

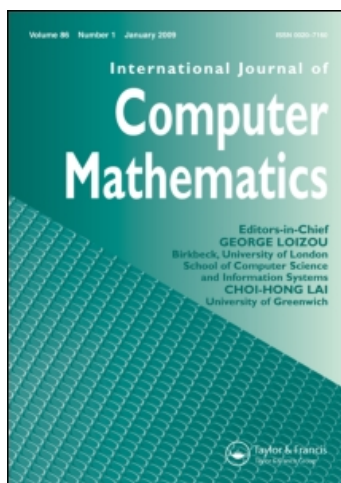
This article was downloaded by:

On: 7 January 2010

Access details: *Access Details: Free Access*

Publisher *Taylor & Francis*

Informa Ltd Registered in England and Wales Registered Number: 1072954 Registered office: Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK



## International Journal of Computer Mathematics

Publication details, including instructions for authors and subscription information:

<http://www.informaworld.com/smpp/title~content=t713455451>

## Investigation of block-sorting of multiset permutations

Ziya Arnavut <sup>a</sup>; Meral Arnavut <sup>a</sup>

<sup>a</sup> Department of Mathematics and Computer Science, SUNY Fredonia, NY, USA

**To cite this Article** Arnavut, Ziya and Arnavut, Meral(2004) 'Investigation of block-sorting of multiset permutations', International Journal of Computer Mathematics, 81: 10, 1213 – 1222

**To link to this Article:** DOI: 10.1080/00207160410001712279

**URL:** <http://dx.doi.org/10.1080/00207160410001712279>

PLEASE SCROLL DOWN FOR ARTICLE

Full terms and conditions of use: <http://www.informaworld.com/terms-and-conditions-of-access.pdf>

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

The publisher does not give any warranty express or implied or make any representation that the contents will be complete or accurate or up to date. The accuracy of any instructions, formulae and drug doses should be independently verified with primary sources. The publisher shall not be liable for any loss, actions, claims, proceedings, demand or costs or damages whatsoever or howsoever caused arising directly or indirectly in connection with or arising out of the use of this material.

# INVESTIGATION OF BLOCK-SORTING OF MULTISET PERMUTATIONS\*

ZIYA ARNAVUT<sup>†</sup> and MERAL ARNAVUT<sup>‡</sup>

*Department of Mathematics and Computer Science, SUNY Fredonia, NY 14063, USA*

*(Revised 26 January 2004; In final form 2 February 2004)*

A recent development in data compression area is Burrows–Wheeler Compression algorithm (BWCA). Introduced by Burrows and Wheeler, the BWCA achieves compression ratio closer to the best compression techniques, such as partial pattern matching (PPM) techniques, but with a faster execution speed. In this paper, we analyze the combinatorial properties of the Burrows–Wheeler transformation (BWT), which is a block-sorting transformation and an essential part of the BWCA, introduce a new transformation, and delineate the new transformation with the BWT based on the multiset permutations.

*Keywords:* Block-sorting; BWT transformation; Multiset permutations; Lossless compression

*C.R. Categories:* E.4.0; G.2.1

## 1 INTRODUCTION

Although the communication channels and storage mediums have been getting bigger and cheaper, the amount of information we require or desire in our daily lives is increasing at an even faster rate. Entertainment, telecommunications and the Internet are part of our daily life. People enjoy movies and music, use telephones to communicate with their friends, families or conduct business, and utilize the Internet to surf, read on-line news, search for information and communicate. Unlike in the past, today we use digital communication systems and networks, and digital representation of movies, newspapers, articles, books, TV, music, images and voice.

As the amount of information that is needed and available increases, a tremendous amount of data are communicated through networks and other communication channels. To handle the traffic and increase the throughput, effective data compression schemes are needed.

Data compression is the process of transforming an input data stream (the source stream or original raw data) into another data stream (the output or the compressed stream) that has smaller size. A stream is either a file or buffer in the memory. Data compression is popular for two reasons: (1) People like to accumulate things and hate to throw anything away. No matter

---

\* Some sections of this work have been presented at two different meetings of the IEEE Data Compression Conference, DCC-1998 and DCC-2002.

<sup>†</sup> Corresponding author. Tel.: (716)-673-3864; Fax: (716)-763-3804; E-mail: arnavut@cs.fredonia.edu

<sup>‡</sup> E-mail: arnavut@cs.fredonia.edu

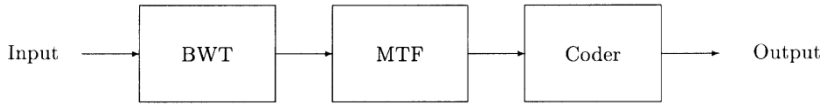


FIGURE 1 Burrows–Wheeler Compression algorithm.

how big a storage device one has, sooner or later it will overflow. Data compression is useful because it delays this inevitability. (2) People hate to wait a long time for data transfers. When waiting for a web page to show up on the screen or when downloading a file, human beings naturally feel that anything longer than a few seconds is too long to wait.

One of the recent developments in the data compression area is the block-sorting lossless data compression algorithm (also known as BW94, block-sorting coder, Burrows–Wheeler compression) technique introduced by Burrows and Wheeler [9]. We use the name Burrows–Wheeler Compression algorithm (BWCA) as the name BWCA is more widely in use. When applied to text or image data, BWCA achieves better compression rates than Ziv–Lempel techniques with comparable speed, while its compression performance is close to that of context-based methods, such as partial pattern matching (PPM) based techniques.

The scheme of the BWCA is presented in Figure 1. The first step performs the lexical (alphabetical) sorting transformation, which is widely known as Burrows–Wheeler transformation (BWT).

