On the Size of Overlapping Lempel-Ziv and Lyndon Factorizations

Yuki Urabe
Department of Informatics, Kyushu University, Japan
yuki.urabe@inf.kyushu-u.ac.jp

Yuto Nakashima
Department of Informatics, Kyushu University, Japan
yuto.nakashima@inf.kyushu-u.ac.jp

Shunsuke Inenaga
Department of Informatics, Kyushu University, Japan
inenaga@inf.kyushu-u.ac.jp

Hideo Bannai
Department of Informatics, Kyushu University, Japan
bannai@inf.kyushu-u.ac.jp

Masayuki Takeda
Department of Informatics, Kyushu University, Japan
takeda@inf.kyushu-u.ac.jp

Abstract

Lempel-Ziv (LZ) factorization and Lyndon factorization are well-known factorizations of strings. Recently, Kärkkäinen et al. studied the relation between the sizes of the two factorizations, and showed that the size of the Lyndon factorization is always smaller than twice the size of the non-overlapping LZ factorization [STACS 2017]. In this paper, we consider a similar problem for the overlapping version of the LZ factorization. Since the size of the overlapping LZ factorization is always smaller than the size of the non-overlapping LZ factorization and, in fact, can even be an $O(\log n)$ factor smaller, it is not immediately clear whether a similar bound as in previous work would hold. Nevertheless, in this paper, we prove that the size of the Lyndon factorization is always smaller than four times the size of the overlapping LZ factorization.

2012 ACM Subject Classification Mathematics of computing → Combinatorics on words

Keywords and phrases Lyndon factorization, Lyndon words, Lempel-Ziv factorization

Digital Object Identifier 10.4230/LIPIcs.CPM.2019.29

Funding *Yuto Nakashima*: Supported by JSPS KAKENHI Grant Number JP18K18002.
Shunsuke Inenaga: Supported by JSPS KAKENHI Grant Number JP17H01697.
Hideo Bannai: Supported by JSPS KAKENHI Grant Number JP16H02783.
Masayuki Takeda: Supported by JSPS KAKENHI Grant Number JP18H04098.

1 Introduction

A *factorization* of a string w is a sequence of non-empty substrings of w such that the concatenation of the substrings in the sequence is w. Various types of factorizations of strings have been proposed so far, and most, if not all, of them are categorized into two (not necessarily disjoint) categories. One is to factorize a given string w into *combinatorial objects* such as squares (square factorization [9, 18]), repetitions (repetition factorization [14]), palindromes (palindromic factorization [13, 10, 4, 2]), closed words (closed factorization [1]), and Lyndon words (Lyndon factorization [6]), while the other is to factorize a given string w as *efficient preprocessing* for text processing, in particular, text compression [21, 22, 20].

30th Annual Symposium on Combinatorial Pattern Matching (CPM 2019).
Editors: Nadia Pisanti and Solon P. Pissis; Article No. 29; pp. 29:1–29:11
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Amongst the variety of string factorizations, the Lyndon factorization [6] and the Lempel-Ziv (LZ for short) factorization [21] are probably those that are most well-known and extensively studied from the above categories, respectively, and this paper also deals with these factorizations.

As will be seen below, the definitions of LZ and Lyndon factorizations are rather different, and hence the results of these factorizations of the same string can also be very different. On the other hand, quite interestingly, both LZ and Lyndon factorizations have been used as efficient preprocessing for linear-time computation of *runs* or *maximal repetitions* in a given string [16, 5, 7, 3, 17, 11, 8]. Another connection between LZ and Lyndon factorizations is that both of the sizes of the LZ and Lyndon factorizations of a string w are lower bounds of the output size of any grammar compression for w [19, 12]. Here, by the size of a factorization we mean the number of factors in the factorization. Now, a natural question would be: How much the sizes of the LZ and Lyndon factorizations of the same string can differ?

This question was first considered by Kärkkäinen et al. [15] for the non-overlapping variant of LZ factorization. The non-overlapping LZ factorization of a string w is a sequence $p_1, \ldots, p_{z_{no}}$ of z_{no} factors such that each p_i is a single character if it is the first occurrence of the character in w, or p_i is the longest prefix of $p_i \cdots p_{z_{no}}$ that has an occurrence in $p_1 \cdots p_{i-1}$. A string ℓ is said to be a *Lyndon word*, if ℓ is lexicographically smaller than all of its non-empty proper suffixes. A factorization $f_1^{e_1}, \ldots, f_m^{e_m}$ is said to be the Lyndon factorization of a string w if f_i is a Lyndon word, $e_i \geq 1$, and f_i is lexicographically larger than f_{i+1} for all i. For many strings, the size m of Lyndon factorization is smaller than the size z_{no} of non-overlapping LZ factorization. However, they showed that there is a series of strings for which $m = z_{no} + \Theta(\sqrt{z_{no}})$ holds. In addition, they proved that the inequality $m < 2z_{no}$ holds for any string.

