Combinatorics on 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, jwd@dcs.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

December 18, 2008

Abstract

Suppose a set $\mathcal{W}$ of strings contains exactly one rotation (cyclic shift) of every primitive string on some alphabet Σ. Then $\mathcal{W}$ is a circ-UMFF iff every word in Σ^+ has a unique maximal factorization over $\mathcal{W}$. The classic circ-UMFF is the set of Lyndon words based on lexicographic ordering. Duval designed a linear sequential Lyndon factorization algorithm; a corresponding PRAM parallel algorithm was described by J. Daykin, Iliopoulos and Smyth. Daykin and Daykin defined new circ-UMFFs based on various methods for totally ordering sets of strings, and further described the structure of all circ-UMFFs. 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.

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

1 Introduction

This paper is concerned with concatenating and factoring strings over specified alphabets. We are interested in analyzing and constructing sets of strings which are closed under the reciprocal operations of concatenating and factoring. We consider cases where, given an instance of a string and a set of strings, the string either belongs to the set or can be factored uniquely into longest strings belonging to the set. We therefore call these sets Unique Maximal Factorization Families (UMFFs). In particular, we consider ***circ-UMFFs*** — that is, UMFFs that contain exactly one rotation of every primitive string on the given alphabet.

We believe that the set of Lyndon words was the first example of a circ-UMFF [CFL-58, L-83] (although Lyndon factorization was originally introduced for computing free monoids in Lie algebras). However the subsequent importance of Lyndon factorization is expressed by the 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 factorization.

The notion of order is inherent both in the definition of Lyndon words, which involves lexicographic ordering, and in the concatenation operation. Daykin and Daykin found additional factorization families each based on methods for totally ordering sets of strings [DD-03]. They then established fundamental properties, independent of techniques for ordering strings, related to concatenating and factoring words in circ-UMFFs [DD-08]. The existence of distinct circ-UMFFs means that, for example, substrings can be re-factored allowing for string theoretic operations on the substrings.

In this paper we establish new combinatorial properties of factorization families, for instance on the ordering of prefixes and suffixes of factors. We show that maximal factors in a factorization over any UMFF cannot be overlapping. 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-UMFF, 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 the characterization by Daykin and Daykin of circ-UMFFs for the particular case of Lyndon words, for instance showing that any Lyndon word (which is not a letter) can be partitioned into two ordered Lyndons. We compare the Lyndon circ-UMFF to the symmetric co-Lyndon circ-UMFF, which is based on a simple modification of lexicographic ordering. As all circ-UMFFs are totally ordered sets of strings, we compare and contrast 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 Lyndons.

Our exposition commences 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.

2 Unique Maximal Factorization Families (UMFFs)

Given an integer $n \geq 1$ and a set Σ, $\boldsymbol{x} = \boldsymbol{x}[1..n]$ is a ***string*** of ***length*** n on Σ iff for every $i \in 1..n$, $\boldsymbol{x}[i] \in \Sigma$. We also write $n = |\boldsymbol{x}|$. Σ is called the ***alphabet*** and Σ^+ denotes the set of all strings on Σ. 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 a ***factor*** of $\boldsymbol{x}[1..n]$ iff $\boldsymbol{w} = \boldsymbol{x}[i..j]$ 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 $\boldsymbol{x}$; if every factor $\boldsymbol{w_j}$, $j \in 1..k$, belongs to a set $\mathcal{W}$, then the factorization is denoted $F_{\mathcal{W}}(\boldsymbol{x})$.

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 Σ there exists a factorization $F_{\mathcal{W}}(\boldsymbol{x})$.*

Observe that every FF must contain Σ; moreover, every subset of Σ^+ containing Σ 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 exist no suffix $\boldsymbol{u'}$ of $\boldsymbol{u}$ and 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 Σ. Then $\mathcal{W}$ is a **unique maximal factorization family** (UMFF) if and only if there exists a max factorization $F_{\mathcal{W}}(\boldsymbol{x})$ for* every *string $\boldsymbol{x} \in \Sigma^+$.*

Observe that Σ is an UMFF, further that the definition of UMFF does not require that Σ be ordered. The following result is a characterization of UMFFs, and we introduce a new proof of this lemma here.

