THEORIE-TAG 2023 A. Lin, G. Zetzsche (Eds.): Theorietag 2023, Kaiserslautern, 4. – 6.10.2023
RPTU & MPI-SWS Kaiserslautern, Technical Report, pp. 46–50.

Lyndon Partial Arrays

Meenakshi Paramasivan$^{(A)}$

$^{(A)}$Faculty of Business Administration, Economics, Social Sciences, Mathematics and Computer Sciences, Universität Trier, 54286, Trier, Germany

Abstract

Lyndon words have been extensively studied in different contexts of free Lie algebra and combinatorics. Lyndon partial words, arrays and trees have been recently introduced by us and we study on free monoid morphisms that preserve finite Lyndon partial words and check whether a morphism preserves or does not preserve the lexicographic order. We proposed an algorithm to determine Lyndon partial words of given length over the binary alphabet. Image analysis in several way of scanning via automata and grammars has a significance in two-dimensional models, we connect 2D Lyndon partial words with few automata and grammar models.

1. Introduction

Partial words are nothing but words with holes over the alphabet. The study of partial words was initiated by Berstel and Boasson [1] and later the work was extended by Blanchet Sadri [3, 4]. Lyndon words serve to be a useful tool for a variety of problems in combinatorics. There are many applications of Lyndon words in semigroups, pattern matching, representation theory of certain algebras and combinatorics such as they are used to describe the generators of the free Lie algebras. All of these applications make use of the combinatorial properties of Lyndon words, in particular the factorisation theorem. Their role in factorising a string over an ordered alphabet was initially illustrated by Chen et.al [5]. Duval [7] presented a algorithm to derive a factorisation of strings over an ordered alphabet known as Lyndon factorisation. Lyndon trees [6] are associated with Lyndon words under the name of standard lexicographic sequences. The Lyndon arrays[9, 2] of Lyndon words has recently become of interest since it could be used to efficiently compute all the maximal periodicities in a word. Both Lyndon and partial words have wide application in pattern matching. In [8], the authors have derived an automaton model namely Boustrophedon finite automata (BFA) for picture processing, which is equivalent to Regular matrix grammars (RMGs). The paper has the following organisation. In Section 2 we introduce Lyndon partial words and Lyndon partial arrays. A relation between the Lyndon partial words and trees is established. In Section 3 we characterise $\ell_{\Diamond}$-morphism and show that they are order-preserving morphism. In Section 4 we investigate few connections to 2D Lyndon words through 2D Lyndon partial words.

2. Lyndon Partial Words

Here we introduce and study the generalisation of finite Lyndon partial words by using trees. In [11], the authors have defined that a primitive partial word is a partial Lyndon word if and only if it is minimal in its conjugate class with respect to alphabetical order by assuming the order of $\Diamond$ as $\{a \prec b \prec \ldots \prec \Diamond\}$. The order of $\Diamond$ does not play a special role in the definition by studying properties of partial Lyndon words since the $\Diamond$ is considered as a letter with highest order which makes the definition similar to that of Lyndon words. In our definition of Lyndon partial word, the order of $\Diamond$ plays a special role in studying certain properties.

Definition 2.1 *A* Lyndon partial word $l_\Diamond = l_\Diamond[1\ldots n]$ *over the ordered alphabet* $\Sigma_{k,\Diamond} = \Sigma_k \bigcup \{\Diamond\} = \{a_1 \prec a_2 \prec \ldots \prec a_k\} \bigcup \{\Diamond\}, k > 1$ *is less than all its conjugates (rotations) with respect to the alphabetical order. Here the order of* $\Diamond$ *is considered as* $a_1 \preceq \Diamond$, $\Diamond \preceq a_k$ *and* $\Diamond$ *is compatible with all other elements of* Σ_k. *A* Lyndon partial language *over* Σ *is a subset of* $\Sigma_\Diamond^*$, *the set of all Lyndon partial words over* $\Sigma_\Diamond$.