In this paper, we consider the relationship between the size of *overlapping variant of LZ factorization* and Lyndon factorization of the same string. The non-overlapping LZ factorization of a string w is a sequence $q_1, \ldots, q_z$ of z factors such that each q_i is a single character if it is the first occurrence of the character in w, or q_i is the longest prefix of $q_i \cdots q_{z_{no}}$ that has another occurrence in w beginning at a position within $q_1 \cdots q_{i-1}$. It is known that $z \leq z_{no}$ always holds, and there are cases where z is by a factor of $O(\log n)$ smaller than z_{no}: E.g., for a trivial string a^n, $z = 2$ while $z_{no} = \Theta(\log n)$. These facts make it more challenging to show an upper bound for m in terms of z. Still, in this paper, we prove that the inequality $m < 4z$ holds for any string. Our proof generally follows the scheme introduced by Kärkkäinen et al. [15], but our analysis leading to the inequality $m < 4z$ is original and seems to be interesting.

2 Preliminaries

2.1 Strings

Let Σ be an ordered *alphabet*. An element of Σ^* is called a *string*. The length of a string w is denoted by $|w|$. The empty string ε is a string of length 0. Let Σ^+ be the set of non-empty strings, i.e., $\Sigma^+ = \Sigma^* - \{\varepsilon\}$. For a string $w = xyz$, x, y and z are called a *prefix*, *substring*, and *suffix* of w, respectively. The i-th character of a string w is denoted by $w[i]$, where $1 \leq i \leq |w|$. For a string w and two integers $1 \leq i \leq j \leq |w|$, let $w[i..j]$ denote the substring of w that begins at position i and ends at position j. For convenience, let $w[i..j] = \varepsilon$ when $i > j$. For any string w let $w^1 = w$, and for any integer $k \geq 2$ let $w^k = ww^{k-1}$, i.e., w^k is a k-times repetition of w.

If character a is lexicographically smaller than another character b, then we write $a \prec b$. For any strings x, y, let $lcp(x, y)$ be the length of the longest common prefix of x and y. We write $x \prec y$ iff either $x[lcp(x, y) + 1] \prec y[lcp(x, y) + 1]$ or x is a proper prefix of y.

2.2 Lyndon words and Lyndon factorization of strings

A string w is said to be a *Lyndon word*, if w is lexicographically strictly smaller than all of its non-empty proper suffixes. The *Lyndon factorization* of a string w is the factorization $f_1^{e_1}, \ldots, f_m^{e_m}$ of w, such that each $f_i \in \Sigma^+$ is a Lyndon word, $e_i \geq 1$, and $f_i \succ f_{i+1}$ for all $1 \leq i < m$. We call m the size of the Lyndon factorization of w. We also refer to each f_i as a Lyndon factor and each $F_i = f_i^{e_i}$ as a Lyndon run of w.

2.3 Lempel-Ziv factorization of strings

The *overlapping Lempel-Ziv factorization* (LZ factorization for short) of a string w is the factorization $p_1, \ldots, p_z$ of w such that either p_i is a character which does not appear in $p_1 \cdots p_{i-1}$ or p_i is the longest prefix of $p_i \cdots p_z$ which has another occurrence to the left. We refer to each p_i as an *LZ phrase*. For any substring $w[i..j]$ $(1 \leq i \leq j \leq |w|)$ in w, $w[i..j]$ is said to contain an *LZ phrase boundary* if there exists an LZ phrase which begins in $[i, j]$.

3 Tools for non-overlapping LZ factorization

In this paper, we give the following result.

▶ **Theorem 1.** *Let m be the size of the Lyndon factorization of a string w and z the size of the (overlapping) LZ factorization of w. For any string w, $m < 4z$ holds.*

We prove Theorem 1 in Section 4. Our proof follows similar techniques for non-overlapping version which was introduced by Kärkkäinen et al. [15]. In this section, we explain their techniques which can be also applied for overlapping version.

3.1 Leftmost occurrence and factorizations

Each factorization catches the leftmost occurrences of particular substrings. Lemma 2 can be easily obtained by the definition of LZ factorization.