Lemma 2.3 *(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 $\boldsymbol{xyz} \in \mathcal{W}$.*

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 $\boldsymbol{w'_k = uyz}$ 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 $\boldsymbol{xyz} \in \mathcal{W}$.

Next suppose that for some FF $\mathcal{W}$ and for all strings $\boldsymbol{x}$, $\boldsymbol{y} \neq \boldsymbol{\varepsilon}$ and $\boldsymbol{z}$ such that $\boldsymbol{xy}, \boldsymbol{yz} \in \mathcal{W}$, it follows that $\boldsymbol{xyz} \in \mathcal{W}$. We need to show that the factorization of every string $\boldsymbol{v} = \boldsymbol{v}[1..n]$ is max. 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 $\boldsymbol{w_2} = \boldsymbol{v}[i_1{+}1..i_2] \in \mathcal{W}$. Suppose there exists $i < i_1$ such that $\boldsymbol{x} = \boldsymbol{v}[1..i] \in \mathcal{W}$ and $\boldsymbol{z} = \boldsymbol{v}[i{+}1..i'_2] \in \mathcal{W}$ for some $i'_2 > i_1$. Then taking $\boldsymbol{y} = \boldsymbol{v}[i{+}1..i_1]$ and applying the $\boldsymbol{xyz}$ condition, we find that $\boldsymbol{v}[1..i'_2] \in \mathcal{W}$, contradicting the maximality of i_1. Thus no such i'_2 exists, and $\boldsymbol{w_2}$ is max on the left. If $i_2 = n$, $\boldsymbol{w_2}$ is also max on the right, and so the factorization is max. If not, we continue left to right across $\boldsymbol{v}$, at each step adding factors that are max on the right and using the $\boldsymbol{xyz}$ condition to check that each factor is also max on the left. After a finite number of steps, this process yields a max factorization of $\boldsymbol{v}$, and so $\mathcal{W}$ is an UMFF. □

It is an immediate consequence of Lemma 2.3 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 2.4 *Suppose* $\boldsymbol{x = u_1 u_2 \cdots u_m}$ *and* $\mathcal{W}$ *is an UMFF, where for every* $j \in 1..m$, $\boldsymbol{u_j} \in \mathcal{W}$. *Then* $F_{\mathcal{W}}(\boldsymbol{x}) = \boldsymbol{w_1 w_2 \cdots w_k}$, *where*

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

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

Proof. Suppose that for some $i \in 1..k$, $\boldsymbol{w_i = u_{j_r+1} \cdots u_{j_{r+1}} u'_{j_{r+1}+1}}$, where $\boldsymbol{u'_{j_{r+1}+1}}$ is a nonempty prefix of $\boldsymbol{u_{j_{r+1}+1}}$. From Lemma 2.3 it follows that $\boldsymbol{u'_{j_{r+1}+1} = u_{j_{r+1}+1}}$. Similarly if we suppose $\boldsymbol{w_i}$ has a nonempty prefix $\boldsymbol{u'_{j_r}}$ that is a suffix of $\boldsymbol{u_{j_r}}$. □

Clearly this result has implication for the complexity analysis of factorization algorithms (see for example [DIS-94]).

If $\boldsymbol{x = uv}$, 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*** iff 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.

Definition 2.5 *An UMFF $\mathcal{W}$ over Σ^+ is a* ***circ-UMFF*** [1] *if and only if it contains exactly one rotation of every primitive string $\boldsymbol{x} \in \Sigma^+$.*

If Σ 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 $\boldsymbol{u} = \boldsymbol{ras}$, $\boldsymbol{v} = \boldsymbol{rbt}$ for some $a, b \in \Sigma$ such that $a < b$ and for some $\boldsymbol{rst} \in \Sigma^*$. 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 Σ. (Note that the choices of rotations for the words of length two for a circ-UMFF in fact induce a total order on a given unordered alphabet, see [DD-08].) Consider the following selection of Lyndons based on different orderings of letters in the alphabet.

Example 2.6 *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 2.7 *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 choose 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 2.6, in the case that b is the least letter, the rule in Example 2.7, with the order for 'least' being lexorder, leads to the choice of $R_3(\boldsymbol{x}) = acaab$ for a new circ-UMFF ***co-Lyndon*** (**co-$\mathcal{L}$**). We call the ordering based on lexorder of reversed strings ***co-lexorder*** [2]. So for example, over the Roman alphabet the word *google*, although not a Lyndon is a co-Lyndon, 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 Σ^+.