The second major step of the BWCA is the move-to-front (MTF) coder. Introduced by Bentley *et al.* [8] (independently discovered by Elias [14]), MTF coding is an adaptive technique, which is used when the data have locality of reference [8]. The MTF coder has been an essential part of BWCA and all other variants, such as Bzip [20], Szip [18] and others. Since the introduction of BWCA, several researchers [6, 7, 10, 12, 13, 15, 17, 21] have modified or replaced the MTF coding technique in BWCA.

The final stage of the BWCA coder is a statistical compressor, such as an arithmetic [22] or Huffman coder [16].

While many researchers investigated on improving the compression gain by modifying or replacing MTF coder in the BWCA, little work has been done on the mathematical theory of BWT. The first mathematical analyzes of BWT based on group theory appeared in Refs. [1–3]. Recently, the authors [4] investigated possibility of other block transformations for multiset permutations, and in Ref. [5] the author generalized the BWT for multiset permutations. Schindler [18, 19] gives an algorithmic, non-mathematical description of the block-sorting technique associated with the BWT. Later, Schindler’s work was explained more throughly by Yokoo [23].

This work further investigates combinatorial properties of block-sorting schemes using multiset permutation (data strings), introduces a different block-sorting transformation, linear order transformation (LOT), and investigates the relationship between LOT and BWT.

## 2 MATHEMATICAL PRELIMINARIES

In this work, we assume knowledge of some basic mathematical concepts, including familiarity with elementary properties of standard objects of discrete mathematics. However, to set the stage for our later discussion, we begin by recalling some definitions and corresponding notations.

By a permutation  $\pi$  of a finite set  $X$  we mean a bijection (*i.e.* a one-to-one and onto function) from  $X$  onto itself. We use standard functional notation to denote permutations;

for example, if  $X = \{x_1, x_2, x_3, x_4, x_5\}$  and  $\pi: X \rightarrow X$  such that  $\pi(x_1) = x_2, \pi(x_2) = x_3, \pi(x_3) = x_1, \pi(x_4) = x_5$  and  $\pi(x_5) = x_4$ , then we denote  $\pi$  by:

$$\pi = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \\ x_2 & x_3 & x_1 & x_5 & x_4 \end{pmatrix}.$$

If we select a particular fixed order for the elements of  $X$ , once and for all, say  $[x_1, x_2, x_3, x_4, x_5]$ , then we can specify  $\pi$  by simply writing the sequence corresponding to the bottom row,  $[x_2, x_3, x_1, x_5, x_4]$ , in the functional notation above. Thus, we can also write:

$$\pi = [x_2, x_3, x_1, x_5, x_4].$$

This is called the Cartesian form of  $\pi$ . The set of all permutations on  $X$  forms a group under functional composition, called the symmetric group on  $X$ , and denoted by  $\mathcal{S}_X$ . If  $X = \{1, 2, \dots, n\}$ , then we simply denote  $\mathcal{S}_X$  by  $\mathcal{S}_n$ .

The idea of permutations on a set  $X$  can be extended to multiset. By a multiset  $\mathcal{M}$  based on a set  $X$  we mean a pair  $(X, f)$ , where  $f: X \rightarrow \mathbb{N}$  is a function from  $X$  into  $\mathbb{N}$  and called as the frequency (multiplicity) function. The size of  $\mathcal{M}$  is defined by  $|\mathcal{M}| = \sum_{x \in X} f(x)$ , say  $|\mathcal{M}| = n$ .

A multiset permutation  $\omega$  based on multiset  $\mathcal{M} = (X, f)$  is a mapping  $\omega: \{1, 2, \dots, n\} \rightarrow X$ , such that if  $x \in X$

$$f(x) = |\{j: 1 \leq j \leq n, \omega[j] = x\}| =: f_x.$$

This mapping is onto but not always one-to-one. Thus, unlike permutations, multiset permutations are not always bijective maps. Intuitively, we can think of a multiset permutation  $\omega: \{1, 2, \dots, n\} \rightarrow X$  as a linear array  $[\omega[1], \omega[2], \dots, \omega[n]]$ , where  $x \in X$  appears as an element in  $\omega$  exactly  $f(x)$  times. If the underlying set has an implicit or explicit linear order, say  $X = \{x_1, x_2, \dots, x_m\}$  with  $x_1 < x_2 < \dots < x_m$ , then we sometimes denote the multiset by product-exponential form

$$\mathcal{M} = x_1^{f_{x_1}} \cdot x_2^{f_{x_2}} \cdot \dots \cdot x_m^{f_{x_m}},$$

where  $f_{x_i} (= f(x_i))$  is the frequency of element  $x_i$ .

From now on, in this paper, we assume  $X = \{1, 2, \dots, m\}$ . Hence,  $[2, 1, 3, 3, 1, 3, 2]$  is a multiset permutation of  $\mathcal{M} = (1, 1, 2, 2, 2, 3, 3) = 1^2 \cdot 2^3 \cdot 3^2$ . In functional notation, a multiset permutation  $\sigma = [1, 3, 3, 1, 2]$  is represented as

$$\sigma = \begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 1 & 3 & 3 & 1 & 2 \end{pmatrix}.$$

Unlike permutations, multiset permutations do not form a group under functional composition. We define a new operation denoted by ‘ $\cdot$ ’, on multiset permutations as follows.

