix}$ true if both z, t are in $\rrbracket x, y \rrbracket$ or neither in $\rrbracket x, y \rrbracket$.

Example. — Consider the alphabet ${}_sX_\ell$ with $s = 3$ and $\ell = 2$, and the biword $\alpha = \begin{bmatrix} 1\,2\,3\,4\,5\,6\,7\,8\,9 \\ 4\,5\,4\,1\,1\,2\,3\,5\,3 \\ 4\,4\,5\,1\,3\,1\,2\,3\,5 \end{bmatrix}$. The minimal sequence is then $2, 3, 2, 1, 4, 3, 2, 5, 4, 3, 6, 5, 4, 8, 7, 6, 5$. By using Com_H in **SORTB** we obtain the sequence of commutations

$$\alpha = \begin{bmatrix} 1\,\mathbf{2}\,\mathbf{3}\ldots \\ 4\,5\,4\ldots \\ 4\,4\,5\ldots \end{bmatrix} \mapsto \begin{bmatrix} 1\,2\,\mathbf{3}\,\mathbf{4}\ldots \\ 4\,4\,5\,1\ldots \\ 4\,5\,4\,1\ldots \end{bmatrix} \mapsto \begin{bmatrix} 1\,\mathbf{2}\,\mathbf{3}\,4\ldots \\ 4\,4\,1\,5\ldots \\ 4\,5\,1\,4\ldots \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{1}\,\mathbf{2}\,3\,4\ldots \\ 4\,1\,4\,5\ldots \\ 4\,5\,1\,4\ldots \end{bmatrix}$$

$$\cdots \mapsto \begin{bmatrix} 1\,2\,3\,4\,\mathbf{5}\,\mathbf{6}\,7\,8\,9 \\ 1\,1\,2\,3\,4\,3\,4\,5\,5 \\ 4\,5\,1\,4\,1\,3\,2\,3\,5 \end{bmatrix} \mapsto \begin{bmatrix} 1\,2\,3\,4\,5\,6\,7\,8\,9 \\ 1\,1\,2\,3\,3\,4\,4\,5\,5 \\ 4\,5\,1\,4\,1\,3\,2\,3\,5 \end{bmatrix} = \mathbf{SORTB}(\alpha, \mathrm{Com}_H)$$

5. The main algorithm. — In the description of the following algorithm we take the alphabet ${}_sX_\ell$ having s small letters $1, 2, \ldots, s$ and ℓ large letters $(s+1), \ldots, r$ $(s + \ell = r)$.

The total order used on the *places* will always be the natural order of the integers. However the order "$\preceq$" used on the *letters* of the words will be the order "$\preceq$" defined by

$$r \prec (r-1) \prec \cdots \prec (s+1) \prec 1 \prec 2 \prec \cdots \prec s. \tag{5.1}$$

If there are no large letters so that $\ell = 0$ and $s = r$, the order "$\preceq$" is identical with the usual order "$\leq$." To avoid any kind of ambiguity we make precise each time which one of the orderings is involved by using notations such as "$\preceq$-greatest" or "$\leq$-nondecreasing"...

Algorithm T. — Let Com be a commutation induced by a four-variable Boolean function Q. The following algorithm transforms each word b into a rearrangement c of b.

Prototype $c := \mathbf{T}(b, \mathrm{Com})$.

(1) Let h be the $\leq$-nondecreasing rearrangement of b. Form the standard circuit $\Gamma(b) = (h, b)$:

$$\begin{bmatrix} \mathrm{id} \\ h \\ b \end{bmatrix} = \begin{bmatrix} 1 & 2 & \dots & m \\ h_1 & h_2 & \dots & h_m \\ b_1 & b_2 & \dots & b_m \end{bmatrix},$$

let $c := b$ and α be the empty cycle $\alpha = \begin{pmatrix} \cdot \\ \cdot \\ \cdot \end{pmatrix}$.

(2a) If all the places $1, 2, \dots, m$ occur in α, **RETURN** c (the juxtaposition product of the bottom words in α.)

(2b) Else, let D be the $\leq$-greatest *place* not occurring in α.

