On the combinatorics of suffix arrays

Gregory Kucherov, Lilla Tóthmérész, Stéphane Vialette

On the combinatorics of suffix arrays

Gregory Kucherov[a], Lilla Tóthmérész[a,b,*], Stéphane Vialette[a]

[a] *Université Paris-Est & CNRS, Laboratoire d'Informatique Gaspard Monge, Marne-la-Vallée, France*
[b] *Loránd Eötvös University, Pázmány Péter sétány 1/C, H-1117 Budapest, Hungary*

Abstract

We present a bijective characterization of suffix array permutations obtained from a characterization of Burrows-Wheeler arrays given in [1]. We show that previous characterizations [2, 3, 4], or their analogs, can be obtained in a simple and elegant way using this relationship. To demonstrate the usefulness of our approach, we obtain simpler proofs for some known enumeration results about suffix arrays [3]. Our characterization of suffix arrays is the first based on their relationship with Burrows-Wheeler permutations.

Keywords: combinatorics, permutations, suffix array, Burrows-Wheeler transform

1. Introduction

Suffix array is a very popular data structure in string algorithms, both in theoretical studies and practical applications, that has been designed as a space-efficient alternative to suffix trees [5, 6, 7]. With the discovery of direct linear-time construction algorithms for suffix arrays [8, 9, 10], this data structure received an increasing attention during the last decade. A good deal of work has been devoted to improving the practical efficiency of suffix arrays.

A suffix array for a string of length n is essentially a permutation of $[1, n]$ corresponding to the starting positions of all suffixes sorted lexicographically. Obviously, if the alphabet has a fixed size $k < n$, then for large n, only a proper subset (at most k^n) of all $n!$ permutations are suffix arrays for some word over this alphabet. A main motivation of this paper is to provide a characterization for suffix array permutations for bounded-size alphabets.

Our results take advantage of a very close relationship between suffix arrays and the *Burrows-Wheeler transform* [11]. The Burrows-Wheeler transform of a string is a permutation of string letters which allows the string to be effectively reconstructed. Among other applications, it is the basis for many *compact text indexes* that have been intensively studied and used in practical applications (see e.g. [12] and references therein).

Crochemore et al. [1] pointed out a very nice characterization of *Burrows-Wheeler arrays*, which are close relatives of suffix arrays. This characterization, attributed to Gessel and Reutenauer [13], uses the key notion of *linking permutation* which is similar to (but different from) the notion of Ψ-function studied for suffix arrays [14]. We show

*Corresponding author

that the approach of [1] can be successfully applied to obtain characterization results for suffix arrays, using a relation between orderings of suffixes and cyclic shifts.

As far as related works are concerned, He at al. [4] provided a characterization of suffix arrays for the case of binary alphabet ($k = 2$) and an assumption that the terminal sentinel symbol is ranked between the two main symbols in the alphabet ordering. We show that the characterization of [4] easily follows from the characterization that we propose in this paper.

In [3], Schürmann and Stoye prove several counting results for suffix arrays and corresponding strings. They use a characterization of suffix array permutations through Ψ-functions, that they call R_+-arrays, which are mappings from $[1, n]$ to $[0, n]$. A crucial parameter in countings is the number of *descents* in R_+-arrays, directly related to the minimal alphabet size on which the corresponding suffix array can be realized (see also [2]). Compared to our approach, an important difference is that the set of R_+-arrays is not characterized, while the set of linking permutations admit a neat combinatorial characterization as permutations with only one orbit. This allows us to provide a bijection between suffix arrays and a certain well-defined class of permutations. To demonstrate the usefulness of our approach, we obtain much simpler proofs of counting theorems from [3]. Such proofs can be useful in the study of compressibility of suffix arrays (see [3]), which is an important practical issue. Our characterization of suffix arrays is the first based on their relationship with Burrows-Wheeler permutations.

2. Preliminaries

In what follows, $\Sigma = \{a_1, a_2, \ldots, a_k\}$ is an ordered alphabet of size k, where $a_1 < a_2 < \cdots < a_k$. The set of all permutations $\pi = \pi(1)\,\pi(2)\,\ldots\,\pi(n)$ of length n is denoted by $\mathbf{S}_n$, and id denotes the identity permutation $1\,2\,\ldots\,n$. The composition of permutations σ and π is denoted $\pi\sigma$, *i.e.*, $(\pi\sigma)(i) = \pi(\sigma(i))$. Throughout the paper, we assume that the addition (subtraction) of a constant value to a permutation value verifies the identities $n + 1 \equiv 1$, $1 - 1 \equiv n$. In other words, $\pi(i) + k = ((\pi(i) - 1 + k) \mod n) + 1$. For a permutation $\pi \in \mathbf{S}_n$ and $k \in [1, n]$, we define $\pi + k$ to be the permutation defined by $(\pi + k)(i) = \pi(i) + k$. Note that $\pi + k = (\mathrm{id} + k)\pi$. The permutation $(\pi - k)$ is defined similarly.

Definition 1 (suffix array)**.** *Given a word $w = w_1 w_2 \ldots w_n$ on alphabet Σ, its* suffix array *is a permutation π such that $\pi(i) = j$ iff the suffix $w_j \ldots w_n$ is the i^{th} in the lexicographic ordering of all suffixes of w.*

For example, the suffix array of *bbaba* is 5 3 4 2 1.