Let  $\sigma = [1, 3, 3, 1, 2]$  and  $\phi = [3, 1, 1, 2, 3]$  be multiset permutations represented in the Cartesian form defined on  $\mathcal{M} = (1, 1, 2, 3, 3)$ . By the  $\cdot$  product of two multiset permutations,  $\sigma$  and  $\phi$ , we mean that  $\sigma$  and  $\phi$  are to be performed in that order. For example, in functional form,

$$\sigma \cdot \phi = \begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 1 & 3 & 3 & 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 3 & 1 & 1 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 3 & 2 & 3 & 1 & 1 \end{pmatrix} =: \theta.$$

To form  $\theta$ , scan the top row of  $\sigma$  from left to right, and for each value  $i$ , which appears at the top row of  $\sigma$ , determine  $\sigma(i) = j$ , where  $j$  is the value that occurs below  $i$  at the bottom row of  $\sigma$ . If  $j$  occurred  $l$  times previously, find the  $l$ th  $j$  in the top row of  $\phi$ , by scanning from left to right, and determine the value  $k$ , which occurs at the bottom row under  $j$  in  $\phi$ . If,  $\phi(j) = k$  is determined, then  $\sigma \cdot \phi(i) = k = \theta(i)$ ; so, for our example, in the first iteration, we see that the first entry of  $\sigma$  has a value 1, and  $\sigma(1) = 1$ ; while the first entry at the top row of  $\phi$  is 1, and  $\phi(1) = 3$ . Thus,  $\theta(1) = 3$ . In the second iteration we use the second 1 from the top row of the functional representation of  $\sigma$  and because  $\sigma(1) = 3$ , we determine where 3 is mapped in  $\phi$ . Since this is the first time that a 3 has occurred,  $\phi(3) = 2$ . Thus,  $\theta(1) = 2$ . In the third iteration,  $\sigma(2) = 3$ . As this is the second occurrence of 3, the second 3 from the top row of the functional representation of  $\phi$  is utilized. Thus,  $\phi(3) = 3$  and  $\theta(2) = 3$ . Repeating the process in this manner, we form  $\theta$ .

In this work, we define the identity permutation of a multiset  $\mathcal{M} = 1^{f_1} \cdot 2^{f_2} \dots m^{f_m}$  as  $I_{\mathcal{M}} = [1, 1, \dots, 1, 2, 2, \dots, m, m, \dots, m]$ . Note that, the elements in  $I_{\mathcal{M}}$  give rise to a partition of blocks,  $B_1, B_2, \dots, B_m$ , where for all  $v \in B_i, v = i$  and  $f_i = |B_i|$ , for  $1 \leq v \leq m$ . We define that the image of an element  $x \in X$  of a multiset permutation  $\sigma$ ,  $\text{Im}_{\sigma}(x)$ , is the set of all elements where  $x$ s at the top row of the functional representation of  $\sigma$  are mapped to the bottom row. For example,  $\text{Im}_{\sigma}(1) = \{1, 3\}$ ,  $\text{Im}_{\sigma}(2) = \{3\}$  and  $\text{Im}_{\sigma}(3) = \{1, 2\}$ . Similarly, we define that the preimage of an element  $x \in X$  of a multiset permutation  $\sigma$ ,  $\text{Pim}_{\sigma}(x)$ , is the set of all elements where  $x$ s at the bottom row of the functional representation of  $\sigma$  are mapped to the top row elements. For example,  $\text{Pim}_{\phi}(1) = \{1, 2\}$ ,  $\text{Pim}_{\phi}(2) = \{3\}$  and  $\text{Pim}_{\phi}(3) = \{1, 3\}$ .

Let  $\omega$  be a multiset permutation. By taking the set of all the possible permutations of the preimages of each different element in  $\omega$ , we define the inverse set of  $\omega$  and denote it by  $T_{\omega}$ . For example, in the case of the multiset permutation  $\phi$ , the inverse set  $T_{\phi}$  is

$$\{[1, 2, 3, 1, 3], [1, 2, 3, 3, 1], [2, 1, 3, 1, 3], [2, 1, 3, 3, 1]\}, \tag{1}$$

and the inverse set for each of the multiset permutation given in Eq. (1) is

$$\{[1, 3, 1, 2, 3], [1, 3, 1, 3, 2], [3, 1, 1, 2, 3], [3, 1, 1, 3, 2]\}. \tag{2}$$

Notice that  $\phi$  is the third multiset permutation appearing in Eq. (2), and  $\phi \cdot [1, 2, 3, 1, 3] = I_{\mathcal{M}}$ .

There is always a unique element in the inverse set  $T_{\omega}$ , say  $\gamma$ , such that  $\omega \cdot \gamma = I_{\mathcal{M}}$ . We call  $\gamma$  the inverse of  $\omega$ . Hence,  $[1, 2, 3, 1, 3]$  is the inverse of  $\phi$ .

Clearly,  $\gamma$  can be obtained as follows: Interchange the lines of the two-line representation (functional form) of a multiset permutation and then do a stable sort of the columns in order to bring the top row into non-decreasing order. From now on, unless specified otherwise, when we say inverse of a multiset permutation, we mean the inverse obtained by applying the process explained above. Hence, the inverse of the multiset permutation  $\phi$  is:

$$\phi^{-1} = \begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 3 & 1 & 1 & 2 & 3 \end{pmatrix}^{-1} = \begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 1 & 3 \end{pmatrix}.$$

Note that, for any multiset permutation  $\phi, \phi \cdot \phi^{-1} = I_{\mathcal{M}}$ , but  $\phi^{-1} \cdot \phi \neq I_{\mathcal{M}}$ . Hence, in general for every multiset permutation  $\gamma$  in  $T_{\phi}, \gamma \cdot \phi \neq I_{\mathcal{M}}$ . Hence, multiset permutations do not form a group under the operation  $\cdot$ , in general.