(2c) Let M be the $\preceq$-greatest *letter* in h not in α. Let i be the $\leq$-rightmost *place* such that $M = h_i$ and $i \leq D$. If $i \leq D-1$ apply the commutation

$$(h, c) := \mathrm{Com}(h, c; i, i+1, \dots, D-1).$$

[If there are no large letters in h, we always have $i = D$ and the above commutation is bypassed.] At the end of (2c) we have $h_D = M$ so that the initial biword has been changed into:

$$\begin{bmatrix} \mathrm{id} \\ h \\ c \end{bmatrix} = \begin{bmatrix} * \dots * & D \\ * \dots * & M \\ * \dots * & * \end{bmatrix} \overbrace{\begin{pmatrix} * \dots * \\ * \dots * \\ * \dots * \end{pmatrix} \cdots \begin{pmatrix} * \dots * \\ * \dots * \\ * \dots * \end{pmatrix}}^{\alpha}.$$

(3a) Let $B := c_D$.

(3b) If $B = M$, terminate the $\preceq$-dominated cycle $\begin{pmatrix} * & \dots & * \\ * & \dots & M \\ M & \dots & * \end{pmatrix}$ (possibly of length 1) and add it to the left of α, so that the new α reads $\begin{pmatrix} * & \dots & * \\ * & \dots & M \\ M & \dots & * \end{pmatrix} \alpha$. Go to (2a).

(3c) Else, look for the $\leq$-greatest *place* $j \leq D-1$ such that $B = h_j$; in short

$$\begin{bmatrix} \mathrm{id} \\ h \\ c \end{bmatrix} = \begin{bmatrix} \dots & j & \dots & D & \dots & * \\ \dots & B & \dots & * & \dots & M \\ \dots & * & \dots & B & \dots & * \end{bmatrix} \begin{pmatrix} * \dots * \\ * \dots * \\ * \dots * \end{pmatrix} \cdots \begin{pmatrix} * \dots * \\ * \dots * \\ * \dots * \end{pmatrix}.$$

If $j \le D-2$, apply the commutation:

$$(h,c) = \mathrm{Com}(h,c;j,j+1,\ldots,D-2),$$

so that $h_{D-1} = B$ after running the commutation; in short:

$$\begin{bmatrix}\mathrm{id}\\ h\\ c\end{bmatrix} = \begin{bmatrix}\ldots j \ldots D-1 & D \ldots & *\\ \ldots * \ldots \quad B & * \ldots & M\\ \ldots * \ldots \quad * & B \ldots & *\end{bmatrix} \begin{pmatrix}* \ldots *\\ * \ldots *\\ * \ldots *\end{pmatrix} \cdots \begin{pmatrix}* \ldots *\\ * \ldots *\\ * \ldots *\end{pmatrix}.$$

(3d) Let $D := D-1$ and go to (3a).

We can verify that each step in the previous algorithm is feasible. For example, the place j in step (3c) is well-defined: at this stage $\binom{h}{c}$ is the product of the left factor (in square brackets) $\left[\begin{smallmatrix}h'\\ c'\end{smallmatrix}\right]$ by α and h' is necessarily a rearrangement of c'.

The algorithm $\mathbf{T}$ will also be denoted by ${}_s\mathbf{T}_\ell$ when we want to emphasize that it is applied to words in the alphabet ${}_sX_\ell$. With $r = s+\ell$ and using the two commutations Com_{CF} and Com_H we can derive four different transformations:

$$\begin{aligned} \Phi(b) &= {}_r\mathbf{T}_0(b,\mathrm{Com}_{CF}), & \Phi^q(b) &= {}_r\mathbf{T}_0(b,\mathrm{Com}_H),\\ {}_s\Phi_\ell(b) &= {}_s\mathbf{T}_\ell(b,\mathrm{Com}_{CF}), & {}_s\Phi^q_\ell(b) &= {}_s\mathbf{T}_\ell(b,\mathrm{Com}_H). \end{aligned} \tag{5.2}$$

Example. — Consider the biword

$$\begin{bmatrix}\mathrm{id}\\ h\\ b\end{bmatrix} = \begin{bmatrix}1\,2\,3\,4\,5\,6\,7\,8\,9\\ 1\,1\,2\,3\,3\,4\,4\,5\,5\\ 4\,5\,1\,4\,1\,3\,2\,3\,5\end{bmatrix}.$$

We calculate the image of b under the four transformations defined in (5.2).

For $\Phi(b) = {}_r\mathbf{T}_0(b,\mathrm{Com}_{CF})$ we obtain

$$\begin{bmatrix}\mathrm{id}\\ h\\ c\end{bmatrix} \mapsto \begin{pmatrix}1\,2\\ 3\,4\\ 4\,3\end{pmatrix} \begin{pmatrix}3\,4\,5\\ 1\,2\,4\\ 4\,1\,2\end{pmatrix} \begin{pmatrix}6\,7\,8\\ 1\,3\,5\\ 5\,1\,3\end{pmatrix} \begin{pmatrix}9\\ 5\\ 5\end{pmatrix},$$

so that $c = \Phi(b) = 4\,3\,4\,1\,2\,5\,1\,3\,5$.

