*Fundamenta Informaticae XXI (2001) 1001–1015* 1001
*IOS Press*

# Combinatorics of Unique Maximal Factorization Families (UMFFs)

**David E. Daykin**
*Department of Mathematics*
*University of Reading, UK*

**Jacqueline W. Daykin**
*Department of Computer Science*
*Royal Holloway & King's College, University of London, UK*
*J.Daykin@cs.rhul.ac.uk, jackie.daykin@kcl.ac.uk*

**W. F. (Bill) Smyth**
*Algorithms Research Group, Department of Computing & Software*
*McMaster University, Hamilton ON L8S 4K1, Canada*
*smyth@mcmaster.ca*
*Digital Ecosystems and Business Intelligence Institute*
*Curtin University, GPO Box U1987*
*Perth WA 6845, Australia*

**Abstract.** Suppose a set $\mathcal{W}$ of strings contains exactly one rotation (cyclic shift) of every primitive string on some alphabet $\Sigma$. Then $\mathcal{W}$ is a circ-UMFF if and only if every word in $\Sigma^+$ has a unique maximal factorization over $\mathcal{W}$. The classic circ-UMFF is the set of Lyndon words based on lexicographic ordering (1958). Duval (1983) designed a linear sequential Lyndon factorization algorithm; a corresponding PRAM parallel algorithm was described by J. Daykin, Iliopoulos and Smyth (1994). Daykin and Daykin defined new circ-UMFFs based on various methods for totally ordering sets of strings (2003), and further described the structure of all circ-UMFFs (2008). Here we prove new combinatorial results for circ-UMFFs, and in particular for the case of Lyndon words. We introduce Acrobat and Flight Deck circ-UMFFs, and describe some of our results in terms of dictionaries. Applications of circ-UMFFs pertain to structured methods for concatenating and factoring strings over ordered alphabets, and those of Lyndon words are wide ranging and multidisciplinary.

Address for correspondence: J. W. Daykin, Department of Computer Science, Royal Holloway College, University of London, Egham, TW20 0EX, UK

**Keywords:** alphabet, circ-UMFF, concatenate, dictionary, factor, lexicographic order, Lyndon, maximal, string, total order, UMFF, word

# 1. Introduction

In this paper we study infinite sets $\mathcal{W}$ of strings on a given alphabet $\Sigma$, $|\Sigma| \geq 2$, that are closed, according to a specified rule, under the reciprocal operations of concatenation and factorization. In particular,

* $\lambda \in \Sigma \Longrightarrow \lambda \in \mathcal{W}$;

* (concatenation) $\boldsymbol{u}, \boldsymbol{v} \neq \boldsymbol{u} \in \mathcal{W} \Longrightarrow$ exactly one of $\boldsymbol{uv}, \boldsymbol{vu} \in \mathcal{W}$.

The concatenation rule implies that every factor $\boldsymbol{w} \in \mathcal{W}$ can be factored, that is, $\boldsymbol{w} \in \mathcal{W}$ and $|\boldsymbol{w}| > 1 \Longrightarrow$ there exist $\boldsymbol{u}, \boldsymbol{v} \neq \boldsymbol{u} \in \mathcal{W}$ such that $\boldsymbol{uv} = \boldsymbol{w}$. We consider cases where, given a string $\boldsymbol{x}$ and a set $\mathcal{W}$, either $\boldsymbol{x} \in \mathcal{W}$ or else $\boldsymbol{x}$ can be factored uniquely into its longest factors that belong to $\mathcal{W}$. We therefore call these sets Unique Maximal Factorization Families (UMFFs) [DD-03]. In particular, we consider ***circ-UMFFs*** — that is, UMFFs that contain exactly one rotation of every primitive string on the given alphabet [DD-08].

We believe that the set of Lyndon words was the first example of a circ-UMFF [CFL-58, L-83]. Although the Lyndon factorization was originally introduced for computing free monoids in Lie algebras, it has since found a wide range of applications. Lyndon words arise in string theoretic problems involving lexicographic ordering such as sorting and searching for substrings, prefixes and suffixes [Du-83], and computing the canonical form of a circular string [IS-92]. Further, Lyndon words have arisen in the analysis of African music [C-04], and even cryptanalysis [P-05]. Naturally then, efficient methods are required for factoring strings, and both sequential [Du-83, D-08] and CRCW parallel RAM algorithms [DIS-94] have been designed for computing Lyndon factorizations of strings (or equivalently words).

The rule that determines whether $\boldsymbol{uv}$ or $\boldsymbol{vu}$ is chosen to belong to $\mathcal{W}$ may depend on a total ordering of the elements of $\mathcal{W}$. For the Lyndon circ-UMFF the elements of $\mathcal{W}$ are ordered lexicographically; thus for $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}$, we choose $\boldsymbol{uv} \in \mathcal{W}$ if and only if $\boldsymbol{u} < \boldsymbol{v}$ in lexicographical order. However, in [DD-03] Daykin and Daykin identified other circ-UMFFs based on alternative definitions of total orders of $\Sigma^*$. Then later [DD-08] they established fundamental properties, independent of the definitions of these orderings, that determine concatenation and factorization over circ-UMFFs.

In this paper we establish new combinatorial properties of factorization families, for instance on the ordering of prefixes and suffixes of factors. We also show that although words in a factorization family may themselves be composed of smaller overlapping factors, by contrast, maximal factors in a factorization over any UMFF are not only disjoint and hence non-overlapping, but unique. This observation has impact on the complexity of factorization algorithms, and arose in the analysis of the parallel Lyndon algorithm of Daykin, Iliopoulos and Smyth [DIS-94]. We further introduce two classes of circ-UMFFs, namely Flight Deck and Acrobat, reflecting the type of order present amongst the letters or substrings in the factors of the defining circ-UMFF.

Lexicographic order is also relevant to this paper. We explore Daykin and Daykin's [DD-08] characterization of circ-UMFFs in the particular case of Lyndon words, and also co-Lyndon words which are based on a simple modification of lexicographic ordering. As all circ-UMFFs are totally ordered sets of strings, we compare them to a classically ordered dictionary. In these dictionaries the ordering of

some factors is forced; however we give new results for other cases where there is a choice of ordering factors. Finally we generalize lexicographic order, from the usual case of ordering words according to their individual letters, to ordering Lyndon factorizations according to their individual Lyndon factors.

We begin by extending existing theory on UMFFs and circ-UMFFs with some new results in Section 2, which are illustrated for Lyndon words in Section 3. We propose some new research problems in Section 4. Note that the terms *string* and *word* mean the same thing (see References) hence we use both throughout.

## 2. Unique Maximal Factorization Families (UMFFs)

Given an integer $n \geq 1$ and a nonempty set of symbols $\Sigma$ (bounded or unbounded), a ***string of length n*** over $\Sigma$ takes the form $\boldsymbol{x} = x_1 ... x_n$ with each $x_i \in \Sigma$. For brevity, we write $\boldsymbol{x} = \boldsymbol{x}[1..n]$ and we let $\boldsymbol{x}[i]$ denote the $i$-th symbol of $\boldsymbol{x}$. The length $n$ of a string $\boldsymbol{x}$ is denoted by $|\boldsymbol{x}|$. The set $\Sigma$ is called an ***alphabet*** whose members are ***letters***, and $\Sigma^+$ denotes the set of all nonempty finite strings over $\Sigma$. The string of length zero is called the ***empty string***, denoted $\boldsymbol{\varepsilon}$; we write $\Sigma^* = \Sigma^+ \cup \{\boldsymbol{\varepsilon}\}$.

A string $\boldsymbol{w}$ is called a ***factor*** of $\boldsymbol{x}[1..n]$ if and only if $\boldsymbol{w} = \boldsymbol{x}[i..j]$ for $1 \leq i \leq j \leq n$. Note that a factor is necessarily nonempty. If $\boldsymbol{x} = \boldsymbol{w_1 w_2 \cdots w_k}$, $1 \leq k \leq n$, then $\boldsymbol{w_1 w_2 \cdots w_k}$ is said to be a ***factorization*** of $\boldsymbol{x}$; moreover, when every factor $\boldsymbol{w_j}$, $1 \leq j \leq k$, belongs to a specified set $\mathcal{W}$, this is a ***factorization of x over*** $\mathcal{W}$, denoted by $F_{\mathcal{W}}(\boldsymbol{x})$.

**Definition 2.1.** A subset $\mathcal{W} \subseteq \Sigma^+$ is a ***factorization family*** (FF) if and only if for every nonempty string $\boldsymbol{x}$ on $\Sigma$ there exists a factorization $F_{\mathcal{W}}(\boldsymbol{x})$.

Observe that every FF must contain $\Sigma$; moreover, every subset of $\Sigma^+$ containing $\Sigma$ is an FF.

For some string $\boldsymbol{x}$ and some FF $\mathcal{W}$, suppose $\boldsymbol{x} = \boldsymbol{w_1 w_2 \cdots w_k}$, where $\boldsymbol{w_j} \in \mathcal{W}$ for every $j \in 1..k$. For some $k' \in 1..k$, write $\boldsymbol{x} = \boldsymbol{u w_{k'} v}$, where $\boldsymbol{u} = \boldsymbol{w_1 w_2 \cdots w_{k'-1}}$ (empty if $k' = 1$) and $\boldsymbol{v} = \boldsymbol{w_{k'+1} w_{k'+2} \cdots w_k}$ (empty if $k' = k$). Suppose that there does not exist a suffix $\boldsymbol{u'}$ of $\boldsymbol{u}$ nor a prefix $\boldsymbol{v''}$ of $\boldsymbol{v}$ such that $\boldsymbol{u' w_{k'} v''} \neq \boldsymbol{w_{k'}}$ and $\boldsymbol{u' w_{k'} v''} \in \mathcal{W}$; then $\boldsymbol{w_{k'}}$ is said to be a ***max factor*** of $\boldsymbol{x}$. If *every* factor $\boldsymbol{w_{k'}}$ is max, then the factorization $F_{\mathcal{W}}(\boldsymbol{x})$ is itself said to be ***max***. Observe that a max factorization must be unique: there exists no other max factorization of $\boldsymbol{x}$ that uses only elements of $\mathcal{W}$.