▶ **Lemma 2.** *If a substring $w[i..j]$ does not have any occurrence to the left, $w[i..j]$ contains an LZ phrase boundary.*

▶ **Lemma 3** (Lemma 4 of [15])**.** *Let $d \geq 1$ and $1 \leq i \leq m - d + 1$, and assume that $F_i \cdots F_{i+d-1}$ has an occurrence to the left of the trivial one in w. Then:*

1. *The leftmost occurrence of $F_i \cdots F_{i+d-1}$ is a prefix of f_j for some $j < i$;*
2. *$F_i \cdots F_{i+d-1}$ is a prefix of every f_k with $j \leq k < i$.*

3.2 Domains

Due to Lemma 3, each concatenation of several Lyndon runs has a range such that every Lyndon run in the range has the concatenation as a prefix.

▶ **Definition 4** (Definition 5 of [15])**.** *Let $d \geq 1$ and $1 \leq i \leq m - d + 1$. d-domain of a Lyndon run F_i, denoted by $\mathsf{dom}_d(F_i)$, is the substring $F_j \cdots F_{i-1}$ where F_j is the Lyndon run starting at the same position as the leftmost occurrence of $F_i \cdots F_{i+d-1}$ in w. Note that*

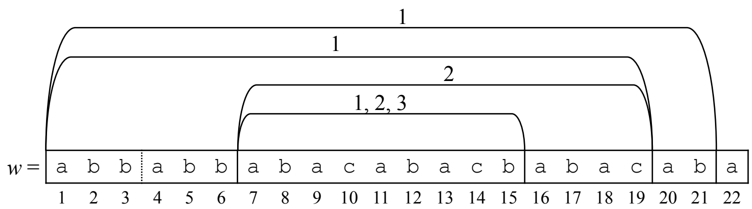

Figure 1 All non-empty domains in string $w =$ abbabbabacabacbabacaba are illustrated. Since the leftmost occurrence of $w[20..22]$ in w is $w[7..9]$, 2-domain of Lyndon run $w[20..21]$ is $w[7..19]$. Moreover, $w[7..9]$ is associated with this 2-domain. (This figure imitates Figure 1 of [15].)

if $F_i \cdots F_{i+d-1}$ does not have any occurrence to the left of the trivial one then $\mathsf{dom}_d(F_i) = \varepsilon$. The integers d and $i - j$ are called the order *and* size *of the domain, respectively. The* extended d-domain *of F_i is the substring $\mathsf{extdom}_d(F_i) = \mathsf{dom}_d(F_i) \cdot F_i \cdots F_{i+d-1}$ of w.*

By the definition and Lemma 2, each domain contains an LZ phrase boundary. For any domain $\mathsf{dom}_d(F_i)$, we say that the leftmost occurrence of $F_i \cdots F_{i+d-1}$ is *associated* with $\mathsf{dom}_d(F_i)$ (Definition 7 of [15]).

▶ **Lemma 5** (Lemma 8 of [15]). *Each substring associated with a domain contains an LZ phrase boundary.*

We show an example of domains in Figure 1.

3.3 Tandem domains

▶ **Definition 6** (Definition 9 of [15]). *Let $d \geq 1$ and $1 \leq i \leq m - d$. A pair of domains $\mathsf{dom}_{d+1}(F_i), \mathsf{dom}_d(F_{i+1})$ is called a* tandem domain *if $\mathsf{dom}_{d+1}(F_i) \cdot F_i = \mathsf{dom}_d(F_{i+1})$ or, equivalently, if $\mathsf{extdom}_{d+1}(F_i) = \mathsf{extdom}_d(F_{i+1})$. Note that we permit $\mathsf{dom}_{d+1}(F_i) = \varepsilon$.*

Let $\mathsf{dom}_{d+1}(F_i), \mathsf{dom}_d(F_{i+1})$ be a tandem domain. By Lemma 3, F_i can be written as $F_i = F_{i+1} \cdots F_{i+d} \cdot x$ for some $x \in \Sigma^+$. Thus, $F_i \cdots F_{i+d} = F_{i+1} \cdots F_{i+d} \cdot x \cdot F_{i+1} \cdots F_{i+d}$. We say that the occurrence of $x \cdot F_{i+1} \cdots F_{i+d}$ in the leftmost occurrence of $F_i \cdots F_{i+d}$ is *associated* with the tandem domain $\mathsf{dom}_{d+1}(F_i), \mathsf{dom}_d(F_{i+1})$ (Definition 10 of [15]).

In Figure 1, a pair of 3-domain of Lyndon run $w[16..19]$ and 2-domain of Lyndon run $w[20..21]$ is a tandem domain. Moreover, $w[10..13]$ is associated with the tandem domain.