We next calculate $\Phi^q(b) = {}_r\mathbf{T}_0(b,\mathrm{Com}_H)$. We get

$$\begin{bmatrix}\mathrm{id}\\ h\\ c\end{bmatrix} \mapsto \begin{pmatrix}1\,2\,3\,4\,5\,6\,7\,8\\ 4\,4\,1\,1\,3\,2\,3\,5\\ 5\,4\,4\,1\,1\,3\,2\,3\end{pmatrix} \begin{pmatrix}9\\ 5\\ 5\end{pmatrix},$$

so that $c = \Phi^q(b) = {}_r\mathbf{T}_0(b,\mathrm{Com}_H) = 5\,4\,4\,1\,1\,3\,2\,3\,5$.

For the next two transformations assume that 1, 2, 3 are small letters, while 4, 5 are large letters, so that $s = 3$ and $\ell = 2$. The total order "$\preceq$" is then: $5 \prec 4 \prec 1 \prec 2 \prec 3$. The calculation of ${}_s\Phi_\ell(b) = {}_s\mathbf{T}_\ell(b, \mathrm{Com}_{CF})$ yields:

$$\begin{bmatrix} \mathrm{id} \\ h \\ c \end{bmatrix} \mapsto \begin{pmatrix} 1\,2\,3\,4\,5 \\ 4\,1\,2\,4\,3 \\ 3\,4\,1\,2\,4 \end{pmatrix} \begin{pmatrix} 6\,7\,8\,9 \\ 5\,5\,1\,3 \\ 3\,5\,5\,1 \end{pmatrix},$$

so that $c = {}_s\Phi_\ell(b) = {}_s\mathbf{T}_\ell(b, \mathrm{Com}_{CF}) = 3\,4\,1\,2\,4\,3\,5\,5\,1$.

Finally, ${}_s\Phi^q_\ell(b) = {}_s\mathbf{T}_\ell(b, \mathrm{Com}_H)$ is then

$$\begin{bmatrix} \mathrm{id} \\ h \\ c \end{bmatrix} \mapsto \begin{pmatrix} 1 \\ 4 \\ 4 \end{pmatrix} \begin{pmatrix} 2\,3 \\ 5\,4 \\ 4\,5 \end{pmatrix} \begin{pmatrix} 4 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 5\,6\,7 \\ 1\,2\,3 \\ 3\,1\,2 \end{pmatrix} \begin{pmatrix} 8\,9 \\ 5\,3 \\ 3\,5 \end{pmatrix},$$

so that $c = {}_s\Phi^q_\ell(b) = {}_s\mathbf{T}_\ell(b, \mathrm{Com}_H) = 4\,4\,5\,1\,3\,1\,2\,3\,5$.

6. The algorithm U. — Again we take the alphabet ${}_sX_\ell$ and the order defined in (4.1).

Algorithm U. — Given a commutation Com, the following algorithm transforms a word c into a word b such that $b = \mathbf{U}(c, \mathrm{Com})$.

Prototype $b := \mathbf{U}(c, \mathrm{Com})$.
(1) Let $i := 1$, $S := c_1$;
(2a) If $i = \mathrm{length}(c)$, let $h_i := S$, $(h, b) := \mathbf{SORTB}(h, c, \mathrm{Com})$. **RETURN** b.
(2b) Else, let $B := c_{i+1}$.
(2c) If $B \succeq S$, let $h_i := S$, $S := B$. Else, let $h_i := B$.
(3) Let $i := i + 1$. Go to (2a).

PROPERTY. — *Algorithm* $\mathbf{U}$ *is the inverse of algorithm* $\mathbf{T}$*, i.e., for each word b we have*

$$\mathbf{U}(\mathbf{T}(b, \mathrm{Com}), \mathrm{Com}) = \mathbf{T}(\mathbf{U}(b, \mathrm{Com}), \mathrm{Com}) = b.$$

Proof. — First examine algorithm $\mathbf{T}$. Before returning c in step (2a) the algorithm provides the juxtaposition product $\alpha = \gamma^1\gamma^2\ldots$ of cycles. Let u^1, $u^2, \ldots$ be the bottom words of those cycles and let $\mathrm{pre}(u^1)$, $\mathrm{pre}(u^2)$, $\ldots$ be the first letters of those bottom words. Steps (2c) and (3b) say that each cycle γ^i was terminated as soon as $\mathrm{pre}(u^i)$ was $\preceq$-greater than all the other letters in the cycle. Accordingly, all the cycles γ^i are $\preceq$-*dominated.* Furthermore, $\mathrm{pre}(u^1) \preceq \mathrm{pre}(u^2) \preceq \cdots$