**Definition 2.2.** Let $\mathcal{W}$ be an FF on an alphabet $\Sigma$. Then $\mathcal{W}$ is a ***unique maximal factorization family*** (UMFF [1]) if and only if there exists a max factorization $F_{\mathcal{W}}(\boldsymbol{x})$ for every string $\boldsymbol{x} \in \Sigma^+$.

We will assume throughout, that when factoring over an UMFF, the factorization is chosen to be the one which is maximal.

Observe that $\Sigma$ is an UMFF, and moreover the definition of UMFFs does not require that $\Sigma$ be ordered. The following result is a characterization of UMFFs, and we provide a new proof of this lemma here.

**Lemma 2.1.** (The **xyz** Lemma [DD-03]) An FF $\mathcal{W}$ is an UMFF if and only if whenever $\boldsymbol{xy}, \boldsymbol{yz} \in \mathcal{W}$ for some nonempty $\boldsymbol{y}$, then $\boldsymbol{xyz} \in \mathcal{W}$.

[1] We read UMFF as a word, hence we will write an UMFF rather than a U-M-F-F.

**Proof:**
First suppose that $\mathcal{W}$ is an UMFF with some $\boldsymbol{xy}, \boldsymbol{yz} \in \mathcal{W}$ for which $\boldsymbol{xyz} \notin \mathcal{W}$. Consider the factorization of $\boldsymbol{xyz}$. Since $\boldsymbol{xy} \in \mathcal{W}$, there must exist a factorization $\boldsymbol{xyz} = \boldsymbol{w_1 w_2 \cdots w_j}$, $j > 1$, where $\boldsymbol{w_1} = \boldsymbol{xyv}$ for some $\boldsymbol{v} \in \Sigma^*$, so that $|\boldsymbol{w_j}| \leq |\boldsymbol{z}|$. Since $\boldsymbol{yz} \in \mathcal{W}$, there must also exist a factorization $\boldsymbol{xyz} = \boldsymbol{w'_1 w'_2 \cdots w'_k}$, $k > 1$, where $\boldsymbol{w'_k} = \boldsymbol{uyz}$ for some $\boldsymbol{u} \in \Sigma^*$. Since $\boldsymbol{y} \neq \boldsymbol{\varepsilon}$, $|\boldsymbol{w_j}| \leq |\boldsymbol{z}| < |\boldsymbol{yz}| \leq |\boldsymbol{w'_k}|$, and so the two factorizations are distinct, contradicting the uniqueness requirement of Definition 2.2. We conclude that $\boldsymbol{xyz} \in \mathcal{W}$.

We need to show that every string $\boldsymbol{v} = \boldsymbol{v}[1..n]$ has a max factorization. Since $\boldsymbol{v}[1] \in \mathcal{W}$, there exists some largest $i_1$ such that $\boldsymbol{w_1} = \boldsymbol{v}[1..i_1] \in \mathcal{W}$. If $i_1 = n$, the factorization is max. If not, there exists some largest $i_2$ such that $\boldsymbol{w_2} = \boldsymbol{v}[i_1+1..i_2] \in \mathcal{W}$. Clearly, since $\mathcal{W}$ is an FF, we can continue in this way to complete a factorization $\boldsymbol{w_1 w_2 \cdots w_k}$ of $\boldsymbol{v}$ such that, at each step, the chosen factor $\boldsymbol{w_j}$ is the longest that exists in $\mathcal{W}$. We claim that this factorization is max. Suppose otherwise. Then there exists $\boldsymbol{u} \in \mathcal{W}$ and a least $j \in 1..k$ such that $\boldsymbol{w_j}$ is a proper factor of $\boldsymbol{u}$. We cannot have $j = 1$ because then $\boldsymbol{w_1}$ could not be max, contrary to our construction. Thus $\boldsymbol{u} = \boldsymbol{pw_jq}$ with at least one of $\boldsymbol{p}, \boldsymbol{q}$ nonempty. If $\boldsymbol{p} = \boldsymbol{\varepsilon}$, then $\boldsymbol{w_jq} \in \mathcal{W}$, so that $\boldsymbol{w_j}$ is not the longest possible factor, again contradicting the construction. Thus $\boldsymbol{p}$ is nonempty and since $j > 1$, there exists $\boldsymbol{w_{j-1}} = \boldsymbol{w'p} \in \mathcal{W}$ for some nonempty $\boldsymbol{w'}$. Applying the **xyz** condition to $\boldsymbol{xy} = \boldsymbol{w'p}$, $\boldsymbol{yz} = \boldsymbol{pw_jq}$, we conclude that $\boldsymbol{w_{j-1}w_jq} \in \mathcal{W}$, contradicting the maximality of $\boldsymbol{w_{j-1}}$. Thus the factorization $\boldsymbol{w_1 w_2 \cdots w_k}$ is max, as required. □

It is an immediate consequence of Lemma 2.1 that there can be no overlapping factors in a unique maximal factorization of a string. In other words, if $F_{\mathcal{W}}(\boldsymbol{x}) = \boldsymbol{w_1 w_2 \cdots w_k}$, then every element of $\mathcal{W}$ is either a factor of some $\boldsymbol{w_i}$, $i \in 1..k$, or else does not occur at all as a factor of $\boldsymbol{x}$. We state this more formally as follows:

**Corollary 2.1.** Suppose $\boldsymbol{x} = \boldsymbol{u_1 u_2 \cdots u_m}$ and $\mathcal{W}$ is an UMFF, where for every $j \in 1..m$, $\boldsymbol{u_j} \in \mathcal{W}$. Then the factorization $F_{\mathcal{W}}(\boldsymbol{x}) = \boldsymbol{w_1 w_2 \cdots w_k}$, where

$$\boldsymbol{w_1} = \boldsymbol{u_{j_0+1} \cdots u_{j_1}}, \boldsymbol{w_2} = \boldsymbol{u_{j_1+1} \cdots u_{j_2}}, \ \ldots\ , \boldsymbol{w_k} = \boldsymbol{u_{j_{k-1}+1} \cdots u_{j_k}},$$

$0 = j_0 < j_1 < j_2 < \cdots < j_{k-1} < j_k = m$.

**Proof:**
Suppose that for some $i \in 1..k$, $\boldsymbol{w_i} = \boldsymbol{u_{j_r+1} \cdots u_{j_{r+1}} u'_{j_{r+1}+1}}$, where $\boldsymbol{u'_{j_{r+1}+1}}$ is a nonempty prefix of $\boldsymbol{u_{j_{r+1}+1}}$. From Lemma 2.1 it follows that $\boldsymbol{u'_{j_{r+1}+1}} = \boldsymbol{u_{j_{r+1}+1}}$. Similarly if we suppose $\boldsymbol{w_i}$ has a nonempty prefix $\boldsymbol{u'_{j_r}}$ that is a suffix of $\boldsymbol{u_{j_r}}$. □

Given two factored strings $\boldsymbol{x}$ and $\boldsymbol{y}$, suppose that it is required, as in the parallel RAM algorithm proposed in [DIS-94], to factor $\boldsymbol{xy}$. This result tells us that the factorization of $\boldsymbol{xy}$ can take place by considering only factors $\boldsymbol{w} \in \mathcal{W}$ that are suffixes of $\boldsymbol{x}$ and prefixes $\boldsymbol{w'} \in \mathcal{W}$ of $\boldsymbol{y}$: such factors are either concatenated or remain disjoint, but will not be split. This observation suggests that the algorithm of [DIS-94] can be extended from Lyndon factorization to circ-UMFFs.

If $\boldsymbol{x} = \boldsymbol{uv}$, then $\boldsymbol{vu}$ is said to be a ***rotation*** (cyclic shift) of $\boldsymbol{x}$, specifically the $|\boldsymbol{u}|^{\text{th}}$ rotation $R_{|\boldsymbol{u}|}(\boldsymbol{x})$ of $\boldsymbol{x}$, where $|\boldsymbol{u}| \in 0..|\boldsymbol{x}|$. Note that $R_0(\boldsymbol{x}) = R_{|\boldsymbol{x}|}(\boldsymbol{x})$. A string $\boldsymbol{x}$ is said to be a ***repetition*** if and only if it has a factorization $\boldsymbol{x} = \boldsymbol{u}^k$ for some integer $k > 1$; otherwise, $\boldsymbol{x}$ is said to be ***primitive***. Observe

that every rotation of a repetition is also a repetition. A string which is both a proper prefix and a proper suffix of a nonempty string $\boldsymbol{x}$ is called a ***border*** of $\boldsymbol{x}$. A string $\boldsymbol{x} = \boldsymbol{x}[1..n]$ has ***period*** $p$ if and only if for every $i \in 1..n-p$, $\boldsymbol{x}[i] = \boldsymbol{x}[i+p]$; the shortest period of $\boldsymbol{x}$ is called ***the period***. Note that $\boldsymbol{x}$ has a border $\boldsymbol{b}$ of length $b$ if and only if it has period $n-b$.