For readability we use $L_\Diamond$ notation for partial languages which shall not be confused with the $\ell_\Diamond$ notation for Lyndon partial languages. Table 1 shows the set of all Lyndon partial words with length at most five over the ordered alphabet $\Sigma_\Diamond = \{a \prec b\} \bigcup \{\Diamond\}$.

Remark 2.2 *It is easy to observe that Lyndon partial words on binary alphabet takes the same integer sequence starting from 2, 3, 6, 9 by excluding the first three numbers namely 1, 2, 1 of that of Lyndon words as compared and evidenced in Table 1.*

Definition 2.3 *A Lyndon partial factor* $l_\Diamond[i\ldots j]$ *of a Lyndon partial word* $l_\Diamond[i\ldots n]$ *for any* $j \leq n$ *is* a maximal Lyndon partial factor *if it is Lyndon.*

Definition 2.4 *A* Lyndon partial array *(denoted as* $l_\Diamond^A$*) of* $l_\Diamond[1\ldots n]$ *is an array of integers in the range* $[1\ldots n]$ *such that, at each position* $i = 1\ldots n$ *stores the length of the longest Lyndon partial factor of* $l_\Diamond[1\ldots n]$ *starting at* i.

Example 2.5 *Consider a Lyndon partial word* $l_\Diamond[1\ldots 7] = aabab\Diamond b$*. The maximal Lyndon partial factor starting at position 1 is* $aabab$*, so* $l_\Diamond^A[1] = 5$*. The maximal Lyndon partial factor at position 2 is* ab*, so* $l_\Diamond^A[2] = 3$*. The maximal Lyndon partial factor starting at position 3 is* b*, so* $l_\Diamond^A[3] = 3$*. The maximal Lyndon partial factor starting at position 4 is* ab*, so* $l_\Diamond^A[4] = 5$*. The maximal Lyndon partial factor starting at position 5 is* b*, so* $l_\Diamond^A[1] = 5$*. The maximal Lyndon partial factor starting at position 6 is* $\Diamond b$*, so* $l_\Diamond^A[6] = 7$*. The maximal Lyndon partial factor starting at position 7 is* b*, so* $l_\Diamond^A[7] = 7$*. Therefore,* $l_\Diamond^A = [5\ \ 3\ \ 3\ \ 5\ \ 5\ \ 7\ \ 7]$.

Definition 2.6 *A* tree ζ associated with a Lyndon partial word *is described with its minimal among all of its rotations.* $\Im$ *denotes set of such trees. A sub-tree of* ζ *is a tree with set of nodes as a subset of* ζ.

Theorem 2.7 *No proper sub-tree exists as both initial and terminal of the tree* ζ.

Theorem 2.8 ζ *is a tree of a Lyndon partial word if and only if* $\zeta = P + vQ, v \in \delta(P)$ *where* $\zeta, P, Q \in \Im$ *and* $P \prec Q$.

Table 1: Lyndon words along with Lyndon partial words

Length	Lyndon words	Lyndon partial words
0	λ	-
1	a, b	-
2	ab	$a\Diamond, \Diamond b$
3	aab, abb	$aa\Diamond, a\Diamond b, \Diamond bb$
4	$aaab$, $aabb$, $abbb$	$aaa\Diamond, aa\Diamond b, a\Diamond ab, a\Diamond bb, ab\Diamond b, \Diamond bbb$
5	$aaaab, aaabb, aabab,$ $aabbb, ababb, abbbb$	$aaaa\Diamond, aaa\Diamond b, aa\Diamond ab, aa\Diamond bb, aab\Diamond b,$ $a\Diamond abb, a\Diamond bbb, ab\Diamond bb, \Diamond bbbb$
$\vdots$	$\cdots$	$\ddots$

Theorem 2.9 *Any tree ζ over the alphabet $\Sigma_{\Diamond}^{+}$ can be uniquely written as $\zeta = P_0 + v_1P_1 + v_2P_2 + \ldots .v_kP_k, v_m \in \delta(v_nP_n)$ for some $n \succeq m$ such that $P_0 \succeq P_1 \succeq P_2 \ldots . \succeq P_k$.*

3. $\ell_{\Diamond}$ - Morphism

In this section we characterise $\ell_{\Diamond}$-morphism and show that they are order-preserving morphism. A non-empty morphism g over an ordered alphabet $\Sigma_{k,\Diamond}$ containing atleast two letters is an order-preserving morphism if for all partial words $r_{\Diamond}, s_{\Diamond}$ over $\Sigma_{\Diamond}$, $r_{\Diamond} \prec s_{\Diamond} \Rightarrow g(r_{\Diamond}) \prec g(s_{\Diamond})$.