3.4 Groups

▶ **Definition 7.** *Let $d \geq 1, 2 \leq p \leq m$, and $1 \leq i \leq m - d - p + 2$. A set of p domains $\mathsf{dom}_{d+p-1}(F_i), \ldots, \mathsf{dom}_d(F_{i+p-1})$ is called a* p-group *if for all $t = 0, \ldots, p-2$ the equality $\mathsf{dom}_{d+p-1-t}(F_{i+t}), \mathsf{dom}_{d+p-2-t}(F_{i+t+1})$ hols or, equivalently, $\mathsf{extdom}_{d+p-1}(F_i) = \ldots = \mathsf{extdom}_d(F_{i+p-1})$. Note that we permit $\mathsf{dom}_{d+p-1}(F_i) = \varepsilon$.*

Let $\mathsf{dom}_{d+p-1}(F_i), \ldots, \mathsf{dom}_d(F_{i+p-1})$ is a p-group. F_i has $F_{i+p-1} \cdots F_{i+p+d-2}$ as a prefix by Lemma 3. Then, $F_i \cdots F_{i+p+d-2} = F_{i+p-1} \cdots F_{i+p+d-2} \cdot x \cdot F_{i+1} \cdots F_{i+p+d-2}$ for some $x \in \Sigma^*$. We say that the occurrence of $x \cdot F_{i+1} \cdots F_{i+p+d-2}$ in the leftmost occurrence of $F_i \cdots F_{i+p+d-2}$ is *associated* with the group.

▶ **Lemma 8** (Lemma 16 of [15]). *The substring associated with a p-group is the concatenation, in reverse order, of the $p - 1$ substrings associated with the tandem domains belonging to the p-group.*

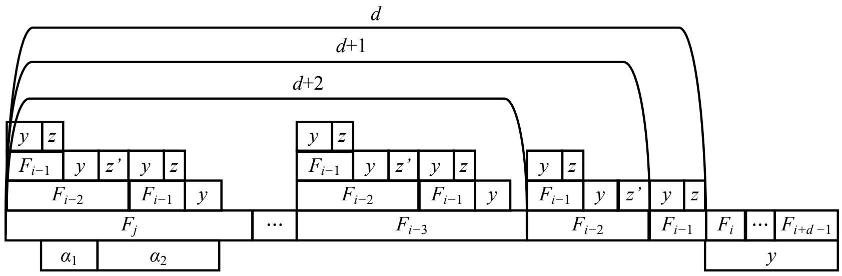

Figure 2 This figure illustrated 3-group $\mathsf{dom}_{d+2}(F_{i-2}), \mathsf{dom}_{d+1}(F_{i-1}), \mathsf{dom}_{d}(F_i)$. $\alpha_1(= zy)$ is the substring associated with tandem domain $\mathsf{dom}_{d+1}(F_{i-1}), \mathsf{dom}_{d}(F_i)$, and $\alpha_2(= z'yzy)$ is the substring associated with tandem domain $\mathsf{dom}_{d+2}(F_{i-2}), \mathsf{dom}_{d+1}(F_{i-1})$. Moreover, $\alpha_1\alpha_2$ is the substring associated with the 3-group. (This figure imitates Figure 2 of [15].)

Two groups $\mathsf{dom}_{d+p-1}(F_i), \ldots, \mathsf{dom}_d(F_{i+p-1})$ and $\mathsf{dom}_{d'+p'-1}(F_k), \ldots, \mathsf{dom}_{d'}(F_{k+p'-1})$ are said to be *disjoint* if $i + p - 1 < k$ or $k + p' - 1 < i$. For any disjoint groups, the following property holds.

▶ **Lemma 9** (Lemma 18 of [15]). *Substring associated with disjoint groups do not overlap.*

3.5 Subdomains

▶ **Definition 10** (Definition 19 of [15]). $\mathsf{dom}_e(F_k)$ *is said to be a* subdomain *of* $\mathsf{dom}_d(F_i) = F_j \cdots F_{i-1}$ *if either*

- $k = i$ *and* $e = d$, *or*
- $j \le k < i$ *and* $\mathsf{extdom}_e(F_k)$ *is a substring of* $\mathsf{extdom}_d(F_i)$.