**Definition 2.3.** An UMFF $\mathcal{W}$ over $\Sigma^+$ is a ***circ-UMFF*** [2] if and only if it contains exactly one rotation of every primitive string $\boldsymbol{x} \in \Sigma^+$.

If $\Sigma$ is a totally ordered alphabet then ***lexicographic ordering*** (***lexorder***) $\boldsymbol{u} < \boldsymbol{v}$ with $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^+$ is defined if and only if either $\boldsymbol{u}$ is a proper prefix of $\boldsymbol{v}$, or $\boldsymbol{u} = \boldsymbol{r}a\boldsymbol{s}$, $\boldsymbol{v} = \boldsymbol{r}b\boldsymbol{t}$ for some $a, b \in \Sigma$ such that $a < b$ and for some $\boldsymbol{r}, \boldsymbol{s}, \boldsymbol{t} \in \Sigma^*$. We can therefore say that the set of all Lyndon words is a circ-UMFF, where the rotation chosen from the set of rotations of each primitive string is the one that is least in the lexorder derived from an ordering of the letters of the alphabet $\Sigma$ (see [CFL-58], [DD-08], [Du-83], and [L-83] for further discussion of the Lyndon circ-UMFF). (Note that the choices of rotations for the words of length two for a circ-UMFF actually induces a total order on a given unordered alphabet, see [DD-08].) Consider the following selection of Lyndon words based on different orderings of letters in the alphabet $\Sigma = \{a, b, c\}$.

**Example 2.1.** Let $\mathcal{L}$ denote the Lyndon circ-UMFF, and $\boldsymbol{x} = aabac$ on $\Sigma = \{a, b, c\}$.
(i) If $a$ is the least letter, then $R_0(\boldsymbol{x}) = aabac \in \mathcal{L}$.
(ii) If $b$ is the least letter, then $R_2(\boldsymbol{x}) = bacaa \in \mathcal{L}$.
(iii) If $c$ is the least letter, then $R_4(\boldsymbol{x}) = caaba \in \mathcal{L}$.

Indeed, we could make use of other consistent rules to select the rotation of a string to be assigned to a circ-UMFF:

**Example 2.2.** Suppose that for each primitive $\boldsymbol{x}$ we consider the reversed string

$$\overline{\boldsymbol{x}} = \boldsymbol{x}[n]\boldsymbol{x}[n-1]\cdots\boldsymbol{x}[1],$$

and observe that for every $j \in 0..n-1$, $\overline{R_j(\boldsymbol{x})} = R_{n-j}(\overline{\boldsymbol{x}})$. Then a circ-UMFF is formed by choosing the rotation of each $\boldsymbol{x}$ to be $\overline{\boldsymbol{y}}$, where $\boldsymbol{y}$ is the least rotation of $\overline{\boldsymbol{x}}$.

Referring to Example 2.1, in the case that $b$ is the least letter, the rule in Example 2.2, with the order for 'least' being lexorder, leads to the choice of $R_3(\boldsymbol{x}) = acaab$ for a new circ-UMFF, called ***co-Lyndon*** (**co-$\mathcal{L}$**). We call the ordering based on lexorder of reversed strings ***co-lexorder***[3]. So for example, over the Roman alphabet the word *google*, although not a Lyndon word is a co-Lyndon word, as it is least amongst its rotations in co-lexorder.

We now define an order that is specific to each circ-UMFF and determined only by its particular properties, not necessarily by any ordering of the strings of $\Sigma^+$.

**Definition 2.4.** If a circ-UMFF $\mathcal{W}$ contains strings $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{uv}$, we write $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$ (called the ***$\mathcal{W}$-order***).

[2] circ-UMFFs were originally defined with respect to circulant matrices in [DD-08]; here we adopt the equivalent terminology of rotations.
[3] See [KS-98, p. 45]; other definitions exist in the literature, for example [CDP-05].

We will show that, in essence, the $\mathcal{W}$-order $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$ 'means' that you can concatenate $\boldsymbol{u}$ and $\boldsymbol{v}$ with respect to $\mathcal{W}$, whereas $\geq_{\mathcal{W}}$ 'means' that concatenation is not possible and hence implies factoring (see Theorem 2.2(3) for the case of concatenation, and Theorem 2.3 for the case of factorization). Furthermore, we will also show that $\mathcal{W}$-order is a total order (see Theorem 2.2(4)). For the Lyndon circ-UMFF, its specific $\mathcal{W}$-order is lexorder, as we see by:

**Theorem 2.1.** (Duval [Du-83]) Let $\mathcal{L}$ be the set of Lyndon words, and suppose $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{L}$. Then $\boldsymbol{uv} \in \mathcal{L}$ if and only if $\boldsymbol{u}$ comes before $\boldsymbol{v}$ in lexorder.

Interestingly, the analogue of Theorem 2.1 does not hold for every circ-UMFF. That is, if the elements of $\Sigma^*$ are somehow totally ordered under $<$, it may happen that for every pair of distinct strings $\boldsymbol{u}$ and $\boldsymbol{v}$, $\boldsymbol{u} < \boldsymbol{v}$ while $\boldsymbol{v} <_{\mathcal{W}} \boldsymbol{u}$. We illustrate this phenomenon for the co-Lyndon circ-UMFF. The primitive words $\boldsymbol{u} = cba$ and $\boldsymbol{v} = cbba$ are clearly co-Lyndon words over the Roman alphabet. Analysis of all of the rotations of $\boldsymbol{uv}$ shows that it is co-Lyndon, and by Definition 2.4 we have $\boldsymbol{u} <_{\text{co-}\mathcal{L}} \boldsymbol{v}$. However, $\boldsymbol{v}$ comes before $\boldsymbol{u}$ in co-lexorder, that is $\boldsymbol{v} <_{\text{co-lex}} \boldsymbol{u}$! In other words, $\mathcal{W}$-order can be defined quite independently of the ordering of the elements of $\Sigma^*$.

The following theorem reveals structural properties of circ-UMFFs that prescribe ordered concatenating and factoring of strings. The theorem also shows that not every rotation of a primitive string can necessarily be chosen to belong to a circ-UMFF.

**Theorem 2.2.** ([DD-08]) Let $\mathcal{W}$ be a circ-UMFF.
(1) If $\boldsymbol{u} \in \mathcal{W}$ then $\boldsymbol{u}$ is border-free.
(2) If $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}$ and $\boldsymbol{u} \neq \boldsymbol{v}$ then $\boldsymbol{uv}$ is primitive.
(3) If $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}$ and $\boldsymbol{u} \neq \boldsymbol{v}$ then $\boldsymbol{uv} \in \mathcal{W}$ or $\boldsymbol{vu} \in \mathcal{W}$ (but not both).
(4) If $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{uv} \in \mathcal{W}$ then $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$ and $<_{\mathcal{W}}$ is a total order of $\mathcal{W}$.
(5) If $\boldsymbol{w} \in \mathcal{W}$ and $|\boldsymbol{w}| \geq 2$ then there exist $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}$ with $\boldsymbol{w} = \boldsymbol{uv}$.

From this theorem we conclude that for arbitrary strings $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}$, exactly one of the following is true: $\boldsymbol{u} = \boldsymbol{v}$, $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$, $\boldsymbol{v} <_{\mathcal{W}} \boldsymbol{u}$. In particular, although the order $<_{\mathcal{W}}$ over $\mathcal{W}$ is not reflexive, by its transitivity deduced from part (4) above, it is a *strict order relation*.

Applying part (1) of this theorem to Example 2.1, we see that the string $R_1(\boldsymbol{x}) = abaca$, with border $a$, can never belong to a circ-UMFF, no matter what rule for selection is employed. In fact we can exclude certain classes of strings from circ-UMFFs (see [DD-08] for further limiting examples):

**Proposition 2.1.** Suppose that $\boldsymbol{w}$ is an element of a circ-UMFF $\mathcal{W}$ and $\boldsymbol{u}$ is a nonempty prefix (respectively, suffix) of $\boldsymbol{w}$. Then for every rotation $\boldsymbol{u_j} = R_j(\boldsymbol{u})$, $j \in 0..|\boldsymbol{u}|-1$, $\boldsymbol{wu_j}$ (respectively, $\boldsymbol{u_j w}$) $\notin \mathcal{W}$.

**Proof:**
For prefix $\boldsymbol{u}$, let $\boldsymbol{w} = \boldsymbol{uv}$ and $m = |\boldsymbol{u}|$, then observe that

$$\boldsymbol{u}[1..m]\boldsymbol{vu}[j{+}1..m]\boldsymbol{u}[1..j]$$

is always bordered, contradicting Theorem 2.2(1). The proof when $\boldsymbol{u}$ is a suffix is analogous. □

For the remainder of this section we demonstrate various applications of Theorem 2.2 giving new combinatorial insights into circ-UMFFs.

**Proposition 2.2.** Given a circ-UMFF $\mathcal{W}$ and a string $\boldsymbol{w}$, $|\boldsymbol{w}| \geq 2$, $\boldsymbol{w} \in \mathcal{W}$ if and only if $\boldsymbol{w} = \boldsymbol{uv}$, where $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}$ and $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$.

**Proof:**
Sufficiency is a consequence of Theorem 2.2(3) and Definition 2.4; necessity is Theorem 2.2(5). □

As a consequence, the following result, modified from [DD-08], is easily established. It generalizes the Lyndon factorization theorem [CFL-58] to circ-UMFFs (*cf.* Corollary 2.1).