Definition 2.8 *If a circ-UMFF $\mathcal{W}$ contains strings $\boldsymbol{u}$, $\boldsymbol{v}$ and $\boldsymbol{uv}$, we say that $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$ ($\mathcal{W}$-**order**).*

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. For the Lyndon circ-UMFF, its specific order is lexorder, as we see by

[1] circ-UMFFs were originally defined with respect to circulant matrices in [DD-08]; here we adopt the equivalent terminology of rotations.
[2] Other definitions exist in the literature.

Lemma 2.9 *(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 Lemma 2.9 does not hold for every circ-UMFF. That is, if the elements of Σ^* 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 co-$\mathcal{L}$. The primitive words $\boldsymbol{u} = cba$ and $\boldsymbol{v} = cbba$ are clearly co-Lyndons over the Roman alphabet. Analysis of all the rotations of $\boldsymbol{uv}$ shows that it is co-Lyndon, and by Definition 2.8 we have $\boldsymbol{u} <_{\text{co-}\mathcal{L}} \boldsymbol{v}$. However, $\boldsymbol{v} <_{\text{co-lex}} \boldsymbol{u}$! In other words, $\mathcal{W}$-order can be defined quite independently of the ordering of the elements of Σ^*.

The following characterization reveals structural properties of circ-UMFFs that prescribe ordered concatenating and factoring of words. The theorem also shows that not every rotation of a primitive string can necessarily be chosen to belong to a circ-UMFF. Recall that a ***border*** of a string $\boldsymbol{x}$ is a nonempty prefix of $\boldsymbol{x}$ that is also a suffix of $\boldsymbol{x}$.

Theorem 2.10 (DD-08) *Let $\mathcal{W}$ be a circ-UMFF, and for every positive integer d let $\mathcal{W}(d) = \{\boldsymbol{x} \in \mathcal{W}, |\boldsymbol{x}| \leq d\}$.*
(1) If $\boldsymbol{u} \in \mathcal{W}(d)$ then $\boldsymbol{u}$ is border-free.
(2) If $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}(d)$ and $\boldsymbol{u} \neq \boldsymbol{v}$ then $\boldsymbol{uv}$ is primitive.
(3) If $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}(d)$ and $\boldsymbol{u} \neq \boldsymbol{v}$ then $\boldsymbol{uv} \in \mathcal{W}$ or $\boldsymbol{vu} \in \mathcal{W}$ (but not both).
(4) If $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}(d)$ and $\boldsymbol{uv} \in \mathcal{W}$ (so $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$), then $<_{\mathcal{W}}$ is a total order of $\mathcal{W}(d)$.
(5) If $\boldsymbol{w} \in \mathcal{W}(d+1)$, $|\boldsymbol{w}| \geq 2$, then there exist $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{W}(d)$ 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: $\boldsymbol{u} = \boldsymbol{v}$, $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{v}$, $\boldsymbol{v} <_{\mathcal{W}} \boldsymbol{u}$. Also, in Example 2.6, part (1) of this theorem tells us that $R_1(\boldsymbol{x}) = abaca$, 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):

Lemma 2.11 *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 2.10(1). The proof for suffix $\boldsymbol{u}$ is analogous. □

For the remainder of this section we demonstrate various applications of Theorem 2.10 giving new combinatorial insights into circ-UMFFs.

Lemma 2.12 *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 2.10(3) and Definition 2.8; necessity is Theorem 2.10(5). □

Then the following result, modified from [DD-08], is easily established, which generalizes the Lyndon factorization theorem [CFL-58] to circ-UMFFs. Compare Corollary 2.4.

Lemma 2.13 *Suppose $\boldsymbol{x} = \boldsymbol{u_1 u_2 \cdots u_m}$ and $\mathcal{W}$ is a circ-UMFF, where for every $j \in 1..m$, $\boldsymbol{u_j} \in \mathcal{W}$. 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 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 2.3 and Theorem 2.10.

Lemma 2.14 *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 2.3 and Theorem 2.10(1),(2), and (3). □

We show next that the case $\boldsymbol{xyyz} <_{\mathcal{W}} \boldsymbol{xyz}$ of Lemma 2.14(3) occurs for the Lyndon circ-UMFF based on lexicographic ordering.