Definition 2 (primitive word)**.** *A word $u \in \Sigma^+$ is called* primitive *if it is not a proper power of another word,* i.e.*, $u = v^n$, $v \in \Sigma^+$, implies $n = 1$.*

Primitive words are exactly those words whose cyclic shifts are all distinct. Therefore, for a primitive word, we can consider the permutation defined by the lexicographic ordering of its cyclic shifts. We call this permutation the *Burrows-Wheeler array* because of its direct relation to the Burrows-Wheeler transform [11].

Definition 3 (BW-array)**.** *Given a primitive word $w = w_1 \ldots w_n$, its* Burrows-Wheeler array *(hereafter* BW-array*) is a permutation π such that $\pi(i) = j$ iff the word $w_j \ldots w_n w_1 \ldots w_{j-1}$ is the i^{th} in the lexicographic ordering of all cyclic shifts of w.*

For example, the BW-array of *bbaba* is 3 5 2 4 1.

We write $\mathbf{S}_n^c$ for the set of all permutations of $\mathbf{S}_n$ with one orbit. The following notion has proved to be very helpful for characterizing BW-arrays [13, 1]. It is related to Ψ-functions [14] or R_+-arrays [3] defined on suffix arrays, but defines a mapping on permutations.

Definition 4 (linking permutation, linking mapping)**.** *Let $\pi \in \mathbf{S}_n$. The* linking permutation *of π is the permutation $\varphi = \pi^{-1}(\pi + 1) \in \mathbf{S}_n^c$. The mapping $\pi \mapsto \varphi$ is called the* linking mapping*, and is denoted by Φ.*

As an example, the linking permutation of 5 2 4 1 3 is 4 5 1 2 3. Observe that $\varphi \in \mathbf{S}_n^c$ follows from $\varphi(\pi^{-1}(i)) = \pi^{-1}(i+1)$, where $n+1 \equiv 1$. The linking permutation of a BW-array gives the ranks of the consecutive shifts in the lexicographic order. Furthermore, note that if $\pi(1)$ and $\varphi = \Phi(\pi)$ are known, then one can reconstruct π by iterating $\pi(\varphi(i)) = \pi(i) + 1$ starting with $i = 1$.

Definition 5 (permutation descent)**.** *Let $\pi \in \mathbf{S}_n$. We say that $i \in [1, n-1]$ is a* descent *of π if and only if $\pi(i) > \pi(i+1)$. The set of all descents of π is denoted $\mathcal{D}(\pi)$.*

The number of descents of the linking permutation is related to the minimal alphabet size with which a given BW-array is realizable: $w_{\pi(i)} \dots w_{\pi(i)-1}$ is lexicographically smaller than $w_{\pi(i+1)} \dots w_{\pi(i+1)-1}$, so if $w_{\pi(i)+1} \dots w_{\pi(i)}$ is lexicographically larger than $w_{\pi(i+1)+1} \dots w_{\pi(i+1)}$ (equivalently, $\phi(i) = \pi^{-1}(\pi(i)+1) > \pi^{-1}(\pi(i+1)+1) = \phi(i+1)$), then $w_{\pi(i)} < w_{\pi(i+1)}$.

The following equivalence relation corresponds to the fact, that if the BW-array of w is π, then the BW-array of w cyclically shifted by k characters is $\pi - k$.

Definition 6. *For two permutations π and σ of $\mathbf{S}_n$, define $\pi \sim \sigma$ if and only if there exists $k \in [1, n]$ such that $\sigma = \pi + k$.*

It is easily seen that $\sim$ is an equivalence relation on $\mathbf{S}_n$. The following proposition shows that the linking mapping depends only on these equivalence classes.

Proposition 1. *Φ is well-defined on $\mathbf{S}_n/\!\sim$ and bijective from $\mathbf{S}_n/\!\sim$ to $\mathbf{S}_n^c$.*

Proof. We first show that $\Phi(\pi) = \Phi(\sigma)$ if $\pi \sim \sigma$. Let $\sigma = \pi + k$. Then, $\Phi(\pi) = \pi^{-1}(\pi + 1) = \pi^{-1}(((\pi + k) + 1) - k) = \pi^{-1}(\mathrm{id} - k)((\pi + k) + 1)$. Observe now that $(\pi + k)^{-1} = \pi^{-1}(\mathrm{id} - k)$. Therefore, $\Phi(\pi) = (\pi + k)^{-1}((\pi + k) + 1) = \Phi(\pi + k) = \Phi(\sigma)$. This shows that Φ is well-defined on $\mathbf{S}_n/\!\sim$.

Second, we can uniquely determine π from $\Phi(\pi)$ and $\pi(1)$, and hence Φ is injective on $\mathbf{S}_n/\!\sim$. The number of $\sim$-equivalence classes is $(n-1)! = |\mathbf{S}_n^c|$. Therefore, the mapping is a bijection. □

The following theorem from [1] provides a nice characterization of BW-arrays through the linking mapping. It will play a central role in our study.

Theorem 1 ([1])**.** *Let $r_i \geq 0$, $1 \leq i \leq k$, be integers such that $\sum_{i=1}^{k} r_i = n$. A permutation $\pi \in \mathbf{S}_n$ is the BW-array of a primitive word $w \in \Sigma^n$ with r_i occurrences of letter a_i, $1 \leq i \leq k$, if and only if $\mathcal{D}(\Phi(\pi)) \subseteq \{r_1, r_1 + r_2, \dots, r_1 + \dots + r_{k-1}\}$. Moreover, in this case π is the BW-array of exactly one such word.*

3. Characterization of suffix arrays

In this section, we state our characterization theorems for suffix arrays: Theorems 4, 5 and 6. We use a reduction of suffix sorting to cyclic shift sorting by appending a sentinel symbol to the end of the word, and thereby reduce the characterization of suffix arrays to the characterization of BW-arrays.