**Theorem 2.3.** Let $\mathcal{W}$ be a circ-UMFF and suppose $\boldsymbol{x} = \boldsymbol{u_1 u_2 \cdots u_m}$, with each $\boldsymbol{u_j} \in \mathcal{W}$. Then $F_{\mathcal{W}}(\boldsymbol{x}) = \boldsymbol{u_1 u_2 \cdots u_m}$ if and only if $\boldsymbol{u_1} \geq_{\mathcal{W}} \boldsymbol{u_2} \geq_{\mathcal{W}} ... \geq_{\mathcal{W}} \boldsymbol{u_m}$.

Using the Lyndon factorization as an example, we give a sense of the variation in ordering that may occur in circ-UMFFs, even though some ordering is prescribed by Lemma 2.1 and Theorem 2.2.

**Lemma 2.2.** Let $\mathcal{W}$ be a circ-UMFF with $\boldsymbol{xy}, \boldsymbol{yz} \in \mathcal{W}$ for nonempty $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$ (hence $\boldsymbol{x} \neq \boldsymbol{z}$). Then $\boldsymbol{xyz} \in \mathcal{W}$, $\boldsymbol{xyyz} \in \mathcal{W}$, and
(1) $\boldsymbol{xy} <_{\mathcal{W}} \boldsymbol{xyz} <_{\mathcal{W}} \boldsymbol{yz}$;
(2) $\boldsymbol{xy} <_{\mathcal{W}} \boldsymbol{xyyz} <_{\mathcal{W}} \boldsymbol{yz}$;
(3) either $\boldsymbol{xyyzxyz} \in \mathcal{W}$ or $\boldsymbol{xyzxyyz} \in \mathcal{W}$ (but not both).

**Proof:**
An application of Lemma 2.1 and Theorem 2.2(1),(2), and (3). □

We show next that the case $\boldsymbol{xyyz} <_{\mathcal{W}} \boldsymbol{xyz}$ of Lemma 2.2(3) occurs for the Lyndon circ-UMFF based on lexicographic ordering.

**Proposition 2.3.** Let $\mathcal{L}$ be the Lyndon circ-UMFF with $\boldsymbol{xy}, \boldsymbol{yz} \in \mathcal{L}$ for nonempty $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$. Then $\boldsymbol{xy} <_{\mathcal{L}} \boldsymbol{xyyz} <_{\mathcal{L}} \boldsymbol{xyz} <_{\mathcal{L}} \boldsymbol{yz}$.

**Proof:**
In view of Lemma 2.2, we need only verify that $\boldsymbol{xyyz} <_{\mathcal{L}} \boldsymbol{xyz}$. Since in this case the order $<_{\mathcal{L}}$ is lexorder, we may ignore the common prefix $\boldsymbol{xy}$ and consider only whether $\boldsymbol{yz} <_{\mathcal{L}} \boldsymbol{z}$. But this follows from the fact that $\boldsymbol{yz} \in \mathcal{L}$ and so must be less in lexorder than its every proper suffix [Du-83, Proposition 1.2], in particular $\boldsymbol{z}$. □

An analogous argument to the above shows that in the co-Lyndon circ-UMFF co-$\mathcal{L}$, we have $\boldsymbol{xy} <_{\text{co-}\mathcal{L}} \boldsymbol{xyz} <_{\text{co-}\mathcal{L}} \boldsymbol{xyyz} <_{\text{co-}\mathcal{L}} \boldsymbol{yz}$.

The next result shows that a "Lyndon-like" property, $\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v}$, holds whenever both $\boldsymbol{uv}, \boldsymbol{v} \in \mathcal{W}$:

**Lemma 2.3.** Suppose that $\boldsymbol{w}$ is an element of a circ-UMFF $\mathcal{W}$. For every proper prefix $\boldsymbol{u}$ of $\boldsymbol{w}$ such that $\boldsymbol{u} \in \mathcal{W}$ and every proper suffix $\boldsymbol{v}$ of $\boldsymbol{w}$ such that $\boldsymbol{v} \in \mathcal{W}$, $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{w} <_{\mathcal{W}} \boldsymbol{v}$.

**Proof:**
Since by Theorem 2.2(1),(3) neither of the bordered strings $\boldsymbol{wu}$ and $\boldsymbol{vw}$ can be an element of $\mathcal{W}$, it follows from Definition 2.4 and Theorem 2.2(4) that $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{w} <_{\mathcal{W}} \boldsymbol{v}$. □

In particular, the above result tells us that if $\boldsymbol{w} = \boldsymbol{w}[1..n] \in \mathcal{W}$, $n \geq 2$, then $\boldsymbol{w}[1] <_{\mathcal{W}} \boldsymbol{w} <_{\mathcal{W}} \boldsymbol{w}[n]$. Conversely, if $\boldsymbol{w}[n] <_{\mathcal{W}} \boldsymbol{w}[1]$ or $\boldsymbol{w}[n] = \boldsymbol{w}[1]$, then $\boldsymbol{w} \notin \mathcal{W}$. The following result is an immediate consequence of Lemma 2.3:

**Lemma 2.4.** ([DD-08]) Suppose that $\boldsymbol{w}$ is an element of a circ-UMFF $\mathcal{W}$. If $\boldsymbol{u_1}, \boldsymbol{u_2}, \ldots, \boldsymbol{u_{k_1}}$ are all the proper prefixes of $\boldsymbol{w}$ in increasing order of length that belong to $\mathcal{W}$, and if $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_{k_2}}$ are all the proper suffixes of $\boldsymbol{w}$ in decreasing order of length that belong to $\mathcal{W}$, then

$$\boldsymbol{u_1} <_{\mathcal{W}} \boldsymbol{u_2} <_{\mathcal{W}} \cdots <_{\mathcal{W}} \boldsymbol{u_{k_1}} <_{\mathcal{W}} \boldsymbol{w} <_{\mathcal{W}} \boldsymbol{v_1} <_{\mathcal{W}} \boldsymbol{v_2} <_{\mathcal{W}} \cdots <_{\mathcal{W}} \boldsymbol{v_{k_2}}.$$

Recall that for the Lyndon circ-UMFF $\mathcal{L}$, this lemma holds more generally for *every* prefix of $\boldsymbol{w} \in \mathcal{L}$, no matter whether or not these strings are in $\mathcal{L}$ [Du-83]. The next lemma shows that if $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$, then $\boldsymbol{u}$ is less in $\mathcal{W}$-order than any right extension of $\boldsymbol{v}$ that is also in $\mathcal{W}$:

**Lemma 2.5.** Suppose $\boldsymbol{u} \in \mathcal{W}$ and $\boldsymbol{v} \in \mathcal{W}$, where $\mathcal{W}$ is a circ-UMFF. If $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$, then for every string $\boldsymbol{w}$ such that $\boldsymbol{vw} \in \mathcal{W}$, $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{vw}$.

**Proof:**
Observe first that if $\boldsymbol{u} = \boldsymbol{vw}$, then by Lemma 2.3, $\boldsymbol{v} <_{\mathcal{W}} \boldsymbol{v}$, a contradiction. Thus $\boldsymbol{u} \neq \boldsymbol{vw}$, so that by Theorem 2.2(3) either $\boldsymbol{uvw}$ or $\boldsymbol{vwu}$ is in $\mathcal{W}$. If $\boldsymbol{vwu} \in \mathcal{W}$, Lemma 2.3 implies $\boldsymbol{v} <_{\mathcal{W}} \boldsymbol{u}$, a contradiction. Thus $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{vw}$, as required. □

We can generate certain types of new factors in a circ-UMFF from repetitions of given factors:

**Lemma 2.6.** ([DD-08]) Let $\mathcal{W}$ be a circ-UMFF. If $\boldsymbol{u_1}, \boldsymbol{u_2}, ..., \boldsymbol{u_m} \in \mathcal{W}$ with $\boldsymbol{u_1} <_{\mathcal{W}} \boldsymbol{u_2} <_{\mathcal{W}} ... <_{\mathcal{W}} \boldsymbol{u_m}$ and $m \geq 2$, and if $k_1, k_2, ..., k_m > 0$ are integers, then $\boldsymbol{u_1}^{k_1}\boldsymbol{u_2}^{k_2}...\boldsymbol{u_m}^{k_m} \in \mathcal{W}$.

Of course, Lemma 2.6 also applies to any subsequence of the factors $\boldsymbol{u_1}, \boldsymbol{u_2}, ..., \boldsymbol{u_m}$, so that $\boldsymbol{u_{i_1}^{k_1}u_{i_2}^{k_2}} \cdots \boldsymbol{u_{i_r}^{k_r}} \in \mathcal{W}$, where $1 \leq i_1 < i_2 < \cdots < i_r \leq m$. As a special case of Lemmas 2.5 and 2.6, we see that for $r \in 1..|\Sigma|$ such that $1 \leq i_1 < i_2 < \cdots < i_r \leq |\Sigma|$,

$$\lambda_{i_1} <_{\mathcal{W}} \lambda_{i_1}\lambda_{i_2} <_{\mathcal{W}} \cdots <_{\mathcal{W}} \lambda_{i_1}\lambda_{i_2}\cdots\lambda_{i_r},$$

where $\lambda_{i_j} \in \Sigma, 1 \leq j \leq r$. Note however that the usual lexicographic or positional property of order — that $i_1 < i_2 < i_3 \Rightarrow i_1 i_2 < i_1 i_3$ — does not necessarily hold for circ-UMFFs. For example, on the binary alphabet $\{0, 1\}$, $0 <_{\mathcal{W}} 1$, even though it follows from the above lemmas that for every circ-UMFF, $0 <_{\mathcal{W}} 011 <_{\mathcal{W}} 1$, it may also be true that $010011 \in \mathcal{W}$ — in other words, that $01 <_{\mathcal{W}} 0011$, in which case $\mathcal{W}$ would not be the Lyndon circ-UMFF. (See [DD-08], Section 5 'To Find all circ-UMFFs', for details of the procedure for constructing a circ-UMFF.)