Lemma 2.15 *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 2.14, 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 2.16 *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 2.10(1),(3) neither of the bordered strings $\boldsymbol{wu}$ and $\boldsymbol{vw}$ can be an element of $\mathcal{W}$, it follows from Definition 2.8 and Theorem 2.10(4) that $\boldsymbol{u} <_{\mathcal{W}} \boldsymbol{w} <_{\mathcal{W}} \boldsymbol{v}$. □

In particular, this 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 $\boldsymbol{w}[n] = \boldsymbol{w}[1]$, $\boldsymbol{w} \notin \mathcal{W}$. The following is an immediate consequence of Lemma 2.16:

Lemma 2.17 *[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 or suffix of $\boldsymbol{w} \in \mathcal{L}$, whether themselves in $\mathcal{L}$ or not [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 2.18 *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 2.16 $\boldsymbol{v} <_{\mathcal{W}} \boldsymbol{v}$, a contradiction. Thus $\boldsymbol{u} \neq \boldsymbol{vw}$, so that by Lemma 2.10(3) either $\boldsymbol{uvw}$ or $\boldsymbol{vwu}$ is in $\mathcal{W}$. If $\boldsymbol{vwu} \in \mathcal{W}$, Lemma 2.16 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 2.19 (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 $\boldsymbol{u_1}^{k_1} \boldsymbol{u_2}^{k_2} ... \boldsymbol{u_m}^{k_m} \in \mathcal{W}$.*

Further, we can generate subsequences from given circ-UMFF factors (we include a simple inductive proof for the case of letters):

Lemma 2.20 *Suppose a circ-UMFF $\mathcal{W}$ contains strings $\boldsymbol{u_i}$, $i = 1, 2, \ldots, m$, satisfying*

$$\boldsymbol{u_1} <_{\mathcal{W}} \boldsymbol{u_2} <_{\mathcal{W}} \cdots <_{\mathcal{W}} \boldsymbol{u_m}$$

in $\mathcal{W}$-order. Then for $r \in 1..m$ such that $1 \leq i_1 < i_2 < \cdots < i_r \leq m$,

$$\boldsymbol{w_r} = \boldsymbol{u_{i_1} u_{i_2} \cdots u_{i_r}} \in \mathcal{W}.$$

Proof. Assume all $\boldsymbol{u_i}$ are letters. For $r = 1$, the result is trivial. For $r = 2$, it follows from Definition 2.8 of $\mathcal{W}$-order. For some $r \in 3..m$, consider $\boldsymbol{w_r} = \lambda_{i_1}\lambda_{i_2} \cdots \lambda_{i_r}$ and let $\boldsymbol{y} = \lambda_{i_2}\lambda_{i_3} \cdots \lambda_{i_{r-1}}$. If we suppose the lemma holds for r–1, it follows that both $\lambda_{i_1}\boldsymbol{y}$ and $\boldsymbol{y}\lambda_{i_r}$ are elements of $\mathcal{W}$. But then by Lemma 2.3, $\boldsymbol{w_r} \in \mathcal{W}$. Thus for $r \in 3..m$, the result follows by induction. □

Applying Lemma 2.19 and Definition 2.8 to a new subsequence then yields circ-UMFF factors of the form $\boldsymbol{u_{i_1}^{k_1} u_{i_2}^{k_2} \cdots u_{i_r}^{k_r}}$, and so on.

Taken together with Definition 2.8 and Lemma 2.16, Lemma 2.20 enables us to order some collections of strings: for $r \in 1..|\Sigma|$ such that $1 \leq i_1 < i_2 < \cdots < i_r \leq |\Sigma|$,

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

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 $01 <_{\mathcal{W}} 0011$.

We go on to explore "dictionary" type properties of circ-UMFFs, showing that some orders of concatenations are predetermined.