Consider a symbol $\sharp \notin \Sigma$, and the alphabet $\Sigma' = \{\sharp, a_1, a_2, \ldots, a_k\}$ with $\sharp < a_1 < a_2 \cdots < a_k$. We will examine the suffix arrays of words $w\sharp$ for $w \in \Sigma^n$. The following proposition is obvious.

Proposition 2. *There is a one-to-one correspondence between the suffix arrays of $w \in \Sigma^n$ and the suffix arrays of $w' \in \Sigma^n\sharp$. If $\sigma \in \mathbf{S}_n$ is the suffix array of w, then $\pi \in \mathbf{S}_{n+1}$ is the suffix array of $w\sharp$ if and only if $\pi = (n+1)\,\sigma(1)\,\sigma(2)\ldots\sigma(n)$.*

The following proposition shows, that for words in $\Sigma^n\sharp$, cyclic shift sorting is equivalent to suffix sorting. Note that this property remains true even if we do not assume that $\sharp$ is the smallest element in the ordering of $\Sigma \cup \{\sharp\}$. This will be important later.

Proposition 3. *Let $w' = w\sharp$, where $w \in \Sigma^*$, $\sharp \notin \Sigma$ and the ordering of $\Sigma \cup \{\sharp\}$ is arbitrary. Then the order of the cyclic shifts of $w\sharp$ coincides with the order of the suffixes of $w\sharp$.*

Proof. First observe that w' is a primitive word. If $w' = u^k$ for some word u and a $k > 1$, then k would divide the number of occurrences of $\sharp$. Therefore w' is primitive and the order of its cyclic shifts is well-defined.

As w' has only one occurrence of $\sharp$, then in comparing two different cyclic shifts we necessarily compare $\sharp$ with some other character. This means that the lexicographic order of two cyclic shifts is decided no later than at the position of the first $\sharp$. Therefore if we leave out the characters after the $\sharp$ in both shifts, we get the same ordering. □

Theorems 2 and 3 below characterize the permutations that are suffix arrays for some word $w\sharp$ with $w \in \Sigma^n$. Closely related results have been obtained earlier in [2, 3] . Our contribution is the direct application of the linking permutation from the BW-framework, which makes it possible to deduce these theorems by a very short argument.

Theorem 2. *Let $r_i \geq 0$, $1 \leq i \leq k$, be integers such that $\sum_{i=1}^{k} r_i = n$. A permutation $\pi \in \mathbf{S}_{n+1}$ is the suffix array of a word $w\sharp \in \Sigma^n\sharp$ with r_i occurrences of the letter a_i, $1 \leq i \leq k$, if and only if $\mathcal{D}(\Phi(\pi)) \subseteq \{1, 1+r_1, 1+r_1+r_2, \ldots, 1+r_1+\cdots+r_{k-1}\}$ and $\pi(1) = n+1$. Moreover, in this case, π is the suffix array of exactly one such word.*

Proof. According to Theorem 1, $\pi \in \mathbf{S}_{n+1}$ is the BW-array of a primitive word $w \in (\Sigma \cup \sharp)^{n+1}$ with r_i occurrences of letter a_i, $1 \leq i \leq k$, and one occurrence of symbol $\sharp$, if and only if the first condition is satisfied, and in this case there is only one such primitive word. Here the primitivity is immediate, since we have only one occurrence of $\sharp$. Since $\sharp$ is the smallest letter, condition $\pi(1) = n+1$ is necessary and sufficient for $\sharp$ to be the last letter. Finally the BW-array coincides with the suffix array on the class of words type $w\sharp$, by Proposition 3. □

As an example, consider any word w over alphabet $\{a, b, c\}$ $(a < b < c)$ with 3 occurrences of a, 2 occurrences of b and 3 occurrences of c. By Theorem 2, the suffix array

π of word $w\sharp$ must verify $\pi(1) = 9$ and $\mathcal{D}(\Phi(\pi)) \subseteq \{1, 4, 6\}$. Consider $\pi = 9\,8\,2\,3\,1\,6\,7\,5\,4$. We have $\Phi(\pi) = 5\,1\,4\,9\,3\,7\,2\,6\,8$ and $\mathcal{D}(\Phi(\pi)) = \{1, 4, 6\}$. Since $\pi[1] = 9$, both conditions are verified, therefore π is the suffix array of a unique such word $w\sharp$. Here $w = baaccbca$.