**THEOREM 2.1** *Two multiset permutations are in the inverse sets of each other if and only if the set of preimages of each is equal to the set of images of the other.*

*Proof* Let  $\sigma$  and  $\phi$  be two multiset permutations, and  $T_\sigma$  and  $T_\phi$  be the inverse sets of  $\sigma$  and  $\phi$ , respectively. Assume that  $\sigma \in T_\phi$  and  $\phi \in T_\sigma$ . Let,  $a \in X$  and  $b \in \text{Pim}_\phi(a)$ . Since  $\sigma \in T_\phi$ ,  $\sigma(a) = b$ . Thus,  $b \in \text{Im}_\sigma(a)$ . Now, let  $b \in \text{Im}_\sigma(a)$ , then  $\sigma(a) = b$ . Since  $\sigma \in T_\phi$ ,  $b \in \text{Pim}_\phi(a)$  and therefore,

$$\text{Im}_\sigma(a) = \text{Pim}_\phi(a), \quad \text{for all } a \in X.$$

Similarly,

$$\text{Im}_\phi(a) = \text{Pim}_\sigma(a), \quad \text{for all } a \in X.$$

Conversely, if  $b \in \text{Pim}_\phi(a) = \text{Im}_\sigma(a)$ , for any  $a \in X$ , then  $\sigma(a) = b$ . Therefore  $\sigma \in T_\phi$ . Similarly,  $\phi \in T_\sigma$ . ■

### 3 BLOCK-SORTING OF MULTISET PERMUTATIONS

Given a permutation  $\pi$  of a set  $X = \{1, 2, 3, \dots, n\}$ . Construct a matrix  $N$ , by forming successive rows of  $N$  that are consecutive cyclic left-shifts of the sequence  $\pi$ . By sorting the rows of  $N$  lexically (alphabetically), we may transform  $N$  to a different matrix,  $N'$ . Arnavut and Magliveras [2, 3], have shown that such a matrix forms a cyclic group, and has Eulerian number of generators when  $n$  is a prime number. Otherwise  $N'$  has two generators.

A permutation has distinct elements. Therefore, each row in  $N$  has distinct elements. Arranging rows of  $N$  in lexical order (*i.e.* treat each row as a string and sort them alphabetically) is equivalent to ordering in linear order (*i.e.* sort the rows with respect to the first entry of each row without considering the rest of the symbols). Thus, linear or lexical sorting of rows produces the same permutation matrix,  $N'$ . This may not be true for the general case of permutations, namely for multiset permutations, because, the elements of a multiset permutation may not be distinct. For example, if the multiset permutation is  $\omega = [3, 1, 3, 1, 2]$ , we construct the matrix

$$M = \begin{pmatrix} 3 & 1 & 3 & 1 & 2 \\ 1 & 3 & 1 & 2 & 3 \\ 3 & 1 & 2 & 3 & 1 \\ 1 & 2 & 3 & 1 & 3 \\ 2 & 3 & 1 & 3 & 1 \end{pmatrix},$$

by forming the successive rows of  $M$ , which are consecutive cyclic left-shifts of the sequence  $\omega$ . By sorting the rows of  $M$  lexically we transform it to

$$M' = \begin{pmatrix} 1 & 2 & 3 & 1 & 3 \\ 1 & 3 & 1 & 2 & 3 \\ 2 & 3 & 1 & 3 & 1 \\ 3 & 1 & 2 & 3 & 1 \\ 3 & 1 & 3 & 1 & 2 \end{pmatrix},$$

while by sorting the rows of  $M$  linearly, we transform it to

$$\bar{M} = \begin{pmatrix} 1 & 3 & 1 & 2 & 3 \\ 1 & 2 & 3 & 1 & 3 \\ 2 & 3 & 1 & 3 & 1 \\ 3 & 1 & 3 & 1 & 2 \\ 3 & 1 & 2 & 3 & 1 \end{pmatrix}.$$

Hence, we obtain two distinct matrices, with respect to two different orderings.  $M'$  and  $\bar{M}$  have the same rows, but the ordering of the rows is different. Let  $F'$  be the first,  $S'$  the second and  $L'$  the last column vector of  $M'$ , and let  $\bar{F}$  be the first,  $\bar{S}$  the second and  $\bar{L}$  the last column vector of  $\bar{M}$ . From  $M'$  and  $\bar{M}$ , it is clear that the first columns of both matrices ( $F'$  and  $\bar{F}$ ) are sorted values of  $\omega$  in ascending order, while the other columns are different.

As shown in Figure 2, there is a relationship between  $F'$  and  $L'$  of  $M'$ , which may not hold for other columns of  $M'$ : The  $i$ th  $v$  value in  $F'$ , where  $1 \leq i \leq v$ , occurs as the  $i$ th  $v$  value in  $L'$ . We call this relationship order-relationship and prove it in Lemma 3.1.

A lexically ordered multiset permutation matrix  $M'$  is composed of rows  $(M'_1, M'_2, \dots, M'_n)^T$ , where  $M'_i = [M'_{i,1}, \dots, M'_{i,n}]$  is the  $i$ th row in  $M'$ . Let  $(M'_i)^l$  denote the multiset permutation formed by cyclicly shifting  $M'_i$  to the left once. Clearly,  $(M'_i)^l$  is one of the rows of  $M'$ , since  $M'$  consists of lexically ordered, cyclicly shifted rows of a multiset permutation.