We will now explore "dictionary" type properties of circ-UMFFs, showing that some orders of concatenations are predetermined.

**Proposition 2.4.** Suppose $\mathcal{W}$ is a circ-UMFF defined on $\Sigma = \{\lambda_1, \lambda_2, \ldots.\}$, and let $\boldsymbol{u} \in \Sigma^+$.
(1) If $\boldsymbol{u} \in \mathcal{W}$ and $\lambda_i <_{\mathcal{W}} \boldsymbol{u}$ then $\lambda_i <_{\mathcal{W}} \lambda_i \boldsymbol{u}$.
(2) If $\boldsymbol{u} \in \mathcal{W}$ and $\boldsymbol{u} <_{\mathcal{W}} \lambda_i$ then $\boldsymbol{u}\lambda_i <_{\mathcal{W}} \lambda_i$.
(3) If $\boldsymbol{u} \in \mathcal{W}$ and $\lambda_i <_{\mathcal{W}} \lambda_j$, and $\lambda_j <_{\mathcal{W}} \boldsymbol{u}$ then $\lambda_i <_{\mathcal{W}} \lambda_j \boldsymbol{u}$.
(4) If $\lambda_i \boldsymbol{u} \in \mathcal{W}$ then $\lambda_i <_{\mathcal{W}} \lambda_i \boldsymbol{u}$.
(5) If $\lambda_i <_{\mathcal{W}} \lambda_j$ and $\lambda_j \boldsymbol{u} \in \mathcal{W}$ then $\lambda_i <_{\mathcal{W}} \lambda_j \boldsymbol{u}$.

**Proof:**
Parts (1),(2),(3) are derived from Definition 2.4 and Theorem 2.2, part (4) is a special case of Lemma 2.4, part (5) a special case of Lemma 2.5. □

By contrast, choice for concatenation arises in certain contexts. For instance, even if $\lambda_i <_{\mathcal{W}} \lambda_j$ as above, then for some nonempty $\boldsymbol{u}$, it is possible that either $\lambda_i \boldsymbol{u} <_{\mathcal{W}} \lambda_j$ or $\lambda_j <_{\mathcal{W}} \lambda_i \boldsymbol{u}$ in $\mathcal{W}$; if we choose the former we get:

**Proposition 2.5.** Suppose $\mathcal{W}$ is a circ-UMFF over $\Sigma = \{\lambda_1, \lambda_2, \ldots.\}$, with $\lambda_i <_{\mathcal{W}} \lambda_j$. Suppose $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^*$ and $\lambda_i \boldsymbol{u}, \lambda_j \boldsymbol{v} \in \mathcal{W}$. If $\lambda_i \boldsymbol{u} <_{\mathcal{W}} \lambda_j$, then $\lambda_i \boldsymbol{u} <_{\mathcal{W}} \lambda_j \boldsymbol{v}$.

**Proof:**
From $\lambda_i <_{\mathcal{W}} \lambda_j$ we have that $\lambda_i \boldsymbol{u}$ and $\lambda_j \boldsymbol{v}$ are distinct. Then applying Theorem 2.2(3) to $\lambda_i \boldsymbol{u}$ and $\lambda_j \boldsymbol{v}$, we have either $\lambda_j \boldsymbol{v} \lambda_i \boldsymbol{u} \in \mathcal{W}$ or $\lambda_i \boldsymbol{u} \lambda_j \boldsymbol{v} \in \mathcal{W}$. Without loss of generality, let us assume that $\lambda_j \boldsymbol{v} \lambda_i \boldsymbol{u} \in \mathcal{W}$. Applying Lemma 2.1 to $\lambda_j \boldsymbol{v} \lambda_i \boldsymbol{u}$ and $\lambda_i \boldsymbol{u} \lambda_j$ yields the bordered string $\lambda_j \boldsymbol{v} \lambda_i \boldsymbol{u} \lambda_j \in \mathcal{W}$, contradicting Theorem 2.2(1). Thus $\lambda_i \boldsymbol{u} \lambda_j \boldsymbol{v} \in \mathcal{W}$, and the result follows from Proposition 2.2. □

However, had we instead chosen $\lambda_j <_{\mathcal{W}} \lambda_i \boldsymbol{u}$, we could have gone on to possibly choose either $\lambda_j \boldsymbol{v} <_{\mathcal{W}} \lambda_i \boldsymbol{u}$ or $\lambda_i \boldsymbol{u} <_{\mathcal{W}} \lambda_j \boldsymbol{v}$ in $\mathcal{W}$, and so on.

We now identify two interesting classes of circ-UMFF, which to our knowledge are not exhaustive:

**Definition 2.5.** A circ-UMFF $\mathcal{W}$ is said to be ***Type Flight Deck*** if and only if $\boldsymbol{w}[1...n] \in \mathcal{W}$ with $|\boldsymbol{w}| \geq 2$ implies that for every $i \in 2..n$, $\boldsymbol{w}[1] \leq_{\mathcal{W}} \boldsymbol{w}[i]$.

**Definition 2.6.** A circ-UMFF $\mathcal{W}$ is said to be ***Type Acrobat*** if and only if it contains elements $\boldsymbol{u}\boldsymbol{v_1}$, $\boldsymbol{w}$ and $\boldsymbol{u}\boldsymbol{v_2}$, nonempty $\boldsymbol{u}$ not a prefix of $\boldsymbol{w}$, such that

$$\boldsymbol{u}\boldsymbol{v_1} <_{\mathcal{W}} \boldsymbol{w} <_{\mathcal{W}} \boldsymbol{u}\boldsymbol{v_2}.$$

Suppose $\Sigma = \{a <_{\mathcal{W}} b <_{\mathcal{W}} c <_{\mathcal{W}} d\}$ for some $\mathcal{W}$-order. Then examples of elements chosen for a Flight Deck circ-UMFF over $\Sigma$ are $\lambda_i \boldsymbol{u} = ac$ and $\lambda_j \boldsymbol{v} = bd$, so that $\lambda_i \boldsymbol{u} \lambda_j \boldsymbol{v} = acbd \in \mathcal{W}$, whereas $\lambda_j \boldsymbol{v} \lambda_i \boldsymbol{u} = bdac \notin \mathcal{W}$ since this string contains the internal letter $a$ which is less than its first letter $b$. Instances of circ-UMFFs satisfying the Flight Deck condition include: all binary circ-UMFFs (if any word starts with 0, then they all start with 0 and end with 1 and there are no other letters to consider in the alphabet), and the Lyndon circ-UMFF (no rotation, hence no letter can be lexicographically less than the first letter). To show that the co-Lyndon circ-UMFF cannot be of type Flight Deck, consider the alphabet

of integers $\{1 < 2 < 3 < ...\}$, then the $\mathcal{W}$-order (co-lexorder co-$\mathcal{L}$) is $\{1 >_{\text{co-}\mathcal{L}} 2 >_{\text{co-}\mathcal{L}} 3 >_{\text{co-}\mathcal{L}} ...\}$ and while 321 and 231 are both co-Lyndon words, the latter word 231 does not satisfy the Flight Deck condition since the second letter is less than the first in this $\mathcal{W}$-order, co-lexorder. Observe also that the Lyndon circ-UMFF cannot be of type Acrobat due to the conditions on $\boldsymbol{uv_1}$, $\boldsymbol{w}$ and $\boldsymbol{uv_2}$.

**Lemma 2.7.** Suppose $\mathcal{W}$ is a Flight Deck circ-UMFF over $\Sigma$ and let $\mu \in \Sigma$. Suppose $\boldsymbol{w} \in \mathcal{W}$ with $|\boldsymbol{w}| \geq 2$, and the letter $\lambda$ occurs in $\boldsymbol{w}$ at least once.
(1) If $\boldsymbol{w}[1] = \lambda$, then $\lambda\boldsymbol{w} \in \mathcal{W}$; otherwise, $\boldsymbol{w}\lambda \in \mathcal{W}$.
(2) If $\boldsymbol{w}[1] \geq_{\mathcal{W}} \mu$, then $\mu\boldsymbol{w} \in \mathcal{W}$; otherwise, $\boldsymbol{w}\mu \in \mathcal{W}$.

**Proof:**
In either case, since $\lambda, \mu \in \mathcal{W}$ and $\lambda, \mu \neq \boldsymbol{w}$ we can apply Theorem 2.2(3). Part (1) is then a consequence of Theorem 2.2(1) and the definition of Flight Deck; part (2) follows similarly. □

We now consider the $\mathcal{W}$-order of suffixes for these two types of circ-UMFFs, namely Flight Deck and Acrobat (*cf.* Lemma 2.4).

**Theorem 2.4.** Suppose that $\boldsymbol{w} = \boldsymbol{uv}$ is an element of a circ-UMFF $\mathcal{W}$, with $\boldsymbol{u}$ and $\boldsymbol{v}$ nonempty. Then either $\boldsymbol{wv} \in \mathcal{W}$ or $\boldsymbol{v_2wv_1} \in \mathcal{W}$, where $\boldsymbol{v} = \boldsymbol{v_1v_2}$, $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$ nonempty. In the latter case W can be Type Acrobat.