Lemma 2.21 *Suppose $\mathcal{W}$ is a circ-UMFF defined on $\Sigma = \{\lambda_1, \lambda_2, \ldots.\}$, with $\boldsymbol{u} \in \Sigma^+$.*
(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. (1),(2),(3) are derived from Definition 2.8 and Theorem 2.10, (4) is a special case of Lemma 2.17, (5) a special case of Lemma 2.18. □

By contrast, choice for concatenation arises in certain contexts. For instance, even if $\lambda_i <_{\mathcal{W}} \lambda_j$ as above, then for some non-empty $\boldsymbol{u}$, 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:

Lemma 2.22 *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 2.10(3) to $\lambda_i\boldsymbol{u}$ and $\lambda_j\boldsymbol{v}$, suppose that $\lambda_j\boldsymbol{v}\lambda_i\boldsymbol{u} \in \mathcal{W}$. Applying Lemma 2.3 to $\lambda_j\boldsymbol{v}\lambda_i\boldsymbol{u}$ and $\lambda_i\boldsymbol{u}\lambda_j$ yields the bordered string $\lambda_j\boldsymbol{v}\lambda_i\boldsymbol{u}\lambda_j \in \mathcal{W}$, a contradiction. Thus $\lambda_i\boldsymbol{u}\lambda_j\boldsymbol{v} \in \mathcal{W}$, and the result follows. □

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 classify circ-UMFFs into type Flight Deck or type Acrobat according to certain $\mathcal{W}$-order properties as follows:

Definition 2.23 *A circ-UMFF $\mathcal{W}$ is said to be **Type Flight Deck** iff $\boldsymbol{w}[1...n] \in \mathcal{W}$ with length at least two implies that for every $i \in 2..n$, $\boldsymbol{w}[1] \leq_{\mathcal{W}} \boldsymbol{w}[i]$.*

Definition 2.24 *A circ-UMFF $\mathcal{W}$ is said to be **Type Acrobat** iff it contains elements $\boldsymbol{uv_1}$, $\boldsymbol{w}$ and $\boldsymbol{uv_2}$, nonempty $\boldsymbol{u}$ not a prefix of $\boldsymbol{w}$, such that*

$$\boldsymbol{uv_1} <_{\mathcal{W}} \boldsymbol{w} <_{\mathcal{W}} \boldsymbol{uv_2}.$$

Suppose $\Sigma = \{a <_{\mathcal{W}} b <_{\mathcal{W}} c <_{\mathcal{W}} d\}$ for some $\mathcal{W}$-order. Then an example of an element chosen for a Flight Deck circ-UMFF over Σ is given by $\lambda_i\boldsymbol{u} = ac$ and $\lambda_j\boldsymbol{v} = bd$, so that $\lambda_i\boldsymbol{u}\lambda_j\boldsymbol{v} = acbd$. Whereas with $\lambda_j\boldsymbol{v}\lambda_i\boldsymbol{u} = bdac$, although the first letter is (always) less than the last, here the internal letter a is less than the 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 0 and end 1 and there are no other letters to consider in the alphabet), and the Lyndon circ-UMFF (no rotation, hence letter can be lexicographically less than the first letter). To show that the co-Lyndon circ-UMFF cannot be 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-Lyndons, the latter word 231 does not satisfy the Flight Deck condition since the second letter is less than the first in $\mathcal{W}$-order.

Lemma 2.25 *Suppose $\mathcal{W}$ is a Flight Deck circ-UMFF over Σ and the letter $\mu \in \Sigma$. Suppose $\boldsymbol{w} \in \mathcal{W}$ with length at least 2, and $\boldsymbol{w}$ includes the letter λ 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 2.10(3). (1) is then a consequence of Theorem 2.10(1) and the definition of Flight Deck; (2) follows similarly.

□

With reference again to Duval's observation that Lemma 2.17 holds for any prefix or suffix of Lyndons, we now compare $\mathcal{W}$-order of suffixes for the two types of circ-UMFFs, namely Flight Deck and Acrobat.

Lemma 2.26 *Suppose that $\boldsymbol{w} = \boldsymbol{uv}$ is an element of a circ-UMFF $\mathcal{W}$, $\boldsymbol{u}$ and $\boldsymbol{v}$ nonempty. Then either $\boldsymbol{wv} \in \mathcal{W}$ or else there exist $\boldsymbol{v_2} \in \mathcal{W}$ and nonempty $\boldsymbol{v_1}$ such that $\boldsymbol{v} = \boldsymbol{v_1 v_2}$ and $\boldsymbol{v_2 w v_1} \in \mathcal{W}$; in the latter case $\mathcal{W}$ is Type Acrobat.*