Thus $u^1u^2\ldots$ is the increasing factorization of c (in the terminology of section 2), while $\alpha = \gamma^1\gamma^2\ldots = \begin{pmatrix} \delta u^1 & \delta u^2 & \ldots \\ u^1 & u^2 & \ldots \end{pmatrix}$ is the *increasing* product of $\preceq$-dominated cycles, i.e., α is equal to the well-factorized circuit $\Delta(c)$ We can say that the algorithm $\mathbf{T}$ transforms each *standard circuit* $\Gamma(b)$ into a *well-factorized circuit* $\Delta(c)$, the word c being a rearrangement of b. As each commutation applied to a pointed biword is involutive, $\mathbf{T}$ is a bijection.

Now examine Algorithm $\mathbf{U}$ and let $u^1u^2\cdots$ be the increasing factorization of c as a product of $\preceq$-dominated words. Once we have reached step (2a), verified that the test $i=\operatorname{length}(c)$ was positive and executed $h_i := S$, the biword (h,c) is exactly the well-factorized circuit

$$\Delta(c)=\begin{pmatrix} h\\ c\end{pmatrix}=\begin{pmatrix} \delta u^1 & \delta u^2 & \dots\\ u^1 & u^2 & \dots\end{pmatrix}.$$

Thus Algorithm $\mathbf{U}$ first builds up the well-factorized circuit $\Delta(c)$ and applies algorithm **SORTB** to $\Delta(c)$ to produce a standard circuit $\Gamma(b)$. Again as each local commutation applied to a pointed biword is involutive, $\mathbf{U}$ is a bijection.

Finally, to prove that $\mathbf{T}$ and $\mathbf{U}$ are inverse of each other, we simply examine algorithm $\mathbf{T}$. The commutations are made only in steps (2c) and (3c). In both steps the reverse operation can be written as

$$(h,c) := \mathbf{SORTB}(h,c,\mathrm{Com}). \quad \square$$

7. Statistics on circuits. — We keep the alphabet ${}_sX_\ell$ and consider the class of circuits with letters in ${}_sX_\ell$. Remember that a circuit is a pair of words $\alpha=\binom{v}{w}$, where $v=y_1y_2\dots y_m$ and $w=x_1x_2\dots x_m$ are rearrangements of each other and v is *not* necessarily non-decreasing. For each circuit $\alpha=\binom{v}{w}$, we define $\operatorname{exc}_\ell\binom{v}{w}$ to be the number of integers i such that $1\le i\le m$ and either $x_i>y_i$, or $x_i=y_i$ and x_i large.

The definition of $\operatorname{den}_\ell\binom{v}{w}$ is based on the notion of *cyclic interval*, as introduced in section 4. For each place i $(1\le i\le m)$ define p_i to be the number of j such that $1\le j\le i-1$ and $x_j\in \rrbracket x_i,y_i\rrbracket$. Also let p_{m+1} be the number of large letters in w. The sequence $(p_1,p_2,\dots,p_{m+1})$ is said to be the *ℓ-den-coding* of $\binom{v}{w}$. Furthermore, define

$$\operatorname{den}_\ell\begin{pmatrix} v\\ w\end{pmatrix}=p_1+p_2+\cdots+p_m+p_{m+1}. \tag{7.1}$$

Example. — Consider an alphabet with $s=3$ small letters 1, 2, 3 and $\ell=2$ large letters 4, 5 and form the circuit

$$\alpha=\begin{bmatrix}\mathrm{id}\\ v\\ w\end{bmatrix}=\begin{bmatrix}1&2&3&4&5&6&7&8&9\\ 4&5&4&1&1&2&3&5&3\\ 4&4&5&1&3&1&2&3&5\end{bmatrix}.$$