From Theorem 2, we can immediately deduce the following theorem:

Theorem 3. *A permutation $\pi \in \mathbf{S}_{n+1}$ is the suffix array of a word $w\sharp \in \Sigma^n\sharp$ if and only if (i) $|\,\mathcal{D}(\Phi(\pi)) \setminus \{1\}| \leq k-1$, and (ii) $\pi(1) = n+1$.*

Proof. Let $\pi \in \mathbf{S}_{n+1}$ be the suffix array of a word $w\sharp$ for $w \in \Sigma^n$. Assume w has $r_i \geq 0$ occurrences of letter a_i for each $i \in [1, k]$. Then conditions (i) and (ii) follow immediately from Theorem 2. Conversely, let $\mathcal{D}(\Phi(\pi)) \setminus \{1\} = \{d_1, d_2, \ldots, d_\ell\}$ for $\ell \leq k-1$. Then for $r_1 = d_1 - 1$, $r_2 = d_2 - d_1$, $\ldots$, $r_\ell = d_\ell - d_{\ell-1}$, $r_{\ell+1} = \ldots = r_{k-1} = 0$, we have $\mathcal{D}(\Phi(\pi)) \subseteq \{1, 1+r_1, 1+r_1+r_2, \ldots, 1+r_1+\cdots+r_{k-1}\}$. $\pi(1) = n+1$ is also satisfied. Then, by Theorem 2, there is a word $w\sharp$ with the corresponding numbers of letter occurrences that has π as its suffix array. □

Now we provide a characterization of suffix arrays for the case where we do not assume a sentinel symbol at the end of the word. Proposition 2 combined with Theorem 2 and Theorem 3 respectively implies the following results.

Theorem 4. *Let $r_i \geq 0$, $1 \leq i \leq k$, be integers such that $\sum_{i=1}^{k} r_i = n$. A permutation $\pi \in \mathbf{S}_n$ is the suffix array of a word $w \in \Sigma^n$ with r_i occurrences of the letter a_i, $1 \leq i \leq k$, if and only if, for $\pi' = (n+1)\,\pi(1)\ldots\pi(n)$, $\mathcal{D}(\Phi(\pi')) \subseteq \{1, 1+r_1, 1+r_1+r_2, \ldots, 1+r_1+\cdots+r_{k-1}\}$. Moreover, in this case π is the suffix array of exactly one such word.*

Consider, for example, words w composed of 5 a's and 3 b's. Consider $\pi = 8\,2\,3\,6\,4\,7\,1\,5$. To verify if π is the suffix array of some word w, we define $\pi' = 9\,8\,2\,3\,6\,4\,7\,1\,5$ and compute $\Phi(\pi') = 8\,1\,4\,6\,7\,9\,2\,3\,5$, $\mathcal{D}(\Phi(\pi')) = \{1, 6\}$. This proves that π is the suffix array of a single word w with 5 a's and 3 b's. Here $w = baaababa$. A similar check for $\pi = 8\,2\,3\,4\,7\,1\,6\,5$ leads to $\mathcal{D}(\Phi(\pi')) = \{1, 5\}$ which means that a word $w \in \{a, b\}$ which has π as the suffix array must have 4 a's and 4 b's.

Theorem 5. *A permutation $\pi \in \mathbf{S}_n$ is the suffix array of some word $w \in \Sigma^n$ if and only if, for $\pi' = (n+1)\,\pi(1)\ldots\pi(n)$, we have $|\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}| \leq k-1$.*

Finally, we give our main result establishing a bijection between the suffix arrays and a certain set of permutations. As will be illustrated later, this bijection provides an efficient tool for reasoning about suffix arrays.

Theorem 6. *For a permutation $\pi \in \mathbf{S}_n$, let $\pi' = (n+1)\,\pi(1)\ldots\pi(n)$. The mapping $\pi \mapsto \Phi(\pi')$ is a bijection between the suffix arrays of words $w \in \Sigma^n$ and the permutations $\varphi \in \mathbf{S}^c_{n+1}$ with $|\,\mathcal{D}(\varphi) \setminus \{1\}| \leq k-1$. Moreover, given such $\varphi \in \mathbf{S}^c_{n+1}$, we can easily compute the corresponding suffix array π as follows: $\pi^{-1}(i) = \varphi^i(1) - 1$ for each $i \in [1, n]$.*

Proof. Let us denote the $\sim$ equivalence class of a $\sigma \in \mathbf{S}_{n+1}$ by $[\sigma]$. The mapping $f : \mathbf{S}_n \to \mathbf{S}_{n+1}/\sim$ defined by $f(\pi) = [\pi']$ is a bijection between $\mathbf{S}_n$ and $\mathbf{S}_{n+1}/\sim$. Φ is bijective from $\mathbf{S}_{n+1}/\sim$ to $\mathbf{S}^c_{n+1}$, and hence $\pi \mapsto \Phi(\pi')$ is a bijection from $\mathbf{S}_n$ to $\mathbf{S}^c_{n+1}$. According to Theorem 5, a permutation $\pi \in \mathbf{S}_n$ is a suffix array of some word in Σ^n if and only if, for its image $\Phi(\pi')$, $|\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}| \leq k-1$. Then it follows that the restriction of the mapping to the set of suffix permutations gives a bijection into the set of permutations $\varphi \in \mathbf{S}^c_{n+1}$ with $|\,\mathcal{D}(\varphi) \setminus \{1\}| \leq k-1$.

As for the computation of the inverse mapping, we know that $(\pi')^{-1}(n+1) = 1$ and that $\Phi(\pi')((\pi')^{-1}(i)) = (\pi')^{-1}(i+1)$. Therefore, if $\varphi = \Phi(\pi')$, then $\pi^{-1}(i) = (\pi')^{-1}(i) - 1 = \varphi^i(1) - 1$ for all $i \in [1, n]$. □

Consider $\pi = 8\,2\,3\,4\,7\,1\,6\,5$ from the previous example. We have $\phi = \Phi(\pi') = 7\,1\,4\,5\,9\,2\,3\,6\,8$. π can be recovered from ϕ by applying $\pi^{-1}(i) = \varphi^i(1) - 1$ for all i.