Definition 3.1 *Consider two ordered alphabets $U_{\Diamond}$ and $V_{\Diamond}$ each containing atleast two letters such that a morphism g from $U_{\Diamond}^{*}$ to $V_{\Diamond}^{*}$ is called a $\ell_{\Diamond}$-* morphism *if for any Lyndon partial words $l_{\Diamond}$ over $U_{\Diamond}$, $g(l_{\Diamond})$ is a Lyndon partial word over $V_{\Diamond}$. In short a morphism that preserves the property of Lyndon partial words is defined as $\ell_{\Diamond}$ - morphism.*

Theorem 3.2 *A non-empty morphism g on $\Sigma_{\Diamond}^{+}$ containing atleast two letters is a $\ell_{\Diamond}$- morphism if and only if g is an order preserving morphism such that for each $u_{\Diamond} \in \Sigma_{\Diamond}$, $g(u_{\Diamond})$ is a Lyndon partial word.*

Corollary 3.3 *g is a $\ell_{\Diamond}$- morphism on $\Sigma_{\Diamond} = \{a,b\} \cup \{\Diamond\}$ if and only if $g(a)$ and $g(b)$ are Lyndon partial words with $g(a) \prec g(b)$.*

4. Two-dimensional Lyndon partial words

The concept of Lyndon words are extended as two-dimensional Lyndon words in [10]. Those are useful to capture 2D horizontal periodicity of a matrix in which each row is highly periodic. It is also utilised to solve 2D horizontal suffix–prefix matching among a set of rectangular patterns efficiently. We introduce the following.

Definition 4.1 *A* two-dimensional row Lyndon partial word *is a horizontally primitive matrix which is least among its horizontal conjugates.*

Definition 4.2 *A* regular two-dimensional Lyndon partial word *is a horizontally primitive matrix which is least of its horizontal conjugates by maintaining a regular order.*

In one-dimensional case Lyndon partial words of length 4 over binary alphabet are $aaa\Diamond$, $aa\Diamond b$, $a\Diamond ab$, $a\Diamond bb$, $ab\Diamond b$, $\Diamond bbb$. Now we can derive two-dimensional partial words as follows where there will be many 2D partial words, few sample of those 2D partial words are given in Example below which maintains a specific/regular ordering of Lyndon partial arrays.