It has a ℓ-excedence at places 1, 3, 5, 9, so that $\operatorname{exc}_\ell \alpha = 4$. For its ℓ-Denert coding we first have $p_1 = 0$. As $4 \in \rrbracket 4, 5 \rrbracket = \{4\}$, we have $p_2 = 1$. As $4 \notin \rrbracket 5, 4 \rrbracket = \{5, 1, 2, 3\}$, we have $p_3 = 0$. Also $p_4 = 0$ as $\rrbracket 1, 1 \rrbracket$ is empty. As 4, 4, 5, 1 belong to $\rrbracket 3, 1 \rrbracket = \{4, 5, 1\}$, $p_5 = 4$. As $\rrbracket 1, 2 \rrbracket = \{2\}$, we have $p_6 = 0$. Also $\rrbracket 2, 3 \rrbracket = \{3\}$, so that $p_7 = 1$ and $\rrbracket 3, 5 \rrbracket = \{4\}$, so that $p_8 = 2$. As $\rrbracket 5, 3 \rrbracket = \{5, 1, 2, 3\}$, we get $p_9 = 6$. Finally, as there are 4 large letters in w, we get $p_{10} = 4$. Thus $\operatorname{den}_\ell \alpha = 0{+}1{+}0{+}0{+}4{+}0{+}1{+}2{+}6{+}4 = 18$.

The definitions of $\operatorname{des}_\ell \binom{v}{w}$ and of $\operatorname{maj}_\ell \binom{v}{w}$ depend only on the bottom word w. Put $x_{m+1} = s$ (the largest small letter) and $x_{m+2} = \infty$ (an auxiliary letter greater than every letter of X). Then for each $i = 1, 2, \ldots, m+1$ define q_i to be the number of j such that $1 \le j \le i-1$ and $x_j \in \rrbracket x_i, x_{i+1} \rrbracket$. The sequence $(q_1, q_2, \ldots, q_m, q_{m+1})$ is said to be the *ℓ-maj-coding* of α. Define

$$\operatorname{maj}_\ell \begin{pmatrix} v \\ w \end{pmatrix} = q_1 + q_2 + \cdots + q_m + q_{m+1}. \tag{7.2}$$

Taking the same example as above we can verify that $(\operatorname{des}_\ell, \operatorname{maj}_\ell)\, \alpha = (4, 18)$, which is also the value of $(\operatorname{exc}_\ell, \operatorname{den}_\ell)\, \alpha$. In fact the above circuit is well-factorized and for that class of circuits the two pairs of statistics have the same values, as proved in the next theorem.

Theorem 7.1. — *If $\alpha = \binom{v}{w}$ is a well-factorized circuit, then*

$$(\operatorname{exc}_\ell, \operatorname{den}_\ell)\, \alpha = (\operatorname{des}_\ell, \operatorname{maj}_\ell)\, \alpha. \tag{7.3}$$

Proof. — Let $\begin{pmatrix} a_2 & \ldots & a_{i+1} & a_{i+2} & \ldots & a_k & a_1 \\ a_1 & \ldots & a_i & a_{i+1} & \ldots & a_{k-1} & a_k \end{pmatrix}$ and $\begin{pmatrix} b_2 & \ldots \\ b_1 & \ldots \end{pmatrix}$ be two successive $\preceq$-dominated cycles in the increasing factorization of α, so that

$$\alpha = \begin{pmatrix} \ldots & a_2 & \ldots & a_{i+1} & a_{i+2} & \ldots & a_k & a_1 & b_2 & \ldots \\ \ldots & a_1 & \ldots & a_i & a_{i+1} & \ldots & a_{k-1} & a_k & b_1 & \ldots \end{pmatrix}.$$

Inside each $\preceq$-dominated cycle a pair like (a_i, a_{i+1}) occurs horizontally and vertically, so that there is a ℓ-descent $a_i a_{i+1}$ if and only if there is a ℓ-excedance $\binom{a_{i+1}}{a_i}$. Furthermore, the letters in w to the left of a_i that fall into the cyclic interval $\rrbracket a_i, a_{i+1} \rrbracket$ bring the same contribution in both $\operatorname{maj}_\ell \alpha$ and $\operatorname{den}_\ell \alpha$. If $\binom{a_{i+1}}{a_i}$ is the j-th biletter of α (when read from left to right), we have $p_j = q_j$ in the notations used in (7.2) and (7.3).

At the end of a $\preceq$-dominated cycle we have to compare the contributions of the horizontal pair (a_k, b_1) with the contribution of the vertical pair $\binom{a_1}{a_k}$. But $a_k \prec a_1 \preceq b_1$ by definition of the increasing factorization. If b_1 is large, then a_k and a_1 are also large and we have $a_k > a_1 \geq b_1$, so that the letter a_k creates an ℓ-excedence and an ℓ-descent. If b_1 is small, then $a_k > a_1$ is an ℓ-excedance if and only if a_k is large. This is also equivalent to saying that $a_k > b_1$ is an ℓ-descent.