4. Relation to the characterization of He et al

He et al. [4] proposed a characterization of suffix arrays for a binary alphabet $\Sigma = \{a, b\}$ in the special case where the sentinel character $\sharp$ is ranked between the characters of Σ, *i.e.*, $a < \sharp < b$. In this case, the lexicographic order of suffixes of w can be different from the lexicographic order of the corresponding suffixes of $w\sharp$, therefore this definition gives a slightly different suffix array notion.

In this section, we elucidate how the characterization of [4] is related to our characterizations given in Section 3. In particular, we show that our approach yields a simpler characterization that implies the result of [4]. Before describing the characterization of [4], we show that Theorem 1 allows us to obtain a characterization of suffix arrays for this kind of alphabet ordering as well, similarly to the usual ordering of the previous section.

Theorem 7. *A permutation $\pi \in \mathbf{S}_{n+1}$ is the suffix array of a word $w\sharp$ with $w \in \{a, b\}^n$ and $a < \sharp < b$ if and only if $\mathcal{D}(\Phi(\pi)) \subseteq \{\pi^{-1}(n+1) - 1, \pi^{-1}(n+1)\}$.*

Proof. Let $\pi \in \mathbf{S}_{n+1}$. By Proposition 3, π is the suffix array of $w\sharp$ if and only if it is the BW-arrays of $w\sharp$. Therefore it is enough to prove the theorem for BW arrays instead of suffix arrays.

We first show the 'only if' part. Observe that if π is the BW array of $w\sharp$, then $w_{\pi(i)} = a$ for $i < \pi^{-1}(n+1)$, and $w_{\pi(i)} = b$ for $i > \pi^{-1}(n+1)$. Therefore w has $\pi^{-1}(n+1) - 1$ occurrences of a, 1 occurrence of $\sharp$ and $n + 1 - \pi^{-1}(n+1)$ occurrences of b. By Theorem 1, we immediately obtain $\mathcal{D}(\Phi(\pi)) \subseteq \{\pi^{-1}(n+1) - 1, \pi^{-1}(n+1)\}$.

We now prove the 'if' part. Suppose that $\mathcal{D}(\Phi(\pi)) \subseteq \{\pi^{-1}(n+1) - 1, \pi^{-1}(n+1)\}$. From Theorem 1 there exists exactly one word $w' \in \{a, \sharp, b\}^{n+1}$ that has $\pi^{-1}(n+1) - 1$ occurrences of a, 1 occurrence of $\sharp$ and $n + 1 - \pi^{-1}(n+1)$ occurrences of b and which has π as BW array. From $w'_{\pi(1)} \leq w'_{\pi(2)} \leq \cdots \leq w'_{\pi(n+1)}$, this word is the following: $w'_{\pi(i)} = a$ for $i < \pi^{-1}(n+1)$, $w'_{\pi(\pi^{-1}(n+1))} = w'_{n+1} = \sharp$, and $w'_{\pi(i)} = b$ for $i > \pi^{-1}(n+1)$. We can see, that $w' = w\sharp$ where $w \in \{a, b\}^n$, therefore we have the sufficiency of the condition. □

Now, we repeat the characterization given by He et al. [4]. We need some additional definitions.

Definition 7 (Ascending-to-max [4])**.** *A permutation $\pi \in \mathbf{S}_{n+1}$ is* ascending-to-max *if and only if, for every $i \in [1, n-1]$, we have*

(a) if $\pi^{-1}(i) < \pi^{-1}(n+1)$, $\pi^{-1}(i+1) < \pi^{-1}(n+1)$, then $\pi^{-1}(i) < \pi^{-1}(i+1)$, and
(b) if $\pi^{-1}(i) > \pi^{-1}(n+1)$, $\pi^{-1}(i+1) > \pi^{-1}(n+1)$, then $\pi^{-1}(i) > \pi^{-1}(i+1)$.

Definition 8 (Non-nesting [4]). *A permutation $\pi \in \mathbf{S}_{n+1}$ is* non-nesting *if and only if, for each $i, j \in [1, n]$ such that $\pi^{-1}(i) < \pi^{-1}(j)$, if*

(a) $\pi^{-1}(i) < \pi^{-1}(i+1)$ *and* $\pi^{-1}(j) < \pi^{-1}(j+1)$, *or*
(b) $\pi^{-1}(i) > \pi^{-1}(i+1)$ *and* $\pi^{-1}(j) > \pi^{-1}(j+1)$,

then $\pi^{-1}(i+1) < \pi^{-1}(j+1)$.

The characterization of [4] is as follows.

Theorem 8 ([4]). *A permutation $\pi \in \mathbf{S}_{n+1}$ is the suffix array of a word $w\sharp$ with $w \in \{a, b\}^n$ and $a < \sharp < b$ if and only if it is both ascending-to-max and non-nesting.*

We now show that the condition of Theorem 8 is equivalent to that of Theorem 7. Let $\varphi = \Phi(\pi)$. We have $\varphi(\pi^{-1}(i)) = \pi^{-1}(i+1)$. Therefore, the ascending-to-max property reduces to $i < \varphi(i)$ for $i \in [1, \pi^{-1}(n+1) - 1]$, and $i > \varphi(i)$ for $i \in [\pi^{-1}(n+1) + 1, n+1]$. As for the non-nesting property, we have the following: for $i, j \in [1, n] \setminus \{\pi^{-1}(n+1)\}$, if $i < \varphi(i)$ and $j < \varphi(j)$, or $i > \varphi(i)$ and $j > \varphi(j)$, then $i < j$ implies $\varphi(i) < \varphi(j)$.