**LEMMA 3.1** *Let  $\trianglelefteq$  denote the lexical ordering between any two rows of  $M'$ . If  $M'_i \trianglelefteq M'_j$  and  $M'_{i,1} = M'_{j,1} = v$ , for some  $1 \leq v \leq m$ , then  $(M'_i)^l \trianglelefteq (M'_j)^l$ .*

*Proof* Let  $M'_i = [M'_{i,1}, \dots, M'_{i,n}]$  and  $M'_j = [M'_{j,1}, \dots, M'_{j,n}]$  be any two rows of  $M'$ , with the condition that  $M'_{i,1} = M'_{j,1} = v$ . Clearly, if  $M'_i = M'_j$ , then  $(M'_i)^l = (M'_j)^l$ . If  $M'_i \trianglelefteq M'_j$  and  $M'_i \neq M'_j$ , then there exists a position  $k$ ,  $1 < k \leq n$ , such that  $M'_{i,k} < M'_{j,k}$  and  $M'_{i,l} = M'_{j,k}$  ( $1 \leq l < k$ ). Cyclic left shifting of  $M'_i$  and  $M'_j$  results in  $(M'_i)^l = [M'_{i,2}, \dots, M'_{i,k}, \dots, M'_{i,n}, M'_{i,1}]$  and  $(M'_j)^l = [M'_{j,2}, \dots, M'_{j,k}, \dots, M'_{j,n}, M'_{j,1}]$ . Now,  $(M'_{i,k-1})^l = M'_{i,k}$  and  $(M'_{j,k-1})^l = M'_{j,k}$ , and hence  $(M'_{i,k-1})^l < (M'_{j,k-1})^l$ . Therefore,  $(M'_i)^l \trianglelefteq (M'_j)^l$ . ■

A consequence of Lemma 3.1 is that the last element of  $(M'_i)^l$ ,  $M'_{i,1}$ , appears earlier than the last element of  $(M'_j)^l$ ,  $M'_{j,1}$ , in  $L'$ .

**THEOREM 3.1** *There is a bijection based on order-relationship between the last column  $L'$  and the first column  $F'$  of  $M'$ .*

*Proof* The first column  $F'$  of  $M'$  is the identity multiset permutation  $I_M$  and consists of  $f_1$  1s,  $f_2$  2s,  $\dots$ ,  $f_m$  ms, where  $n = f_1 + f_2 + \dots + f_m$ . Starting from  $v = 1$ , searching  $L'$  from

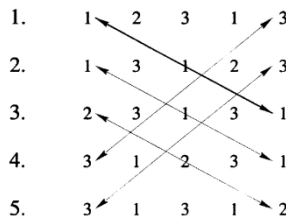


FIGURE 2 Relations of the first and last column elements of  $M'$ .

top to bottom for each occurrence of  $v$  and numbering each occurrence consecutively for each  $v$  ( $1 \leq v \leq m$ ), yields a permutation,  $\pi$ . This permutation defines a bijective map between the first column ( $F'$ ) and the last column ( $L'$ ) of  $M'$ .

It follows from Lemma 3.1 that  $\pi$  has order-relationship: for any given two rows  $M'_i$  and  $M'_j$  of  $M'$ , where  $M'_{i,1} = M'_{j,1} = v$ , if  $M'_i \leq M'_j$ , then  $(M'_i)^l \leq (M'_j)^l$  in  $M'$ . This completes the proof of Theorem 3.1. ■

The permutation obtained from the process described in Theorem 3.1 is called BWT. Permutation matrix  $N'$ , formed by cyclicly shifting a permutation and lexically ordering its rows, forms a cyclic group. This has been shown by Arnavut and Magliveras [2, 3]. Multiset permutations do not form a group, but they possess an interesting property, which is shown in the following theorem.

**THEOREM 3.2** *Let the last column of  $M'$ ,  $L'$ , be denoted by  $\sigma$ . Then  $M'$  is of the form  $(I_M, \sigma^{n-1}, \dots, \sigma^2, \sigma)$ , where  $\sigma^j = \sigma \cdot \sigma^{j-1}$ ,  $j = 2, \dots, n - 1$ .*

*Proof* Let  $M'$  be a  $n \times n$  matrix obtained by lexically sorting the rows formed by cyclicly left shifting  $\omega$ , where  $n = f_1 + f_2 + \dots + f_m$  and  $X = \{1, 2, \dots, m\}$ . Let  $M'_{i,j}$  represent the value that appears in the  $i$ th row and  $j$ th column of  $M'$ . To find  $M'_{i,n-1}$ , shift the  $i$ th row of  $M'$  ( $n - 1$ ) times to the left, so that  $M'_{i,n-1}$  is the last element of the row. Then  $M'_{i,n}$  will be the first element of the row that will be formed. Hence we obtain  $M'_{i,n-1} = \sigma(M'_{i,n})$ , where  $M'_{i,n} = \sigma(x)$ , for some  $x \in X$ . By Lemma 3.1, if  $M'_{i,n} = v$  is the  $k$ th occurrence in  $L'$ , then  $v$  is the  $k$ th occurrence in  $B_v$  of  $F'$ . Thus,

$$M'_{i,n-1} = \sigma(M'_{i,n}) = \sigma \cdot \sigma(x) = \sigma^2(x).$$

Hence, we determine that the  $(n - 1)$ th column of  $M'$  is  $\sigma^2$ . To find  $M'_{i,n-2}$ , left shift  $(n - 2)$  times the  $i$ th row of  $M'$ , so that  $M'_{i,n-2}$  is the last and  $M'_{i,n-1}$  is the first element of the row that will be formed. By the same argument,

$$M'_{i,n-2} = \sigma(M'_{i,n-1}) = \sigma \cdot \sigma^2(x) = \sigma^3(x),$$

where if  $M'_{i,n-1} = v$  is the  $k$ th occurrence in  $L'$ , then  $v$  is the  $k$ th occurrence in  $B_v$  of  $F'$ . Hence, the  $(n - 2)$ nd column of  $M'$  is  $\sigma \cdot \sigma^2 = \sigma^3$ . Continuing this process, we have  $M' = (I_M, \sigma^{n-1}, \dots, \sigma^2, \sigma)$ , where  $\sigma^j = \sigma \cdot \sigma^{j-1}$ ,  $j = 2, \dots, n - 1$ . Notice that,  $\sigma^n = I_M$  and  $\sigma^{n-1} = \sigma^{-1}$  (i.e.  $\sigma \cdot \sigma^{-1} = \sigma \cdot \sigma^{n-1} = I_M$ ). ■

The theory established so far is related to lexical ordering. We now investigate another ordering, which we call linear ordering.