Proof. If $\boldsymbol{v} \in \mathcal{W}$, then since $\boldsymbol{v}$ and $\boldsymbol{w}$ are distinct, applying Theorem 2.10(3) either $\boldsymbol{wv}$ or $\boldsymbol{vw}$ is an element of $\mathcal{W}$; since $\boldsymbol{vw}$ is bordered, it follows from Theorem 2.10(1) that $\boldsymbol{vw} \notin \mathcal{W}$, thus that $\boldsymbol{wv} \in \mathcal{W}$. Furthermore, if $\boldsymbol{w}$ satisfies the Flight Deck condition, then clearly so does $\boldsymbol{wv}$. Hence we suppose that neither $\boldsymbol{v}$ nor $\boldsymbol{wv}$ is an element of $\mathcal{W}$.

Since $\boldsymbol{wv} \notin \mathcal{W}$, then by Definition 2.5, if $\boldsymbol{wv}$ is primitive then some rotation of $\boldsymbol{wv}$ must be in $\mathcal{W}$. So first we establish that $\boldsymbol{wv}$ is primitive and then we choose a rotation for $\mathcal{W}$.

Suppose that $\boldsymbol{wv}$ is periodic of period $p < |\boldsymbol{wv}|$. Therefore $\boldsymbol{wv} = \boldsymbol{t}^r \boldsymbol{t}^*$ where $|\boldsymbol{t}| = p$, $r \geq 1$, $\boldsymbol{t}^*$ is a proper prefix of $\boldsymbol{t}$ ($\boldsymbol{t}^* = \boldsymbol{\varepsilon}$ implies $r > 1$).
(i) Suppose $p \leq |\boldsymbol{v}|$. Then $r \geq 2$ and $\boldsymbol{w}$ is periodic of period p, a contradiction.
(ii) Suppose $p > |\boldsymbol{v}|$. If $r = 1$, then $\boldsymbol{wv} = \boldsymbol{uv}^2$ has a border of length $p' > 0$. If $p' \leq |\boldsymbol{v}|$, then $\boldsymbol{w} = \boldsymbol{uv}$ also has a border of length p', a contradiction. If $p' > |\boldsymbol{v}|$, then $\boldsymbol{w} = \boldsymbol{uv}$ has a border of length $p' - |\boldsymbol{v}|$, also a contradiction. Suppose $r > 1$. Then $p < |\boldsymbol{w}|$, so that $\boldsymbol{w}$ has period p, hence a border, a contradiction.
We conclude that $\boldsymbol{wv}$ is primitive, and proceed to choose a rotation for $\mathcal{W}$.

First suppose that the rotation $\overline{\boldsymbol{w}} = \boldsymbol{u_2 v^2 u_1} \in \mathcal{W}$ for nonempty $\boldsymbol{u_1}$, $\boldsymbol{u_2}$ such that $\boldsymbol{u} = \boldsymbol{u_1 u_2}$. But then applying Lemma 2.3 to $\boldsymbol{xy} = \overline{\boldsymbol{w}}$ and $\boldsymbol{yz} = \boldsymbol{u_1 u_2 v}$ implies that the bordered word $\boldsymbol{u_2 v^2 u_1 u_2 v} \in \mathcal{W}$, contradicting Theorem 2.10(1). Suppose then that the rotation $\overline{\boldsymbol{w}} = \boldsymbol{v_2 v u v_1} \in \mathcal{W}$. Similarly applying Lemma 2.3 to $\boldsymbol{xy} = \boldsymbol{uv_1 v_2}$ and $\boldsymbol{yz} = \overline{\boldsymbol{w}}$ implies that the bordered word $\boldsymbol{uv_1 v_2 v u v_1} \in \mathcal{W}$, again a contradiction. Likewise, the rotations $\overline{\boldsymbol{w}} = \boldsymbol{vvu}$ and $\overline{\boldsymbol{w}} = \boldsymbol{vuv}$ cannot belong to $\mathcal{W}$.