We show that the two conditions together are equivalent to the condition of Theorem 7. The two conditions together trivially imply the condition of Theorem 7. Conversely, suppose that $\mathcal{D}(\Phi(\pi)) \subseteq \{\pi^{-1}(n+1) - 1, \pi^{-1}(n+1)\}$. φ has one orbit, and hence $\varphi(1) > 1$. If for some $j \in [1, n]$ we have $\varphi(j) > j$ and $\varphi(j+1) < j+1$, then j is a descent. Hence for $i \in [1, \pi^{-1}(n+1) - 1]$, $i < \varphi(i)$. Similarly,for $i \in [\pi^{-1}(n+1) + 1, n+1]$, $i > \varphi(i)$. From $\mathcal{D}(\Phi(\pi)) \subseteq \{\pi^{-1}(n+1) - 1, \pi^{-1}(n+1)\}$ it follows that φ is monotone on $[1, \pi^{-1}(n+1) - 1]$ and on $[\pi^{-1}(n+1) + 1, n+1]$, and hence the non-nesting property is also satisfied.

5. Enumerations

Our characterization theorems from Section 3 can also be used to count objects related to suffix arrays. Schürmann and Stoye [3] obtained some counting results using "direct" combinatorial considerations. Here we give shorter proofs of these results, based on bijections provided by Theorems 4-6 from Section 3.

The following enumerations have been studied in [3].

(i) For a permutation $\pi \in \mathbf{S}_n$, count the number of words of length n over an alphabet of size k that have π as their suffix array,
(ii) For a permutation $\pi \in \mathbf{S}_n$, count the number of words of length n over an alphabet of size k, that have at least one occurrence of each letter and have π as their suffix array,
(iii) Count the number of permutations $\pi \in \mathbf{S}_n$ that are suffix arrays of some word over an alphabet of size k.

We start with question (i).

Theorem 9 ([3]). *For a permutation $\pi \in \mathbf{S}_n$, let $\pi' = (n+1)\,\pi(1) \ldots \pi(n)$. The number of words of length n over an alphabet of size k having π as their suffix array is*

$$\binom{n + k - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|}{k - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|}.$$

Proof. Theorem 4 states that if $r_i \geq 0$ for $i = 1 \dots k$, $\sum_{i=1}^{k} r_i = n$, and

$$\mathcal{D}(\Phi(\pi')) \subseteq \{1, 1 + r_1, 1 + r_1 + r_2, \dots, 1 + r_1 + \dots + r_{k-1}\}, \tag{1}$$

then there is exactly one word w with r_i occurrences of a_i that has π as its suffix array. Therefore, we need to count the number of tuples $(r_1, \dots, r_k)$ (Parikh vectors) that satisfy (1) given a permutation $\pi' = (n+1)\,\pi(1) \dots \pi(n)$. We represent a tuple $(r_1, \dots, r_k)$, $\sum_{i=1}^{k} r_i = n$, as a sequence of n dots divided into k (possibly empty) groups separated by $k-1$ separators:

$$(r_1, \dots, r_k), \sum r_i = n \quad \leftrightarrow \quad \underbrace{\circ \cdots \circ}_{r_1} | \underbrace{\circ \cdots \circ}_{r_2} | \dots | \underbrace{\circ \cdots \circ}_{r_k} \tag{2}$$

Clearly, this representation is a bijection.

If $i > 1$ is a descent of $\Phi(\pi')$, there must be a separator between the $(i-1)$-th and the i-th dots. This defines the placement of $|\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|$ separators. The remaining $(k - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|)$ separators can interleave the n dots arbitrarily. This can be done in

$$\binom{n + k - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|}{k - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|} = \binom{n + k - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|}{n}$$

ways. The result follows. □

Note that if $k - 1 < |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|$, there is no word on an alphabet of size k which has π as its suffix array. This is confirmed by Proposition 9, as $\binom{m}{n} = 0$ for $m < n$. The following proposition from [3] answers question (ii).

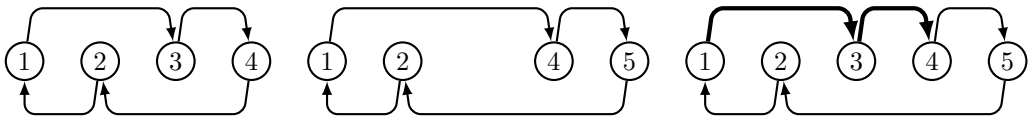
Theorem 10 ([3])**.** *For a permutation $\pi \in \mathbf{S}_n$, let $\pi' = (n+1)\,\pi(1) \dots \pi(n)$. The number of words of length n over an alphabet of size k that have at least one occurrence of each of the k letters and have π as their suffix array is*

$$\binom{n - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|}{k - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|}.$$

Proof. We modify the proof of Proposition 9 to insure that that each letter occurs at least once. We cannot have two adjacent separators, and we cannot start or end with a separator. We then have to distribute $k-1$ separators among $n-1$ possible places between the circles. Like in the proof of Proposition 9, the place of $|\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|$ separators is determined by π', and the remaining $(k - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|)$ separators are distributed among $(n - 1 - |\,\mathcal{D}(\Phi(\pi')) \setminus \{1\}|)$ remaining places. This yields the count of the Theorem. □

Finally, we give a proof for question (iii), based on the results of Section 3. Let $\left\langle {n \atop d} \right\rangle$ denote the Eulerian number, i.e. the number of permutations of $[1, n]$ with exactly d descents.