Example 4.3 $\begin{matrix} a & a & a & \Diamond \\ a & a & a & \Diamond \end{matrix}$, $\begin{matrix} a & a & a & \Diamond \\ a & a & \Diamond & b \end{matrix}$, $\begin{matrix} a & a & a & \Diamond \\ a & \Diamond & a & b \end{matrix}$, $\begin{matrix} a & a & a & \Diamond \\ a & \Diamond & b & b \end{matrix}$, $\begin{matrix} a & a & a & \Diamond \\ \Diamond & b & b & b \end{matrix}$, $\begin{matrix} a & a & \Diamond & b \\ a & a & \Diamond & b \end{matrix}$, $\begin{matrix} a & a & \Diamond & b \\ a & \Diamond & a & b \end{matrix}$, $\begin{matrix} a & a & \Diamond & b \\ a & \Diamond & b & b \end{matrix}$, $\begin{matrix} a & a & \Diamond & b \\ \Diamond & b & b & b \end{matrix}$, $\begin{matrix} a & \Diamond & a & b \\ a & \Diamond & a & b \end{matrix}$, $\begin{matrix} a & \Diamond & a & b \\ a & \Diamond & b & b \end{matrix}$, $\begin{matrix} a & \Diamond & a & b \\ \Diamond & b & b & b \end{matrix}$, $\begin{matrix} a & \Diamond & b & b \\ a & \Diamond & b & b \end{matrix}$, $\begin{matrix} a & \Diamond & b & b \\ \Diamond & b & b & b \end{matrix}$, $\begin{matrix} \Diamond & b & b & b \\ \Diamond & b & b & b \end{matrix}$, $\begin{matrix} a & a & a & \Diamond \\ a & a & a & \Diamond \\ a & a & a & \Diamond \end{matrix}, \cdots, \begin{matrix} \Diamond & b & b & b \\ \Diamond & b & b & b \\ \Diamond & b & b & b \end{matrix}, \begin{matrix} a & a & a & \Diamond \\ a & a & a & \Diamond \\ a & a & a & \Diamond \\ a & a & a & \Diamond \end{matrix}, \cdots, \begin{matrix} \Diamond & b & b & b \\ \Diamond & b & b & b \\ \Diamond & b & b & b \\ \Diamond & b & b & b \end{matrix}, \begin{matrix} a & a & a & \Diamond \\ a & a & a & \Diamond \\ a & a & a & \Diamond \\ a & a & a & \Diamond \\ a & a & a & \Diamond \end{matrix}, \cdots,$

$\begin{matrix} a & a & a & \Diamond \\ a & a & \Diamond & b \\ a & \Diamond & a & b \\ a & \Diamond & b & b \\ \Diamond & b & b & b \end{matrix}, \cdots, \begin{matrix} \Diamond & b & b & b \\ \Diamond & b & b & b \\ \Diamond & b & b & b \\ \Diamond & b & b & b \\ \Diamond & b & b & b \end{matrix}.$

One can observe that these 6 elements that follows are similar to many other elements which are NOT present in above collection: $\begin{matrix} \Diamond & a & a & a \\ b & a & a & \Diamond \\ b & a & \Diamond & a \\ b & a & \Diamond & b \\ b & \Diamond & b & b \end{matrix}, \begin{matrix} a & \Diamond & a & a \\ \Diamond & b & a & a \\ a & b & a & \Diamond \\ b & b & a & \Diamond \\ b & b & \Diamond & b \end{matrix}, \begin{matrix} a & a & \Diamond & b \\ a & \Diamond & a & b \\ a & \Diamond & b & b \\ \Diamond & b & b & b \\ a & a & a & \Diamond \end{matrix}, \begin{matrix} a & a & a & \Diamond \\ a & \Diamond & a & b \\ a & \Diamond & b & b \\ \Diamond & b & b & b \\ a & a & \Diamond & b \end{matrix}, \begin{matrix} a & a & a & \Diamond \\ a & a & \Diamond & b \\ a & \Diamond & b & b \\ \Diamond & b & b & b \\ a & \Diamond & a & b \end{matrix}, \begin{matrix} a & a & a & \Diamond \\ a & a & \Diamond & b \\ a & \Diamond & a & b \\ \Diamond & b & b & b \\ a & \Diamond & b & b \end{matrix}$ as these do not satisfy the property of being a regular 2D Lyndon partial word. Now it is of interest to identify specific/regular 2D Lyndon partial words among those elements. Due to Remark 2.2, we see the following pattern collected as a 2D partial language which shall be named as L_{DD} for further references in our work based on [12].