Now if $a_1 = b_1$, the two cyclic intervals $\rrbracket a_k, b_1 \rrbracket$ and $\rrbracket a_k, a_1 \rrbracket$ that serve in the calculation of $\mathrm{maj}_\ell\, \alpha$ and $\mathrm{den}_\ell\, \alpha$ are identical. If $a_1 \prec b_1$, there is no letter x in w to the left of b_1 such that $a_1 \prec x \preceq b_1$. Four cases are to be considered: (1) a_1, b_1, a_k all small; (2) a_1, b_1 small, a_k large; (3) a_1, b_1, a_k all large; (4) a_1 large, b_1 small, a_k large. For any two sets A, B let $A + B$ denote the union of A and B when the intersection $A \cap B$ is empty. In cases (1) and (2) we have

$$\rrbracket a_k, b_1 \rrbracket = \rrbracket a_k, a_1 \rrbracket + \{x : a_1 \prec x \preceq b_1\},$$

while

$$\rrbracket a_k, a_1 \rrbracket = \rrbracket a_k, b_1 \rrbracket + \{x : a_1 \prec x \preceq b_1\}$$

in cases (3) and (4). Consequently there are as many letters to the left of a_k falling into the interval $\rrbracket a_k, b_1 \rrbracket$ as letters falling into $\rrbracket a_k, a_1 \rrbracket$.

Suppose that α is of length m and take up again the notations of (7.1) and (7.2). We still have to compare the pairs (q_m, q_{m+1}) and (p_m, p_{m+1}). Let $\binom{y_m}{x_m}$ be the rightmost biletter of α. If w contains small letters, the letter y_m is necessarily equal to the greatest small letter occurring in w. Hence the cyclic intervals $\rrbracket x_m, s \rrbracket$ used for evaluating q_m and $\rrbracket x_m, y_m \rrbracket$ for evaluating p_m are equal. If w contains only large letters, y_m is equal to the smallest large letter in w and again the previous two cyclic intervals are equal. Finally, $p_{m+1} = q_{m+1}$, as both coefficients count the large letters in w. $\square$

8. Statistics on words. — To get the definitions of $\mathrm{des}_\ell\, w$, $\mathrm{maj}_\ell\, w$, $\mathrm{exc}_\ell\, w$ and $\mathrm{den}_\ell\, w$ for a word w in the alphabet ${}_sX_\ell$ we simply form the *standard circuit* $\Gamma(w)$ and put

$$\begin{aligned} \mathrm{des}_\ell\, w &= \mathrm{des}_\ell\, \Gamma(w), & \mathrm{maj}_\ell\, w &= \mathrm{maj}_\ell\, \Gamma(w), \\ \mathrm{exc}_\ell\, w &= \mathrm{exc}_\ell\, \Gamma(w), & \mathrm{den}_\ell\, w &= \mathrm{den}_\ell\, \Gamma(w). \end{aligned} \tag{8.1}$$

The definitions given for $\mathrm{des}_\ell\, w$ and $\mathrm{exc}_\ell\, w$ are identical with the definitions given in the introduction. The definition of $\mathrm{den}_\ell\, w$ is new, while that of $\mathrm{maj}_\ell\, w$ differs from the definition given in the introduction.

THEOREM 8.1. — *The statistic* $\mathrm{maj}_\ell\, w$ *given in* (8.1) *and the statistic* $\mathrm{maj}_\ell\, w$ *given in the introduction are identical.*

As can be found in [12] for $\ell = 0$ and in [7] for an arbitrary ℓ, Theorem 8.1 is easy to prove by induction on the length of the word.

The *ℓ-excedance index* of w was defined in the introduction as the sum, $\operatorname{excindex}_\ell w$, of all i such that i is a ℓ-excedance in w. When a certain correcting term is added to $\operatorname{excindex}_\ell w$, we get the second definition of $\operatorname{den}_\ell w$. To fully describe that correcting term we need the further definitions. For each word $w = x_1 x_2 \ldots x_m$ let

$$\begin{aligned}\operatorname{inv}_\ell w &= \#\{1 \le i < j \le m : x_i > x_j \text{ or } x_i = x_j \ge s+1\} \\ &\qquad\qquad + \#\{1 \le i \le m : x_i \ge s+1\}, \\ \operatorname{imv}_\ell w &= \#\{1 \le i < j \le m : x_i > x_j \text{ or } x_i = x_j \le s\}.\end{aligned} \tag{8.2}$$

Notice that $\operatorname{inv}_0 = \operatorname{inv}$ (the usual number of inversions) and $\operatorname{imv}_0 = \operatorname{imv}$ (the number of weak inversions).