Theorem 11 ([3])**.** *The number of permutations $\pi \in \mathbf{S}_n$ that are suffix arrays of a word $w \in \Sigma^n$ with $|\Sigma| = k$ is* $\displaystyle\sum_{d=0}^{k-1} \left\langle {n \atop d} \right\rangle$.



φ $\qquad$ $\mathrm{Aug}_s \circ \varphi \circ \mathrm{Aug}_s^{-1}$ $\qquad$ $T_s(\varphi)$

Figure 1: Illustration of $T_s(\varphi)$ with $\varphi = 3142$ and $s = 3$. Informally, $\mathrm{Aug}_s \circ \varphi \circ \mathrm{Aug}_s^{-1}$ "increments by one" all nodes $s, \ldots, n$. Then $T_s(\varphi)$ "splits" the mapping $(1, \varphi(1))$ into $(1, s)$ and $(s, \varphi(1))$.

Proof. According to Theorem 6, there is a bijection between the suffix arrays of words $w \in \Sigma^n$ and the permutations $\varphi \in \mathbf{S}^c_{n+1}$ such that $|\,\mathcal{D}(\varphi) \setminus \{1\}| \leq k-1$. We then have to count the number of such permutations. Let $P(n, d)$ denote the number of permutations $\varphi \in \mathbf{S}^c_{n+1}$ with $|\,\mathcal{D}(\varphi) \setminus \{1\}| = d$. To prove the theorem, we show that $P(n, d)$ is equal to the Eulerian number $\left\langle {n \atop d} \right\rangle$.

The proof is by induction on n. Trivially, $P(1, 0) = 1 = \left\langle {1 \atop 0} \right\rangle$ (the only good permutation is $\pi = 2\,1$), and $P(1, d) = 0 = \left\langle {1 \atop d} \right\rangle$ when $d \geq 1$. We now show that $P(n, d) = (d+1)P(n-1, d) + (n-d)P(n-1, d-1)$, thereby proving that $P(n, d) = \left\langle {n \atop d} \right\rangle$. For the inductive step, we describe a generative procedure for the considered permutations. Consider $\varphi \in \mathbf{S}^c_n$ and let $s \in [2, n+1]$. Consider the mapping $\mathrm{Aug}_s : [1, n] \to [1, n+1]$ defined by

$$\mathrm{Aug}_s(i) = \begin{cases} i & \text{if } i < s, \\ i+1 & \text{if } i \geq s. \end{cases}$$

Observe that $\mathrm{Aug}_s \circ \varphi \circ \mathrm{Aug}_s^{-1}$ is bijective on the set $\{1, \ldots, s-1, s+1, \ldots, n\}$ and has one orbit. Now consider the transform $T_s : \mathbf{S}^c_n \to \mathbf{S}^c_{n+1}$ defined by

$$T_s(\varphi)(i) = \begin{cases} \mathrm{Aug}_s \circ \varphi \circ \mathrm{Aug}_s^{-1}(i) & \text{if } i \in [1, n+1] \setminus \{1, s\}, \\ s & \text{if } i = 1, \\ \mathrm{Aug}_s \circ \varphi \circ \mathrm{Aug}_s^{-1}(1) = \varphi(1) & \text{if } i = s. \end{cases}$$

It is straightforward to check that $T_s(\varphi) \in \mathbf{S}^c_{n+1}$, *i.e.*, $T_s(\varphi) \in \mathbf{S}_{n+1}$ and it has one orbit. The construction is illustrated in Figure 1.

Furthermore, if $r, s \in [2, n+1]$ and $\varphi, \psi \in \mathbf{S}^c_n$, where $r \neq s$ or $\varphi \neq \psi$, then $T_s(\varphi) \neq T_r(\psi)$. As there are $(n-1)!$ permutations in $\mathbf{S}^c_n$, we get $n \cdot (n-1)! = n!$ different permutations $T_s(\varphi)$ for $s \in [2, n+1]$ and $\varphi \in \mathbf{S}^c_n$. Therefore, $\mathbf{S}^c_{n+1} = \{T_s(\varphi) \;:\; s \in [2, n+1],\ \varphi \in \mathbf{S}^c_n\}$.

Now, we examine how the transform $T_s(\varphi)$ affects the number of descents of φ. For $i \in [2, n-1] \setminus \{s-1\}$, $T_s(\varphi)(\mathrm{Aug}_s(i)) = \mathrm{Aug}_s(\varphi(i))$ and $T_s(\varphi)(\mathrm{Aug}_s(i)+1) = T_s(\varphi)(\mathrm{Aug}_s(i+1)) = \mathrm{Aug}_s(\varphi(i+1))$. Therefore for $i \in [2, n-1] \setminus \{s-1\}$

$$\mathrm{Aug}_s(i) \in \mathcal{D}(T_s(\varphi)) \;\Leftrightarrow\; i \in \mathcal{D}(Aug_s(\varphi)) \;\Leftrightarrow\; i \in \mathcal{D}(\varphi),$$

where the second equivalence follows from the monotonicity of Aug_s. Thus, Aug_s gives a one-to-one correspondence between $\mathcal{D}(\varphi) \setminus \{1, s-1\}$ and $\mathcal{D}(T_s(\varphi)) \setminus \{1, s-1, s\}$). It