$$L_{DD} = \left\{ \begin{matrix} a & \Diamond \\ \Diamond & b \end{matrix}, \begin{matrix} a & a & \Diamond \\ a & \Diamond & b \\ \Diamond & b & b \end{matrix}, \begin{matrix} a & a & a & \Diamond \\ a & a & \Diamond & b \\ a & b & \Diamond & b \\ a & \Diamond & a & b \\ a & \Diamond & b & b \\ \Diamond & b & b & b \end{matrix}, \begin{matrix} a & a & a & a & \Diamond \\ a & a & a & \Diamond & b \\ a & a & b & \Diamond & b \\ a & a & \Diamond & a & b \\ a & a & \Diamond & b & b \\ a & b & \Diamond & b & b \\ a & \Diamond & a & b & b \\ a & \Diamond & b & b & b \\ \Diamond & b & b & b & b \end{matrix}, \begin{matrix} a & a & a & a & a & \Diamond \\ a & a & a & a & \Diamond & b \\ a & a & a & b & \Diamond & b \\ a & a & b & b & \Diamond & b \\ a & a & a & \Diamond & a & b \\ a & a & a & \Diamond & b & b \\ a & a & b & \Diamond & a & b \\ a & a & b & \Diamond & b & b \\ a & a & \Diamond & a & a & b \\ a & a & \Diamond & a & b & b \\ a & a & \Diamond & b & a & b \\ a & a & \Diamond & b & b & b \\ a & \Diamond & a & a & a & b \\ a & \Diamond & a & b & b & b \\ a & \Diamond & b & b & b & b \\ \Diamond & b & b & b & b & b \end{matrix}, \cdots, \right\}.$$

References

[1] J. BERSTEL, L. BOASSON, Partial Words and a Theorem of Fine and Wilf. *Theor. Comput. Sci.* **218** (1999) 1, 135–141.

[2] P. BILLE, J. ELLERT, J. FISCHER, I. L. GØRTZ, F. KURPICZ, J. I. MUNRO, E. ROTENBERG, Space Efficient Construction of Lyndon Arrays in Linear Time. In: A. CZUMAJ, A. DAWAR, E. MERELLI (eds.), *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*. LIPIcs 168, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 14:1–14:18.

[3] F. BLANCHET-SADRI, Primitive partial words. *Discret. Appl. Math.* **148** (2005) 3, 195–213.

[4] F. BLANCHET-SADRI, K. GOLDNER, A. SHACKLETON, Minimal partial languages and automata. *RAIRO Theor. Informatics Appl.* **51** (2017) 2, 99–119.

[5] K. T. CHEN, R. H. FOX, R. C. LYNDON, Free differential calculus, IV. The quotient groups of the lower central series. *Annals of Mathematics* (1958), 81–95.

[6] M. CROCHEMORE, L. M. RUSSO, Cartesian and Lyndon trees. *Theoretical Computer Science* **806** (2020), 1–9.

[7] J. DUVAL, Factorizing Words over an Ordered Alphabet. *J. Algorithms* **4** (1983) 4, 363–381.

[8] H. FERNAU, M. PARAMASIVAN, M. L. SCHMID, D. G. THOMAS, Simple picture processing based on finite automata and regular grammars. *J. Comput. Syst. Sci.* **95** (2018), 232–258.

[9] F. FRANEK, A. S. M. S. ISLAM, M. S. RAHMAN, W. F. SMYTH, Algorithms to Compute the Lyndon Array. In: J. HOLUB, J. ZDÁREK (eds.), *Proceedings of the Prague Stringology Conference 2016, Prague, Czech Republic, August 29-31, 2016*. Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, 2016, 172–184.

[10] S. MARCUS, D. SOKOL, 2D Lyndon Words and Applications. *Algorithmica* **77** (2017) 1, 116–133.

[11] A. C. NAYAK, K. KAPOOR, On the Language of Primitive Partial Words. In: A. DEDIU, E. FORMENTI, C. MARTÍN-VIDE, B. TRUTHE (eds.), *Language and Automata Theory and Applications - 9th International Conference, LATA 2015, Nice, France, March 2-6, 2015, Proceedings*. Lecture Notes in Computer Science 8977, Springer, 2015, 436–445.

[12] M. PARAMASIVAN, R. K. KUMARI, R. ARULPRAKASAM, V. R. DARE, Lyndon Partial Words and Arrays with Applications. In: R. P. BARNEVA, V. E. BRIMKOV, G. NORDO (eds.), *Combinatorial Image Analysis - 21st International Workshop, IWCIA 2022, Messina, Italy, July 13-15, 2022, Proceedings*. Lecture Notes in Computer Science 13348, Springer, 2022, 226–244.