**Proof:**
If $\boldsymbol{v} \in \mathcal{W}$, then since $\boldsymbol{v}$ and $\boldsymbol{w}$ are distinct, applying Theorem 2.2(3) either $\boldsymbol{wv}$ or $\boldsymbol{vw}$ is an element of $\mathcal{W}$; since $\boldsymbol{vw}$ is bordered, it follows from Theorem 2.2(1) that $\boldsymbol{vw} \notin \mathcal{W}$, thus $\boldsymbol{wv} \in \mathcal{W}$. Hence if this case does not hold we may suppose that neither $\boldsymbol{v}$ nor $\boldsymbol{wv}$ is an element of $\mathcal{W}$.

Since $\boldsymbol{wv} \notin \mathcal{W}$, then by Definition 2.3, if $\boldsymbol{wv}$ is primitive it follows that some rotation of $\boldsymbol{wv}$ must be in $\mathcal{W}$. So first we will establish that $\boldsymbol{wv}$ is primitive, and then choose a rotation for $\mathcal{W}$.

Suppose that $\boldsymbol{wv} = \boldsymbol{uvv}$ is a repetition. Then $\boldsymbol{wv} = \boldsymbol{z}^r$ for some integer $r \geq 2$. Therefore $|\boldsymbol{z}| < |\boldsymbol{uv}|$, and so $\boldsymbol{w} = \boldsymbol{uv}$ has period $|\boldsymbol{z}|$, hence a nonempty border, contradicting Theorem 2.2(1). Thus $\boldsymbol{wv}$ is not a repetition, and so some rotation of $\boldsymbol{wv}$ is an element of $\mathcal{W}$.

First suppose that a rotation of the form $\overline{\boldsymbol{w}} = \boldsymbol{u_2v^2u_1}$ is in $\mathcal{W}$ for nonempty $\boldsymbol{u_1}$, $\boldsymbol{u_2}$ such that $\boldsymbol{u} = \boldsymbol{u_1u_2}$. But then applying Lemma 2.1 to $\boldsymbol{xy} = \overline{\boldsymbol{w}}$ and $\boldsymbol{yz} = \boldsymbol{u_1u_2v}$ implies that the bordered word $\boldsymbol{u_2v^2u_1u_2v}$ is in $\mathcal{W}$, contradicting Theorem 2.2(1). Suppose then that a rotation of the form $\overline{\boldsymbol{w}} = \boldsymbol{v''vuv'} \in \mathcal{W}$ for nonempty $\boldsymbol{v'}$, $\boldsymbol{v''}$ such that $\boldsymbol{v} = \boldsymbol{v'v''}$. Similarly applying Lemma 2.1 to $\boldsymbol{xy} = \boldsymbol{uv'v''}$ and $\boldsymbol{yz} = \overline{\boldsymbol{w}}$ implies that the bordered word $\boldsymbol{uv'v''vuv'}$ is in $\mathcal{W}$, again a contradiction. Likewise, the rotations $\overline{\boldsymbol{w}} = \boldsymbol{vvu}$ and $\overline{\boldsymbol{w}} = \boldsymbol{vuv}$ cannot belong to $\mathcal{W}$.

Thus we conclude that the unique rotation of $\boldsymbol{wv}$ that belongs to $\mathcal{W}$ takes the form $\boldsymbol{v_2uvv_1}$, where $\boldsymbol{v_1}$,$\boldsymbol{v_2}$ are by hypothesis nonempty. Then by Theorem 2.2(5) we can split $\boldsymbol{v_2uvv_1}$ into a pair of factors, both of them in $\mathcal{W}$:

- Suppose $\boldsymbol{v_2u_1} \in \mathcal{W}$, $\boldsymbol{u_2vv_1} \in \mathcal{W}$ for some nonempty $\boldsymbol{u_1}$. But then applying Lemma 2.1 to $\boldsymbol{uv} = \boldsymbol{u_1u_2v_1v_2}$ and $\boldsymbol{v_2u_1}$, we find that the bordered word $\boldsymbol{u_1u_2v_1v_2u_1}$ is in $\mathcal{W}$, a contradiction.
- Suppose $\boldsymbol{v_2uv'} \in \mathcal{W}$, $\boldsymbol{v''v_1} \in \mathcal{W}$ for some nonempty $\boldsymbol{v'}$ such that $\boldsymbol{v} = \boldsymbol{v'v''}$. (Assume $\boldsymbol{v''}$ is nonempty for otherwise $\boldsymbol{v_2uv'}$ is bordered.) But then applying Lemma 2.1 to $\boldsymbol{v_2uv'}$ and $\boldsymbol{uv} = \boldsymbol{uv'v''}$, we find that the bordered word $\boldsymbol{v_2uv}$ is in $\mathcal{W}$, again a contradiction.

Thus the partition of $\boldsymbol{v_2uvv_1}$ may take the form $\boldsymbol{v_2} \in \mathcal{W}$, $\boldsymbol{uvv_1} \in \mathcal{W}$, where $\boldsymbol{v_2} <_{\mathcal{W}} \boldsymbol{uvv_1}$. In this case we have distinct $\boldsymbol{uv}$ and $\boldsymbol{v_2}$ both belonging to $\mathcal{W}$, and so applying Theorem 2.2(3),(1) we know $\boldsymbol{v_2uv} \notin \mathcal{W}$. Hence, also applying Theorem 2.2(4) we deduce that

$$\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v_2} <_{\mathcal{W}} \boldsymbol{uvv_1},$$

so that $\mathcal{W}$ is Type Acrobat. □

Moreover, notice above that since $\boldsymbol{v_2uvv_1} \in \mathcal{W}$, by further application of Theorem 2.2 we also have the Acrobat instance

$$\boldsymbol{uvv_2} <_{\mathcal{W}} \boldsymbol{v_2uvv_1} <_{\mathcal{W}} \boldsymbol{uvv_1}.$$

The partition of Theorem 2.2(5) is not necessarily unique, so consider the possibility that $\boldsymbol{v_2} \in \mathcal{W}$ and $\boldsymbol{v_1} = \boldsymbol{v_1'v_1''}$, where $\boldsymbol{v_1'}, \boldsymbol{v_1''}$ are nonempty, and we split $\boldsymbol{v_2uvv_1}$ through $\boldsymbol{v_1}$ so that $\boldsymbol{v_2uvv_1'}, \boldsymbol{v_1''}$ are in $\mathcal{W}$ with $\boldsymbol{v_2uvv_1'} <_{\mathcal{W}} \boldsymbol{v_1''}$. Since $\boldsymbol{v_2}, \boldsymbol{v_2uvv_1'v_1''}$ and $\boldsymbol{v_1''}$ are in $\mathcal{W}$, from Lemma 2.3 we know that $\boldsymbol{v_2} <_{\mathcal{W}} \boldsymbol{v_1''}$. We now have that $\boldsymbol{uv}, \boldsymbol{v_2}, \boldsymbol{v_2v_1''}, \boldsymbol{v_2uvv_1'v_1''}, \boldsymbol{v_2uvv_1'}$ and $\boldsymbol{v_1''}$ are all in $\mathcal{W}$, furthermore they are all distinct. Hence we can apply Theorem 2.2(1),(3) and (4) to order permutations of these distinct factors into a total order. Consider the three possible concatenations $\boldsymbol{v_2v_1''} <_{\mathcal{W}} \boldsymbol{v_2uvv_1'v_1''}$ or $\boldsymbol{v_2uvv_1'v_1''} <_{\mathcal{W}} \boldsymbol{v_2v_1''}$, $\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v_1''}$ or $\boldsymbol{v_1''} <_{\mathcal{W}} \boldsymbol{uv}$, and $\boldsymbol{v_2uvv_1'v_1''} <_{\mathcal{W}} \boldsymbol{uvv_1''}$ or $\boldsymbol{uvv_1''} <_{\mathcal{W}} \boldsymbol{v_2uvv_1'v_1''}$. If we choose the former in each case (recall from earlier in Section 2 that some, but not all, orderings are predetermined) we have

$$\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v_2v_1''} <_{\mathcal{W}} \boldsymbol{v_2uvv_1'v_1''} <_{\mathcal{W}} \boldsymbol{uvv_1''},$$

and so

$$\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v_2v_1''v_2uvv_1'v_1''} <_{\mathcal{W}} \boldsymbol{uvv_1''},$$

and this total order belongs to a type Acrobat circ-UMFF $\mathcal{W}$.

Finally, suppose that $\boldsymbol{v_2} \in \mathcal{W}$ has $|\boldsymbol{v_2}| \geq 2$, and suppose also that we can split $\boldsymbol{v_2uvv_1}$ through $\boldsymbol{v_2} = \boldsymbol{v_2'v_2''}$ so that $\boldsymbol{v_2'}$, $\boldsymbol{v_2''}$ are nonempty and distinct, with $\boldsymbol{v_2'}, \boldsymbol{v_2''uvv_1} \in \mathcal{W}$ and $\boldsymbol{v_2'} <_{\mathcal{W}} \boldsymbol{v_2''uvv_1}$. Then we have the distinct elements $\boldsymbol{uv}, \boldsymbol{v_2'}, \boldsymbol{v_2''uvv_1}$ all in $\mathcal{W}$. When applying Theorem 2.2 as before, if we choose $\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v_2'}$, then since we have both $\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v_2''uvv_1}$ and $\boldsymbol{v_2'} \neq \boldsymbol{v_2''}$, this case yields the Acrobat instance $\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v_2''uvv_1} <_{\mathcal{W}} \boldsymbol{uvv_2'}$.