Thus we need only consider whether rotations of the form $\boldsymbol{v_2 u v v_1} \in \mathcal{W}$. Suppose so. Then by Theorem 2.10(5) we can split $\boldsymbol{v_2 u v v_1}$ into a pair of ordered factors, both of them in $\mathcal{W}$:

- Suppose $\boldsymbol{v_2 u_1} \in \mathcal{W}$, $\boldsymbol{u_2 v v_1} \in \mathcal{W}$ for some nonempty $\boldsymbol{u_1}$. But then applying Lemma 2.3 to $\boldsymbol{uv} = \boldsymbol{u_1 u_2 v_1 v_2}$ and $\boldsymbol{v_2 u_1}$, we find that the bordered word $\boldsymbol{u_1 u_2 v_1 v_2 u_1} \in \mathcal{W}$, a contradiction.

- Suppose $\boldsymbol{v_2 u v'} \in \mathcal{W}$, $\boldsymbol{v'' v_1} \in \mathcal{W}$ for some nonempty $\boldsymbol{v'}$ such that $\boldsymbol{v} = \boldsymbol{v' v''}$. (Assume $\boldsymbol{v''}$ is nonempty for otherwise $\boldsymbol{v_2 u v'}$ is bordered.) But

then applying Lemma 2.3 to $\boldsymbol{v_2uv'}$ and $\boldsymbol{uv} = \boldsymbol{uv'v''}$, we find that the bordered word $\boldsymbol{v_2uv} \in \mathcal{W}$, again a contradiction.

Thus the factorization 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 which case we have distinct $\boldsymbol{uv}$ and $\boldsymbol{v_2}$ both belonging to $\mathcal{W}$, and so applying Theorem 2.10(3),(1) we know $\boldsymbol{v_2uv} \notin \mathcal{W}$. Hence, also applying Theorem 2.10(4) we deduce that

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

so that $\mathcal{W}$ is Type Acrobat. Furthermore, since we have assumed that $\boldsymbol{v_2uvv_1} \in \mathcal{W}$, similarly applying Theorem 2.10 to all these distinct factors we see that

$$\boldsymbol{uv} <_{\mathcal{W}} \boldsymbol{v_2} <_{\mathcal{W}} \boldsymbol{v_2uvv_1} <_{\mathcal{W}} \boldsymbol{uvv_1},$$

so

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

again demonstrating that $\mathcal{W}$ is Type Acrobat.
(Note that with the remaining cases of splitting $\boldsymbol{v_2uvv_1}$ through $\boldsymbol{v_2}$ or $\boldsymbol{v_1}$ similarly yields a further Acrobat instance for the $\boldsymbol{v_2}$ case.) □

3 The Lyndon Dictionary

Here we illustrate parts (1)–(5) of Theorem 2.10 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 $<_{\mathcal{L}}$. Assume $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w} \in \mathcal{L}$ are distinct non-empty Lyndon words:

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

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

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

(4) Assume $\boldsymbol{u} < \boldsymbol{v}$ and $\boldsymbol{v} < \boldsymbol{w}$. Then $\boldsymbol{uv}$ and $\boldsymbol{vw}$ are both Lyndon. If the order is not total, so that $\boldsymbol{w} < \boldsymbol{u}$, then $\boldsymbol{wu} \in \mathcal{L}$. If now we apply Lemma 2.3 to $\boldsymbol{uv}$ and $\boldsymbol{vw}$, we find that $\boldsymbol{uvw} \in \mathcal{L}$, and similarly applying Lemma 2.3 to $\boldsymbol{vw}$ and $\boldsymbol{wu}$, implies that $\boldsymbol{vwu} \in \mathcal{L}$. Since $\boldsymbol{uvw}$ is Lyndon, the rotation $\boldsymbol{vwu}$ cannot be. 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 2.16 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 λ 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"*. Recall that the Lyndon circ-UMFF is of type Flight Deck (see Section 2). Then we compare a Lyndon dictionary, over the ordered Roman alphabet, to the usual English dictionary with the following example.

Example 3.1 *The words* fowl, growl, howl, owl, scowl *and* trowel *all occur in the English dictionary in alphabetical, or lexicographic order, whereas they do not all occur in the Lyndon dictionary:*
(i) fowl, growl, howl *are Lyndon,*
(ii) owl *is co-Lyndon,*
(iii) scowl, trowel *are neither Lyndon nor co-Lyndon.*

Let $\Sigma^*_{\mathcal{L}}$ denote the lexicographic ordering of Σ^*, 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 Lyndons as in Lemma 2.3. For example, over the ordered Roman alphabet, the Lyndon word *abac* contains the unique pair of Lyndons *ab* and *ac*. Similarly *abababac* and *abbbbbbbbbbbb* both comprise unique concatenations, whereas the Lyndon word *abcdefg* contains many overlapping Lyndons such as *abcde* and *bcdefg*.