**DEFINITION 3.1** *The linearly ordered matrix  $\bar{M}$  of a multiset permutation  $\omega$  of a multiset  $M$  with size  $n$  from an underlying set  $X = \{1, 2, \dots, m\}$ , is formed as following:*

1. Starting with  $\omega$ , form a  $n \times n$  matrix  $M$  by cyclicly left-shifting the successive rows.
2. Sort the rows of  $M$  in ascending order with respect to the first element of each row in  $M$ .

The rows of  $\bar{M}$  are formed by scanning  $\omega$  from left to right and for each  $v \in X$  determining the positions  $j_1, j_2, \dots, j_{f_v}$ , where  $v$  occurs in  $\omega$ , and for each occurrence of  $v$  cyclicly left shifting  $\omega$  ( $j_k - 1$ ) positions ( $1 \leq k \leq f_v$ ). In essence, since  $\omega \in \mathcal{M} = 1^{f_1} \cdot 2^{f_2} \dots m^{f_m}$ , we first construct  $f_1$  linearly ordered rows obtained from  $\omega$  whose first entries have value 1, then  $f_2$  linearly ordered rows whose first entries have value 2, and so on. Hence,  $\bar{F}$  of the resulting linearly ordered matrix  $\bar{M}$  consists of the sorted values of  $\omega$ , i.e.  $\bar{F} = I_M$ . However,  $I_M$  could

also be obtained by simply indexing the values in  $\omega$ , and then sorting  $\omega$  in ascending order. This operation would result with an identity multiset permutation and a sorting permutation,  $\pi^s$ . The second column is the arrangement of the values in the first column with respect to the permutation  $\gamma$ , where  $\gamma(i) = (\pi^s(i) + 1) \bmod n$  and  $n = |\mathcal{M}|$  as each value in  $\bar{S}$  is the neighbour of the value in  $\bar{F}$ .

An obvious observation about the linearly ordered matrix  $\bar{M}$  is that: For any given two pairs from the first two columns of  $\bar{M}$ ,  $(\bar{F}_i, \bar{S}_i)$  and  $(\bar{F}_k, \bar{S}_k)$ , with the condition,  $\bar{F}_i = \bar{F}_k$ , then  $(\bar{F}_i, \bar{S}_i)$  appears earlier than the pair  $(\bar{F}_k, \bar{S}_k)$  in  $\bar{M}$  (scanning from top to bottom), if and only if the pair appears earlier in  $\omega$  (scanning from left to right). Obviously, the linear ordering induces a particular order on the pairs of the elements,  $(\bar{F}, \bar{S})$ . Thus, we can recover  $\omega$  if we know  $\bar{S}$  and the row index of  $\omega$  in  $\bar{M}$ : In our example,  $\omega = [3, 1, 3, 1, 2]$  occurs at position 5 and the second column is  $\bar{S} = [3, 2, 3, 1, 1]$ . Given  $\bar{S}$  and the row index of  $\omega$  in  $\bar{M}$ , one can construct  $\omega$  by obtaining the frequencies of the elements in  $\bar{S}$  using the count sort [11] in one pass. Then  $\bar{F} = [1, 1, 2, 3, 3] = I_M$ ; hence, the first two columns  $(\bar{F}, \bar{S})$  of  $\bar{M}$  are

$$\begin{pmatrix} 1 & 1 & 2 & 3 & 3 \\ 3 & 2 & 3 & 1 & 1 \end{pmatrix}^T.$$

Accessing to the fifth position of  $(\bar{F}, \bar{S})$ , we acquire the first two elements of  $\omega$ ,  $(\bar{F}_5, \bar{S}_5) = [3, 1]$ . By marking the fifth entry we eliminate it from further consideration (from the two-tuple  $(\bar{F}, \bar{S})$ ). To find the rest of the elements of  $\omega$  we should determine what follows  $\bar{S}_5 = 1$  in  $\omega$ . Thus, we scan the  $\bar{F}$  from top to bottom and determine the first unused (unmarked) entry that has a value 1. This is the third element of  $\omega$ . In our example, this is the first entry where  $\bar{F}_1 = \bar{S}_5 = 1$ . Hence,  $\bar{S}_1 = 3$  should follow  $\bar{S}_5 = 1$  in  $\omega$ . We then eliminate consideration of the first entry from the two-tuple  $(\bar{F}, \bar{S})$ . Since  $\bar{S}_1 = 3$  is determined, the process is repeated to get the fourth element of  $\omega$ . Again, we scan  $\bar{F}$  to determine the position of the first unused entry that contains  $\bar{S}_1 = 3$ . By finding the first unused entry that contains the value 3 at position four in  $\bar{F}$ , we discover that the fourth element of  $\omega$  is  $\bar{S}_4 = 1$ . This entry is eliminated from further consideration. Finally, to find the fifth element of  $\omega$ , we scan to seek the first unused entry that has value 1 in  $\bar{F}$ . Since the first entry that has a value 1 is used previously, the second entry is considered,  $\bar{F}_2 = 1$ . Therefore, the last element of  $\omega$  is  $\bar{S}_2 = 2$ , and  $\omega = [3, 1, 3, 1, 2]$ .