Observe that, if $\boldsymbol{w} = \boldsymbol{uv}$ in Theorem 2.4 satisfies the Flight Deck condition so that for every $i \in 2..|\boldsymbol{w}|$, $\boldsymbol{w}[1] \leq_{\mathcal{W}} \boldsymbol{w}[i]$, then clearly $\boldsymbol{wv} = \boldsymbol{uvv}$ satisfies the Flight Deck condition too.

## 3. The Lyndon Dictionary

Here we illustrate parts (1)–(5) of Theorem 2.2 for the case that $\mathcal{W}$ is the Lyndon circ-UMFF $\mathcal{L}$, so that UMFF $\mathcal{L}$-order is lexicographic: thus for brevity we write $<$ instead of $<_{\mathcal{L}}$. We emphasize that these are known properties [CFL-58, Du-83] of Lyndon words, briefly reviewed here to link them to the results established in Section 2 more generally for circ-UMFFs.

Assume $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathcal{L}$ are distinct nonempty Lyndon words:

(1) It is well known [Du-83] that Lyndon words are border-free.

(2) If $\boldsymbol{uv}$ is a repetition, then at least one of $\boldsymbol{u}, \boldsymbol{v}$ is bordered, hence not in $\mathcal{L}$, a contradiction.

(3) For $\boldsymbol{u} < \boldsymbol{v}$ Duval [Du-83] shows that $\boldsymbol{uv} \in \mathcal{L}$. Since $\boldsymbol{uv}$ is the lexicographically least rotation, $\boldsymbol{vu} \notin \mathcal{L}$.

(4) Assume $\boldsymbol{u} < \boldsymbol{v}$ and $\boldsymbol{v} < \boldsymbol{w}$. Then $\boldsymbol{uv}$ and $\boldsymbol{vw}$ are both Lyndon words. If the order is not total, so that $\boldsymbol{w} < \boldsymbol{u}$, then $\boldsymbol{wu} \in \mathcal{L}$. If we now apply Lemma 2.1 to $\boldsymbol{uv}$ and $\boldsymbol{vw}$, we find that $\boldsymbol{uvw} \in \mathcal{L}$, and similarly applying Lemma 2.1 to $\boldsymbol{vw}$ and $\boldsymbol{wu}$ implies that $\boldsymbol{vwu} \in \mathcal{L}$. Since $\boldsymbol{uvw}$ is a Lyndon word, the rotation $\boldsymbol{vwu}$ cannot be a Lyndon word too. Thus $\boldsymbol{u} < \boldsymbol{w}$ and $\boldsymbol{u} < \boldsymbol{v} < \boldsymbol{w}$.

(5) Suppose $\boldsymbol{w} = \boldsymbol{w}[1..n] \in \mathcal{L}$, $n \geq 2$. We want to show that we can always partition $\boldsymbol{w} = \boldsymbol{uv}$ such that $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{L}$. Applying Lemma 2.3 we can write $\boldsymbol{w} = \lambda^h \boldsymbol{y} \mu^k$, where $\boldsymbol{w}[1] = \lambda < \mu = \boldsymbol{w}[n]$, the positive integers $h$ and $k$ are both maximal ($\boldsymbol{w}[h+1] \neq \lambda$ and $\boldsymbol{w}[k-1] \neq \mu$), and $\boldsymbol{y}$ is possibly empty. Let $r$ be the position of the rightmost occurrence of $\lambda$ in $\boldsymbol{w}$. If $r = 1$, choose $\boldsymbol{u} = \boldsymbol{w}[1..n-1], \boldsymbol{v} = \boldsymbol{w}[n]$. If $r > 1$, look for the rightmost position $s < r$ such that $\boldsymbol{w}[s] > \boldsymbol{w}[r] = \lambda$. If there is no such $s$, choose $\boldsymbol{u} = \boldsymbol{w}[1], \boldsymbol{v} = \boldsymbol{w}[2..n]$; otherwise, choose $\boldsymbol{u} = \boldsymbol{w}[1..s], \boldsymbol{v} = \boldsymbol{w}[s+1..n] = \lambda^{r-s} \boldsymbol{w}[r+1..n]$.

Since by (4) the infinite set of all Lyndon words over an arbitrary alphabet is totally ordered in lexorder, it may be considered to be a *"dictionary"*, and likewise the infinite set of co-Lyndon words. Recall that the Lyndon circ-UMFF is of type Flight Deck but not the co-Lyndon circ-UMFF (see Section 2). We will now show that the co-Lyndon circ-UMFF is of type Acrobat. Further, the following example compares these two dictionaries, over the ordered Roman alphabet, to the usual English dictionary.

**Example 3.1.** The words fowl, goose, growl, howl, oriole, owl, scowl and trowel all occur in the English dictionary in alphabetical, or lexicographic order, whereas they do not all occur in the Lyndon or co-Lyndon dictionaries:

(i) fowl, growl, howl are each Lyndon and satisfy the Flight Deck condition.
(ii) owl, goose, oriole are each co-Lyndon and while they do not satisfy the Flight Deck condition, the co-Lyndon circ-UMFF satisfies the Acrobat condition, for instance owl $<_{co-\mathcal{L}}$ goose $<_{co-\mathcal{L}}$ oriole.
(iii) scowl, trowel are neither Lyndon nor co-Lyndon.

Note that if $\Sigma^*_{\mathcal{L}}$ denotes the lexicographic ordering of $\Sigma^*$, then the Lyndon total order is a sub-order of $\Sigma^*_{\mathcal{L}}$.

We now consider the partition of the Lyndon circ-UMFF into those words which are the unique concatenation of exactly two smaller non-overlapping Lyndon words, and those words which do contain overlapping Lyndon words as in Lemma 2.1. For example, over the ordered Roman alphabet, the Lyndon word *abac* contains the unique pair of Lyndon words *ab* and *ac*. Similarly *abababababc* and *abbbbbbbbbbbb* both comprise unique concatenations, whereas the Lyndon word *abcdefg* contains many overlapping Lyndon words such as *abcde* and *bcdefg*.

**Theorem 3.1.** Suppose that $\boldsymbol{u} = \boldsymbol{u}[1..m]$, $\boldsymbol{v}[1..n]$, and $\boldsymbol{w} = \boldsymbol{uv}$ are Lyndon words. Suppose further that for every factorization of $\boldsymbol{w}$ of the form $\boldsymbol{w} = \boldsymbol{u'v'}$, $\boldsymbol{u'} \neq \boldsymbol{u}$ and $\boldsymbol{u'}$, $\boldsymbol{v'}$ both nonempty, at least one of $\boldsymbol{u'}$, $\boldsymbol{v'}$ is non-Lyndon. Then $\boldsymbol{w}$ must take one of the following forms:

(1) If $n = 1$, then $\boldsymbol{w} = \mu\boldsymbol{u}[2..m]\lambda$, where $\mu$ and $\lambda$ are letters satisfying $\mu < \lambda \leq \boldsymbol{u}[i]$, for every $i \in 2..m$.

(2) If $n > 1$, then $\boldsymbol{w} = \boldsymbol{u}^k\boldsymbol{u_1}\lambda$, where $k$ is a positive integer, $\boldsymbol{u_1}$ a possibly empty proper prefix of $\boldsymbol{u}$, and the letter $\lambda > \boldsymbol{u}[|\boldsymbol{u_1}|+1]$.

**Proof:**
Suppose $n = 1$ and let $\mu = \boldsymbol{u}[1]$, $\lambda = \boldsymbol{v}$. Since $\boldsymbol{uv} \in \mathcal{L}$, applying Lemma 2.3 we have $\mu < \lambda$, and so if $m = 1$, (1) is proved. For $m > 1$, since $\mu \in \mathcal{L}$ we have $\boldsymbol{u}[2..m]\lambda \notin \mathcal{L}$. For $m = 2$, $\lambda \leq \boldsymbol{u}[2]$, otherwise $\boldsymbol{u}[2]\lambda \in \mathcal{L}$, which is a contradiction; hence (1) holds. For $m > 2$, since $\mu < \lambda \leq \boldsymbol{u}[2]$, it follows that $\boldsymbol{u}[1..2] \in \mathcal{L}$, hence that $\boldsymbol{u}[3..m]\lambda \notin \mathcal{L}$. Similarly, for $m = 3$, $\lambda \leq \boldsymbol{u}[3]$, again establishing (1). Continuing this analysis yields (1) for all finite $m$.

Suppose $n > 1$, and let $\lambda = \boldsymbol{v}[n]$. Since $\boldsymbol{uv} \in \mathcal{L}$, by Lemma 2.3 we have $\lambda > \boldsymbol{u}[1]$. Further, since $\lambda \in \mathcal{L}$ then $\boldsymbol{uv}[1..n-1] \notin \mathcal{L}$. From these observations we deduce that $\boldsymbol{u} = \boldsymbol{v}[i]$ for $i \in 1..n-1$, and (2) holds when $m = 1$. Suppose $m \geq 1$. Then since $\lambda \in \mathcal{L}$, $\boldsymbol{uv}[1..n-1] \notin \mathcal{L}$ and since $\boldsymbol{u} \in \mathcal{L}$ we deduce that $\boldsymbol{v}[1] \leq \boldsymbol{u}[1]$. However, $\boldsymbol{uv} \in \mathcal{L}$ implies $\boldsymbol{u}[1] \leq \boldsymbol{v}[1]$, and so $\boldsymbol{v}[1] = \boldsymbol{u}[1]$. Since $\lambda > \boldsymbol{u}[1]$ this establishes (2) for $m = 1$ and $n = 2$; since $\boldsymbol{v}[1] = \boldsymbol{u}[1]$ then applying Theorem 2.2(1) to $\boldsymbol{uv}$ we have $\lambda > \boldsymbol{u}[2]$ which establishes (2) for $m > 1$ and $n = 2$.