Now if $\operatorname{exc}_\ell w = e$, let $i_1 < i_2 < \cdots < i_e$ be the increasing sequence of the ℓ-excedances of w and let $j_1 < j_2 < \cdots < j_{m-e}$ be the complementary sequence. Form the two subwords

$$\operatorname{Exc}_\ell w = x_{i_1} x_{i_2} \ldots x_{i_e}\,; \qquad \operatorname{Nexc}_\ell w = x_{j_1} x_{j_2} \ldots x_{j_{m-e}}.$$

Then the *ℓ-Denert statistic* of w is also defined to be

$$\operatorname{den}_\ell w = \operatorname{excindex}_\ell w + \operatorname{imv}_\ell \operatorname{Exc}_\ell w + \operatorname{inv}_\ell \operatorname{Nexc}_\ell w. \tag{8.3}$$

For $\ell = 0$ we simply speak of the *Denert statistic*

$$\operatorname{den} w = \operatorname{den}_0 w. \tag{8.4}$$

THEOREM 8.2. — *For every word w the two definitions of* $\operatorname{den}_\ell w$ *occurring in* (8.1) *and* (8.3) *are identical.*

The concept of the Denert statistic is due to Denert [9]. The equivalence between definitions (8.1) and (8.3) when $\ell = 0$ and when w is a permutation (without repetitions) was already given in [11]. Another proof of that result appeared in [5]. Theorem 8.2 that states the equivalence of those two definitions for arbitrary words was proved in [12] for $\ell = 0$ and in [13] for an arbitrary ℓ.

9. Equidistribution properties. — The fundamental properties of the commutations Com_{CF} and Com_H introduced in section 4 are stated in the next theorem. The underlying alphabet is ${}_sX_\ell$.

THEOREM 9.1. — *The commutation "*Com*" being given, let $\alpha = (v, w)$ be a circuit of length m and let $\alpha' = (v', w') = \operatorname{Com}(v, w; i)$ $(1 \le i \le m-1)$ be the image of that pointed circuit under the commutation "*Com.*"*

If Com $=$ Com$_{CF}$*, then*

$$\operatorname{exc}_\ell \begin{pmatrix} v' \\ w' \end{pmatrix} = \operatorname{exc}_\ell \begin{pmatrix} v \\ w \end{pmatrix}. \tag{9.1}$$

If Com $=$ Com$_H$*, then*

$$(\operatorname{exc}_\ell, \operatorname{den}_\ell) \begin{pmatrix} v' \\ w' \end{pmatrix} = (\operatorname{exc}_\ell, \operatorname{den}_\ell) \begin{pmatrix} v \\ w \end{pmatrix}. \tag{9.2}$$

Property (9.1) is evident, as biletters are preserved when the commutation "Com$_{CF}$" is applied. With notations slightly different (9.2) has been proved in [12, Lemma 5.2] in the case $\ell = 0$ and for arbitrary ℓ in [7, Lemma 3.6]. We do not reproduce the proof here, as it essentially consists of a lengthy but easy verification.

Theorem 9.2. — *The transformations*

$$\Phi : b \mapsto {}_r\mathbf{T}_0(b, \mathrm{Com}_{CF}), \qquad \Phi^q : b \mapsto {}_r\mathbf{T}_0(b, \mathrm{Com}_H),$$
$${}_s\Phi_\ell : b \mapsto {}_s\mathbf{T}_\ell(b, \mathrm{Com}_{CF}), \qquad {}_s\Phi^q_\ell : b \mapsto {}_s\mathbf{T}_\ell(b, \mathrm{Com}_H).$$

defined in (5.2) *have the equidistribution properties*

$$\operatorname{exc}(w) = \operatorname{des}\, \Phi(w), \qquad (\operatorname{exc}, \operatorname{den})\, w = (\operatorname{des}, \operatorname{maj})\, \Phi^q(w),$$
$$\operatorname{exc}_\ell(w) = \operatorname{des}_\ell\, {}_s\Phi_\ell(w), \quad (\operatorname{exc}_\ell, \operatorname{den}_\ell)\, w = (\operatorname{des}_\ell, \operatorname{maj}_\ell)\, {}_s\Phi^q_\ell(w).$$

Proof. — With each of those transformations we start with a word b, form the standard circuit $\Gamma(b) = \binom{\overline{b}}{b}$ and apply a well-defined sequence of Cartier-Foata commutations (resp. of contextual commutations) to reach a well-factorized circuit $\Delta(c)$.