remains to analyze values $s-1$ and s. We have $T_s(\varphi)(s+1) = Aug_s(\varphi)(s) < Aug_s(\varphi)(s-1) = T_s(\varphi)(s-1)$ if and only if $s-1 \in \mathcal{D}(\varphi)$. In this case, $s-1$ or s is a descent of $T_s(\varphi)$. The insertion of $T_s(\varphi)(s) = \varphi(1)$ may or may not create a new descent. For a given $\varphi \in \mathbf{S}_n^c$, in each monotonic run of φ on indices $\{2, \ldots, n\}$, there is exactly one position where we can place $\varphi(1)$ without creating a new descent, otherwise we create exactly one new descent.

How many $T_s(\varphi)$ can we have with $|\mathcal{D}(T_s(\varphi)) \setminus \{1\}| = d$? For each $\varphi \in \mathbf{S}_n^c$ with $|\mathcal{D}(\varphi) \setminus \{1\}| = d$, we have $(d+1)$ possibilities to choose s (φ has $d+1$ monotonic runs on $\{2, \ldots, n\}$). For each $\varphi \in \mathbf{S}_n^c$ with $|\mathcal{D}(\varphi) \setminus \{1\}| = d-1$, we have $(n-d)$ possibilities to choose s. These permutations are all different as $T_s(\varphi) \neq T_r(\psi)$ if $s \neq r$ or $\varphi \neq \psi$. There is no other way to get a permutation $\psi \in \mathbf{S}_{n+1}^c$ with $|\mathcal{D}(\psi) \setminus \{1\}| = d$. We conclude that $P(n,d) = (d+1)P(n-1,d) + (n-d)P(n-1,d-1)$. This proves the Theorem. □

[1] M. Crochemore, J. Désarménien, D. Perrin, A note on the Burrows-Wheeler transformation, Theoretical Computer Science 332 (1-3) (2005) 567–572.

[2] H. Bannai, S. Inenaga, A. Shinohara, M. Takeda, Inferring Strings from Graphs and Arrays, in: B. Rovan, P. Vojtás (Eds.), Mathematical Foundations of Computer Science 2003, Proc. of the 28th International Symposium, MFCS 2003, Bratislava, Slovakia, August 25-29, 2003., vol. 2747 of *Lecture Notes in Computer Science*, Springer, 208–217, 2003.

[3] K.-B. Schürmann, J. Stoye, Counting suffix arrays and strings, Theoretical Computer Science 395 (2008) 220 – 234.

[4] M. He, J. I. Munro, S. S. Rao, A categorization theorem on suffix arrays with applications to space efficient text indexes, in: Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, SODA'05, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 23–32, 2005.

[5] U. Manber, E. W. Myers, Suffix Arrays: A New Method for On-Line String Searches, SIAM J. Comput. 22 (5) (1993) 935–948.

[6] G. H. Gonnet, R. A. Baeza-Yates, T. Snider, New Indices for Text: Pat Trees and Pat Arrays, in: Information Retrieval: Data Structures & Algorithms, Prentice-Hall, 66–82, 1992.

[7] R. Grossi, A quick tour on suffix arrays and compressed suffix arrays, Theoretical Computer Science 412 (27) (2011) 2964 – 2973.

[8] J. Kärkkäinen, P. Sanders, Simple Linear Work Suffix Array Construction, in: J. C. M. Baeten, J. K. Lenstra, J. Parrow, G. J. Woeginger (Eds.), Automata, Languages and Programming, Proc. of the 30th International Colloquium, ICALP 2003, Eindhoven, The Netherlands, June 30 - July 4, 2003., vol. 2719 of *Lecture Notes in Computer Science*, Springer, 943–955, 2003.

[9] D. K. Kim, J. S. Sim, H. Park, K. Park, Linear-Time Construction of Suffix Arrays, in: R. A. Baeza-Yates, E. Chávez, M. Crochemore (Eds.), Combinatorial Pattern Matching, Proc. of the 14th Annual Symposium, CPM 2003, Morelia, Michocán, Mexico, June 25-27, 2003., vol. 2676 of *Lecture Notes in Computer Science*, Springer, 186–199, 2003.

[10] P. Ko, S. Aluru, Space Efficient Linear Time Construction of Suffix Arrays, in: R. A. Baeza-Yates, E. Chávez, M. Crochemore (Eds.), Combinatorial Pattern Matching, Proc. of the 14th Annual Symposium, CPM 2003, Morelia, Michocán, Mexico, June 25-27, 2003., vol. 2676 of *Lecture Notes in Computer Science*, Springer, 200–210, 2003.

[11] M. Burrows, D. Wheeler, A block sorting lossless data compression algorithm, Tech. Rep., Systems Research Center, Technical Report 124, Digital Equipment Corporation, 1994.

[12] G. Navarro, V. Mäkinen, Compressed full-text indexes, ACM Comput. Surv. 39 (1).

[13] I. M. Gessel, C. Reutenauer, Counting Permutations with Given Cycle Structure and Descent Set, J. Comb. Theory, Ser. A 64 (2) (1993) 189–215.

[14] R. Grossi, J. S. Vitter, Compressed Suffix Arrays and Suffix Trees with Applications to Text Indexing and String Matching, SIAM J. Comput. 35 (2) (2005) 378–407.