Lemma 3.2 *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* $\boldsymbol{w} = \boldsymbol{u'v'}$, $\boldsymbol{u'} \neq \boldsymbol{u}$ *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 the letters* μ *and* λ *satisfy* $\mu < \lambda \leq \boldsymbol{u}[i]$, *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 2.16 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 $m = 2$, $\lambda \leq \boldsymbol{u}[2]$, thus establishing (1). 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 2.16 we have $\lambda > \boldsymbol{u}[1]$. Further, since $\lambda \in \mathcal{L}$ then $\boldsymbol{uv}[1..n-1] \notin \mathcal{L}$. From these 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 using $\lambda \in \mathcal{L}$, $\boldsymbol{uv}[1..n-1] \notin \mathcal{L}$ and $\boldsymbol{u} \in \mathcal{L}$ 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 2.10(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 2.10(1) to $\boldsymbol{uv}$ we have $\lambda > \boldsymbol{u}[3]$ establishing (2) for $n = 3$. (Note that if $m = 1$ and $n > 2$, then $\boldsymbol{w} = \boldsymbol{u}^{m+n-1}\lambda$.)

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 3.3 *Let $\boldsymbol{u}, \boldsymbol{v} \in \Sigma^+$ with respective Lyndon factorizations $F_{\mathcal{L}}(\boldsymbol{u}) = \boldsymbol{u_1 u_2 ... u_r}$ and $F_{\mathcal{L}}(\boldsymbol{v}) = \boldsymbol{v_1 v_2 ... v_s}$. Then*
(i) $F_{\mathcal{L}}(\boldsymbol{u}) \ll_{pref} F_{\mathcal{L}}(\boldsymbol{v})$ means that $\boldsymbol{u_i} = \boldsymbol{v_i}$ for $0 \leq i < r$ and $(\boldsymbol{u_{i+1} u_{i+2} ... u_r}) <_{pref} \boldsymbol{v_{i+1}}$.
(ii) $F_{\mathcal{L}}(\boldsymbol{u}) \ll_{diff} F_{\mathcal{L}}(\boldsymbol{v})$ means that there is a t in $1 \leq t \leq r, s$ and $\boldsymbol{u_i} = \boldsymbol{v_i}$ for $0 \leq i < t$ and $\boldsymbol{u_t} <_{diff} \boldsymbol{v_t}$.

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

Lemma 3.4 *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*
(i) $\boldsymbol{u} <_{pref} \boldsymbol{v}$ if and only if $F_{\mathcal{L}}(\boldsymbol{u}) \ll_{pref} F_{\mathcal{L}}(\boldsymbol{v})$,
(ii) $\boldsymbol{u} <_{diff} \boldsymbol{v}$ 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 (see [IMS-98]), where words with greater than unit length are the concatenation of the previous two: b, a, ab, aba, $abaab$, $abaababa, \ldots$. A simple application (although not unique) of Lemma 2.3 to the pair of words aba, $abaab$ falsely implies that the word $ababaab$ is Fibonacci. Thus Fibonacci words do not yield unique factorization, and in fact there are many ways to factorize the word $ababaab$ into Fibonacci words: $(ab)(aba)(ab)$, and $(ab)(abaab)$, also $(ab)(ab)(a)(a)(b)$, etcetera.

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, characterize all UMFFs.
2) Apply the inherent construction of Theorem 2.10 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 are isomorphic.
5) Given a string $\boldsymbol{u}$, determine the circ-UMFF(s) which factors $\boldsymbol{u}$ into the maximal or minimal number of factors. So 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.

References

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

[CFL-58] K.T. Chen, R.H. Fox and R.C. Lyndon, Free differential calculus, IV, *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. of Discrete Algorithms*.

[DD-03] D. E. Daykin and J. W. Daykin, Lyndon-like and V-order factorizations of strings, *J. of 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, *International Journal of the Foundations of Computer Science* 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. Comp. Sci.* **127** (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. Combinatorial Math. and Combinatorial Computing* **26** (1998) 227-236.

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

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

[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).