If the commutation used is the Cartier-Foata commutation, we have

$$\begin{aligned}
\operatorname{exc}_\ell(b) &= \operatorname{exc}_\ell \begin{pmatrix} \overline{b} \\ b \end{pmatrix} && \text{[by definition of "exc}_\ell\text{"]}\\
&= \operatorname{exc}_\ell\, \Delta(c) && \text{[because of (9.1)]}\\
&= \operatorname{des}_\ell\, \Delta(c) && \text{[by Theorem 7.1]}\\
&= \operatorname{des}_\ell(c). && \text{[by definition of "des}_\ell\text{."]}
\end{aligned}$$

If we use the contextual commutation, we have the properties

$$\begin{aligned}
(\operatorname{exc}_\ell, \operatorname{den}_\ell)\, (b) &= (\operatorname{exc}_\ell, \operatorname{den}_\ell) \begin{pmatrix} \overline{b} \\ b \end{pmatrix} && \text{[by definition of "(exc}_\ell\text{, den}_\ell\text{)"]}\\
&= (\operatorname{exc}_\ell, \operatorname{den}_\ell)\, \Delta(c) && \text{[because of (9.2)]}\\
&= (\operatorname{des}_\ell, \operatorname{maj}_\ell)\, \Delta(c) && \text{[by Theorem 7.1]}\\
&= (\operatorname{des}_\ell, \operatorname{maj}_\ell)\, (c). \quad \square && \text{[by definition of "maj}_\ell\text{."]}
\end{aligned}$$

Acknowledgements. — The authors should like to thank the referee for his careful reading.

REFERENCES

1. L. Carlitz, q-Bernoulli and Eulerian numbers, *Trans. Amer. Math. Soc.* **76** (1954), 332–350.
2. L. Carlitz, Eulerian numbers and polynomials, *Math. Magazine* **33** (1959), 247–260.
3. L. Carlitz, A combinatorial property of q-Eulerian numbers, *Amer. Math. Monthly* **82** (1975), 51–54.
4. P. Cartier and D. Foata, "Problèmes combinatoires de permutations et réarrangements," Berlin, Springer-Verlag, 1969 (*Lecture Notes in Math.*, **85**).
5. R. J. Clarke, A short proof of a result of Foata and Zeilberger, *Adv. Appl. Math.* **16** (1995), 129–131.
6. R. J. Clarke and D. Foata, Eulerian Calculus, I: Univariable Statistics, *Europ. J. Combinatorics* **15** (1994), 345–362.
7. R. J. Clarke and D. Foata, Eulerian Calculus, II: An Extension of Han's Fundamental Transformation, *Europ. J. Combinatorics* **16** (1995), 221–252.
8. R. J. Clarke and D. Foata, Eulerian Calculus, III: The Ubiquitous Cauchy Formula, *Europ. J. Combinatorics* **16** (1995), 329–355.
9. M. Denert, The genus zeta function of hereditary orders in central simple algebras over global fields, *Math. Comp.* **54** (1990), 449–465.
10. D. Foata, Etude algébrique de certains problèmes d'analyse combinatoire et du calcul des probabilités, *Publ. Inst. Statist. Univ. Paris* **14** (1965), 81–241.
11. D. Foata and D. Zeilberger, Denert's permutation statistic is indeed Euler-Mahonian, *Studies in Appl. Math.* **83** (1990), 31–59.
12. G. N. Han, Une transformation fondamentale sur les réarrangements de mots, *Advances in Math.* **105** (1994), 26–41.
13. G. N. Han, The k-Extension of a Mahonian Statistic, *Adv. Appl. Math.* **16** (1995), 297–305.
14. D. E. Knuth, "The Art of Computer Programming, vol. 3, Sorting and Searching," Addison-Wesley, Reading, 1973.
15. D. E. Knuth, Private communication, 1996.
16. M. Lothaire, "Combinatorics on words," Reading, Addison-Wesley, 1983 (*Encyclopedia of Math. and its Appl.*, **17**).

17. P. A. MacMahon, The indices of permutations and the derivation therefrom of functions of a single variable associated with the permutations of any assemblage of objects, *Amer. J. Math.* **35** (1913), 314–321.
18. P. A. MacMahon, "Collected Papers, vol. 1" [G.E. Andrews, ed.], Cambridge, Mass., The M.I.T. Press, 1978.
19. P. A. MacMahon, "Combinatory Analysis, vol. 1," Cambridge, Cambridge Univ. Press, 1915 (Reprinted by Chelsea, New York, 1955).