We generalize this process by the following theorem.

**THEOREM 3.3** *Knowledge of the second column  $\bar{S}$  of  $\bar{M}$  and the row index of a multiset permutation  $\omega$  in  $\bar{M}$  allows us to recover  $\omega$ .*

*Proof* Let the location of the row  $\omega$  in  $\bar{M}$  be  $l_1$ . Given the row index  $l_1$  and  $\bar{S}$ , we can construct  $\omega$ . Since  $\bar{S}$  is known,  $\bar{F} = I_M$  is constructed by finding the frequencies of different symbols in  $\bar{S}$ . The first two elements of  $\omega$  are then  $\bar{F}_{l_1}$  and  $\bar{S}_{l_1}$ . Eliminate the entry  $l_1$  of  $\bar{F}$  from further consideration. What follows  $\bar{S}_{l_1}$ ? It should be the element that follows the first value  $\bar{S}_{l_1}$  in  $\bar{F}$  (from top to bottom). Let this value be in position  $l_2$  in  $\bar{F}$ . Hence,  $\bar{S}_{l_1} = \bar{F}_{l_2}$  should be followed by  $\bar{S}_{l_2}$ . Eliminate the entry  $l_2$  of  $\bar{F}$ . What follows  $\bar{S}_{l_2}$ ? It is the element in the second row that follows the first value  $\bar{S}_{l_2}$  in  $\bar{F}$ . Let the index for this value in  $\bar{F}$  be  $l_3$ . Hence,  $\bar{S}_{l_2} = \bar{F}_{l_3}$  is followed by  $\bar{S}_{l_3}$ . Repeating this process, we can form  $\omega$  uniquely. ■

The theory established for lexically ordered matrix shows that the knowledge of  $L'$  and the row index of  $w$  in  $M'$  allows us to recover  $w$ . For a linearly ordered matrix, instead of  $\bar{L}$ , we need to know  $\bar{S}$  to recover  $w$ . From Theorem 3.2, we know that  $M'$  is in the form  $(I_M, (L')^{n-1}, (L')^{n-2}, \dots, L')$ . Similarly, is  $\bar{M}$  in the form  $(I_M, \bar{S}, (\bar{S})^2, \dots, (\bar{S})^{n-1})$ ? Unfortunately the answer is no: In our example, the second column  $\bar{S}$  of the  $\bar{M}$  constructed from

$w = [3, 1, 3, 1, 2]$  does not generate the third or fourth column of  $\bar{M}$ . Note that, if we use any multiset permutation  $(M'_i)^l$  ( $1 \leq i \leq n$ ), we obtain the same lexically ordered matrix  $M'$ . Hence, each  $(M'_i)^l$  ( $1 \leq i \leq n$ ) has the same generator,  $L'$ . However, in the case of linear ordering, for each multiset permutation  $(M_i)^l$  ( $1 \leq i \leq n$ ), we may obtain a different linearly ordered matrix  $\bar{M}$ . Hence, each multiset permutation may have a different generator.

Despite the differences exhibited above, another question that needs to be addressed is: Under which condition(s) would the matrix generated by lexical ordering or the matrix formed by the linear ordering be equivalent for a given multiset permutation?

**LEMMA 3.2** *For any given  $a \in X$ ,  $\text{Im}_{\bar{L}}(a) = \text{Im}_{L'}(a)$  and  $\text{Pim}_{\bar{L}}(a) = \text{Pim}_{L'}(a)$ .*

*Proof* The first column of  $M'$  and  $\bar{M}$  is the identity multiset permutation,  $I_M$ . We know that  $I_M$  gives rise to a partition of blocks  $B_1, B_2, \dots, B_m$ , where for all  $B_i$ ,  $v = i$  and  $f_i = |B_i|$ , for  $1 \leq v \leq m$ . Consider the rows in  $M'$  and  $\bar{M}$  to be partitioned with respect to the blocks of the  $I_M$ . Each block in  $M'$  and  $\bar{M}$  consists of the same rows, but may be in different order. Therefore, for every  $a \in X$ ,  $\text{Im}_{\bar{L}}(a) = \text{Im}_{L'}(a)$  and  $\text{Pim}_{\bar{L}}(a) = \text{Pim}_{L'}(a)$ . ■

**THEOREM 3.4**  *$\bar{S}$  of  $\bar{M}$  and  $L'$  of  $M'$  are in the inverse sets of each other.*

*Proof* By Theorem 2.1, it is sufficient to show that for any given  $a \in X$ ,  $\text{Pim}_{\bar{S}}(a) = \text{Im}_{L'}(a)$  and  $\text{Pim}_{L'}(a) = \text{Im}_{\bar{S}}(a)$ . Fix  $a \in X$ . Let  $b \in \text{Pim}_{\bar{S}}(a)$ . Then, consider the rows that start with  $a$  (i.e.  $B_a$ ). Hence,  $b$  has to be the last element of one of the rows in  $B_a$ . Thus,  $b \in \text{Im}_{\bar{L}}(a)$ , and so by Lemma 3.2,  $b \in \text{Im}_{L'}(a)$ . Conversely, if  $b \in \text{Im}_{L'}(a)$ , then  $a$  has to be the last element of one of the rows which start with  $b$  (i.e.  $B_b$ ). Hence,  $b \in \text{Pim}_{L'}(a)$ . Again, by Lemma 3.2,  $b \in \text{Pim}_{\bar{L}}(a)$ . Thus, for every  $a \in X$ ,

$$\text{Pim}_{\bar{S}}(a) = \text{Im}_{L'}(a).$$

Similarly, for every  $a \in X$ ,

$$\text{Pim}_{L'}(a) = \text{Im}_{\bar{S}}(a). \quad \blacksquare$$

## 4 SUMMARY AND CONCLUSIONS

Recently, the block-sorting schemes have taken great attention in data compression area, since the introduction of the BWT by Burrows and Wheeler [9]. Arnavut and Magliveras [2, 3] have given the theoretical settings of block-sorting schemes for permutations and shown that lexical sorting permutation algorithm may yield optimization choices when the data to be transmitted is a permutation.