For $m > 1$ and $n > 2$, it is required that $\boldsymbol{uu}[1]\boldsymbol{v}[2..n-1] \notin \mathcal{L}$. Thus $\boldsymbol{v}[2] \leq \boldsymbol{u}[2]$, while $\boldsymbol{uv} \in \mathcal{L}$ implies $\boldsymbol{v}[2] \geq \boldsymbol{u}[2]$, so that $\boldsymbol{v}[2] = \boldsymbol{u}[2]$. Applying Theorem 2.2(1) to $\boldsymbol{uv}$ we have $\lambda > \boldsymbol{u}[3]$ establishing (2) for $n = 3$. (Note that if $m = 1$ and $n > 2$, then $\boldsymbol{w} = \boldsymbol{u}^{m+n-1}\lambda = \boldsymbol{u}^n\lambda$.)

Proceeding with this analysis yields (2) for all finite $m$ and $n > 1$. □

We conclude by generalizing the lexicographic order $<$ of strings (defined in Section 2) to the lexicographic order $\ll$ of Lyndon factorizations of strings. Suppose two strings $\boldsymbol{u}$ and $\boldsymbol{v}$ happen to be equal, then obviously so are their Lyndon factorizations, that is $\boldsymbol{u} = \boldsymbol{v} \iff F_{\mathcal{L}}(\boldsymbol{u}) = F_{\mathcal{L}}(\boldsymbol{v})$. If $\boldsymbol{u} < \boldsymbol{v}$, then recall that in lexorder there are two cases: $\boldsymbol{u}$ could be a proper prefix of $\boldsymbol{v}$ ($\boldsymbol{u} <_{pref} \boldsymbol{v}$), or $\boldsymbol{u}$ is not a prefix of $\boldsymbol{v}$ and there is a first difference occuring between letters in $\boldsymbol{u}$ and $\boldsymbol{v}$ ($\boldsymbol{u} <_{diff} \boldsymbol{v}$). We now define lexorder $\ll$ of factorizations.

**Definition 3.1.** Let $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^+$ with respective Lyndon factorizations $F_{\mathcal{L}}(\boldsymbol{u}) = \boldsymbol{u_1}\boldsymbol{u_2}\cdots\boldsymbol{u_r}$ and $F_{\mathcal{L}}(\boldsymbol{v}) = \boldsymbol{v_1}\boldsymbol{v_2}\cdots\boldsymbol{v_s}$. Then
(1) $F_{\mathcal{L}}(\boldsymbol{u}) \ll_{pref} F_{\mathcal{L}}(\boldsymbol{v})$ means that either $\boldsymbol{u_i} = \boldsymbol{v_i}$ for $1 \leq i \leq r$ and $r < s$, or for some least $i \leq min\{r, s\}$, $\boldsymbol{u_i} \neq \boldsymbol{v_i}$ and $\boldsymbol{u_i}\boldsymbol{u_{i+1}}...\boldsymbol{u_r} <_{pref} \boldsymbol{v_i}$.
(2) $F_{\mathcal{L}}(\boldsymbol{u}) \ll_{diff} F_{\mathcal{L}}(\boldsymbol{v})$ means that for some least $i \leq min\{r, s\}$, $\boldsymbol{u_i} \neq \boldsymbol{v_i}$ and $\boldsymbol{u_i} <_{diff} \boldsymbol{v_i}$.

We can then relate the lexorder $<$ of distinct strings to the lexorder $\ll$ of their factorizations.

**Proposition 3.1.** Let $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^+$ where $\boldsymbol{u} < \boldsymbol{v}$ in lexorder, with respective Lyndon factorizations $F_{\mathcal{L}}(\boldsymbol{u})$, $F_{\mathcal{L}}(\boldsymbol{v})$. Then
(1) $\boldsymbol{u} <_{pref} \boldsymbol{v}$ if and only if $F_{\mathcal{L}}(\boldsymbol{u}) \ll_{pref} F_{\mathcal{L}}(\boldsymbol{v})$.
(2) $\boldsymbol{u} <_{diff} \boldsymbol{v}$ if and only if $F_{\mathcal{L}}(\boldsymbol{u}) \ll_{diff} F_{\mathcal{L}}(\boldsymbol{v})$.

**Proof:**
In both cases necessity is by definition of the lexorder $\ll$ of factorizations, and sufficiency is by definition of the lexorder $<$ of strings. □

# 4. Problems

Consider the well-known sequence of Fibonacci strings, where commencing with the Fibonacci strings $b$ and $a$, strings with greater than unit length are the concatenation of the previous two: $b,\ a,\ ab,\ aba,\ abaab,$ $abaababa, \ldots$ (these strings are also known as *finite Fibonacci words*; see [BMP-07], [IMS-98], [Lu-95] for related works on Fibonacci strings). A simple application of Lemma 2.1 to the pair of strings $aba,\ abaab$ falsely implies that the string $ababaab$ is Fibonacci. Thus although Fibonacci strings form a factorization family (FF), they do not yield unique factorization, and in fact there are many ways to factor the string $ababaab$ into Fibonacci strings: $(ab)(aba)(ab)$, and $(ab)(abaab)$, also $(ab)(ab)(a)(a)(b)$, etc.

In the quest for more examples and properties of factorization families, we propose the following lines of enquiry:

1. Commencing with the study of border-free UMFFs, describe the structural properties of all UMFFs.
2. Apply the inherent construction of Theorem 2.2 to design algorithms both for constructing all circ-UMFFs, and all binary circ-UMFFs.
3. Design generic algorithms for factoring strings over general, Flight Deck and Acrobat circ-UMFFs.
4. Establish whether or not all circ-UMFFs on the same alphabet are in some sense isomorphic.
5. Given a string $\boldsymbol{u}$, determine the circ-UMFF(s) which factorizes $\boldsymbol{u}$ into the maximal or minimal number of factors. For example, if $\lambda \in \Sigma$ then the repetition $\boldsymbol{\lambda^k}$ has $k$ factors over any circ-UMFF. However, the string $dcba$ over $\{a < b < c < d\}$ can be factored into one co-Lyndon or four Lyndon words.

**Acknowledgements**
We warmly thank the referees for their very helpful comments and corrections which improved the quality of this paper.

# References

[BMP-07] S. Brlek and G. Melançon and G. Paquin, Properties of the extremal infinite smooth words, *Discrete Math. Theor. Comput. Sci.* **9** : **2** (2007) 33-50.

[C-04] M. Chemillier, Periodic musical sequences and Lyndon words, *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, Springer-Verlag, ISSN 1432-7643 (Print) 1433-7479 (Online), Vol. 8, Issue 9 (September 2004) 611-616.

[CDP-05] M. Crochemore, J. Désarménien and D. Perrin, A note on the Burrows-Wheeler transformation, *Theoret. Comput. Sci.* **332** (1-3) (2005) 567-572.

[CFL-58] K.T. Chen, R.H. Fox and R.C. Lyndon, Free differential calculus, IV - The quotient groups of the lower central series, *Ann. Math.* **68** (1958) 81-95.

[D-08] D.E. Daykin, A $2n$ algorithm factors an $n$-string into Lyndon words, to appear in *J. Discrete Algorithms*.

[DD-03] D.E. Daykin and J.W. Daykin, Lyndon-like and V-order factorizations of strings, *J. Discrete Algorithms* **1** (2003) 357-365.

[DD-08] D.E. Daykin and J.W. Daykin, Properties and construction of unique maximal factorization families for strings, *Internat. J. Found. Comput. Sci.* Vol. 19, No. 4 (2008) 1073-1084.

[DIS-94] J.W. Daykin, C.S. Iliopoulos and W.F. Smyth, Parallel RAM algorithms for factorizing words, *Theoret. Comput. Sci.* **127** (1) (1994) 53-67.

[Du-83] J.P. Duval, Factorizing words over an ordered alphabet, *J. Algorithms* **4** (1983) 363-381.

[IMS-98] C.S. Iliopoulos, D. Moore and W.F. Smyth, The covers of a circular Fibonacci string, *J. Combin. Math. Combin. Comput.* **26** (1998) 227-236.

[IS-92] C.S. Iliopoulos and W.F. Smyth, Optimal algorithms for computing the canonical form of a circular string, *Theoret. Comput. Sci.* **92** (1) (1992) 87-105.

[KS-98] D.L. Kreher and D.R. Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*, CRC Press (1998).

[L-83] M. Lothaire, *Combinatorics on Words*, Addison-Wesley, Reading, MA, 1983; 2nd Edition, Cambridge University Press, Cambridge, 1997.

[Lu-95] A. de Luca, A division property of the Fibonacci Word, *Information Processing Letters* **54** (6) (1995) 307-312.

[P-05] L. Perret, A chosen ciphertext attack on a public key cryptosystem based on Lyndon words, Proceedings of International Workshop on Coding and Cryptography (WCC 2005), (January 2005) 235-244.

[S-03] Bill Smyth, *Computing patterns in strings*, Pearson (2003).