In this work, we investigated block-sorting transformations of multiset permutations (data strings). We generalized the BWT based on the multiset permutations, and show that it has an interesting combinatorial formulation. In addition, a new transformation, LOT, is introduced. The relationship between the LOT and BWT is theoretically delineated and shown that the data transformed under LOT or BWT, are in the inverse set of the other transformation.

### Acknowledgement

We would like to thank Prof. Spyros Magliveras, Prof. H. J. Straight, and David Leavitt for helpful suggestions in preparation of this work.

## References

- [1] Arnavut, Z. (1995). Permutations techniques in lossless compression. *Ph. D. dissertation*, University of Nebraska-Lincoln, Lincoln, NE, USA.
- [2] Arnavut, Z. and Magliveras, S. (1997). Block-Sorting and compression. *Proceedings of Data Compression Conference*, Snowbird, Utah, March 25–27, IEEE Computer Society Press, 181–190.
- [3] Arnavut, Z. and Magliveras, S. (1997). Lexical permutation sorting algorithm. *The Computer Journal*, **40**(5), 292–295.
- [4] Arnavut, Z. *et al.* (1998). Block-Sorting transformations. *Proceeding of Data Compression Conference*, Snowbird Utah, March 30 - April 1, IEEE Computer Society Press, 524.
- [5] Arnavut, Z. (2002). Generalization of the burrows-wheeler transformation and inversion ranks. *Proceeding of Data Compression Conference*, Snowbird Utah, April 2–4, IEEE Computer Society Press, 447.
- [6] Arnavut, Z. (2003). Inversion coding. *The Computer Journal*, **47**(1), 46–57.
- [7] Balkenhol, B. and Kurtz, S. (2000). Universal data compression based on the Burrows-Wheeler transformation: theory and practice. *IEEE Transactions on Computers*, **49**(1), 1043–1053.
- [8] Bentley, J. L. *et al.* (1986). A locally adaptive data compression scheme. *Communications of the ACM*, **29**(4), 320–330.
- [9] Burrows, M. and Wheeler, J. D. (1994). A Block-Sorting lossless data compression algorithm. *SRC Research Report 124*, Digital Systems Research Center, Palo Alto, CA, May 1994 (*ftp site: gatekeeper.dec.com, /pub/DEC/SRC/research-reports/SRC-124.ps.Z*)
- [10] Chapin, B. (2000). Switching between two on-line list update algorithms for higher compression of Burrows-Wheeler transformed data. *Proceedings of Data Compression Conference*, Snowbird, Utah, March 28 – 31, IEEE Computer Society Press, 183–192.
- [11] Cormen, H. T. Leiserson, C. E. and Rivest, L. R. (1986). *Introduction to Algorithms*. McGraw-Hill.
- [12] Deorowicz, S. (2000). Improvements to burrows-wheeler compression algorithm. *Software-Practice and Experience*, **30**(13), 1465–1483.
- [13] Deorowicz, S. (2001). Second step algorithms in the burrows-wheeler compression algorithm. *Software-Practice and Experience*, **32**(2), 99–111.
- [14] Elias, P. (1987). Interval and recency rank source coding: two on-line adaptive variable-length schemes. *IEEE Transactions on Information Theory*, **IT-33**(10), 3–10.
- [15] Fenwick, P. (1996). The burrows-wheeler transform for Block-Sorting text compression: principles and improvements. *The Computer Journal*, **39**(9), 731–740.
- [16] Huffman, D. (1952). A method for construction of minimum redundancy codes. *Proceedings of the IRE*, **40**, 1098–1101.
- [17] Isal, R. Y. K. and Moffat, A. (2001). Parsing Strategies for BWT compression, *Proceedings of Data Compression Conference*, Snowbird, Utah, March 27–30, IEEE Computer Society Press, 429–438.
- [18] Schindler, M. (1997). A fast Block-Sorting algorithm for lossless data compression. *Proceedings of Data Compression Conference*, Snowbird, Utah, March 25–27, IEEE Computer Society Press, 469.
- [19] Schindler, M. (1998). *Block-Sorting Algorithm*. Private Communication.
- [20] Seward, J. (1997). The Bzip2 program, vers. 0.1pl2. <http://www.muraroa.demon.co.uk>.
- [21] Wirth, I. A. and Moffat, A. (2001). Can we do without ranks in Burrows Wheeler Transform? *Proceedings of Data Compression Conference*, Snowbird, Utah, March 27–29, IEEE Computer Society Press, 419–428.
- [22] Witten, I. H. Radford, N. M. and Cleary, G. J. (1987). Arithmetic coding for data compression. *Communications of ACM*, **30**(6), 520–540.
- [23] Yokoo, H. (1999). Notes on Block-Sorting data compression. *Electronics and Communications in Japan*, **82**(6), 18–25.