▶ **Lemma 11** (Lemma 20 of [15]). *Let* $\mathsf{dom}_e(F_{k+1}), \mathsf{dom}_{e+1}(F_k)$ *be a tandem domain. If* $\mathsf{dom}_e(F_{k+1})$ *and* $\mathsf{dom}_{e+1}(F_k)$ *are both subdomains of a domain* $\mathsf{dom}_d(F_i)$, *then the substring associated with* $\mathsf{dom}_d(F_i)$ *does not overlap the substring associated with tandem domain* $\mathsf{dom}_e(F_{k+1}), \mathsf{dom}_{e+1}(F_k)$.

From this lemma, if every domain in a group is a subdomain of domain $\mathsf{dom}_d(F_i)$, the substring associated with $\mathsf{dom}_d(F_i)$ does not overlap the substring associated with the group.

3.6 Canonical subdomains

For any domain $\mathsf{dom}_d(F_i) = F_j \cdots F_{i-1}$, we define *canonical subdomain* $C_{i,d}$ as follows. $C_{i,d}$ is the set of subdomains of $\mathsf{dom}_d(F_i)$ which can be obtained by the following conditions. Initially, we set $\delta = d + 1, l = i - 1$. When $l = j$, then we finish the operations.

- If $\mathsf{dom}_\delta(F_l) = F_j \cdots F_{l-1}$, we add $\mathsf{dom}_\delta(F_l)$ into the set $C_{i,d}$, and set $\delta = \delta + 1, l = l - 1$.
- If $\mathsf{dom}_\delta(F_l) = F_{j'} \cdots F_{l-1}$ $(j < j')$, we add $\mathsf{dom}_\delta(F_l)$ into the set $C_{i,d}$, and set $\delta = 1, l = j' - 1$. All domains that were added to the set in this case are called *loose subdomains.*

We refer to each set of consecutive non-loose subdomains as a *cluster*. Note that the number of clusters is the number of loose subdomains plus one. Since $\mathsf{dom}_{d'}(F_j) = \varepsilon$, the domain w.r.t. F_j is always a cluster.

Let t be the number of loose subdomains in canonical sundomains $C_{i,d}$ of domain $\mathsf{dom}_d(F_i)$. We can discuss the number of LZ phrase boundaries contained in $\mathsf{extdom}_d(F_i)$. Let $\mathsf{dom}_{d_1}(F_{i_1}), \ldots, \mathsf{dom}_{d_t}(F_{i_t})$ $(i_1 < \ldots < i_t)$ be the sequence of loose subdomains, and

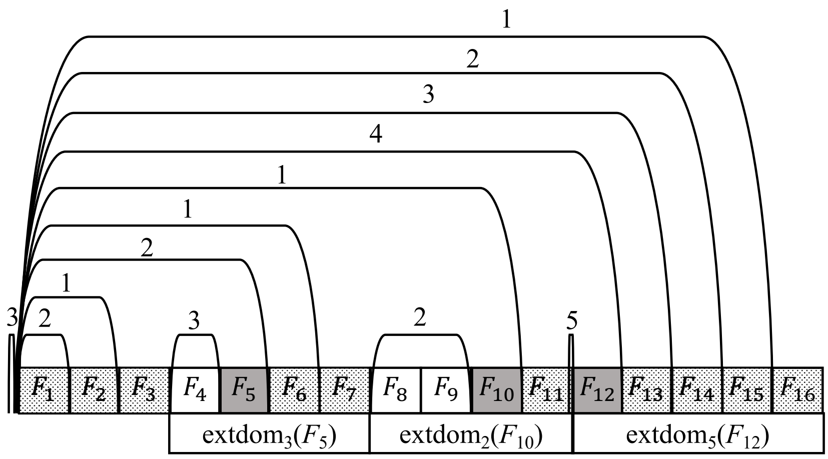

Figure 3 This figure illustrates the canonical subdomains of $\mathsf{dom}_1(F_{16}) = F_1 \cdots F_{15}$. (This figure imitates Figure 3 of [15].)

l (≥ 1) the number of Lyndon runs in the leftmost cluster. By the definition of loose subdomains, we have the following equality.

$$\mathsf{extdom}_d(F_i) = F_j \cdots F_{j+l-1} \cdot \mathsf{extdom}_{d_1}(F_{i_1}) \cdots \mathsf{extdom}_{d_t}(F_{i_t}) \tag{1}$$

Let S be the sum of the number of the LZ phrase boundaries contained in substrings associated with each clusters of $C_{i,d}$. By Lemma 9, these substrings do not overlap each other, and they are in $F_j \cdots F_{j+l-1}$. Moreover, they do not overlap the substring associated with $\mathsf{dom}_d(F_i)$ since they are also subdomains of $\mathsf{dom}_d(F_i)$ (by Lemma 11). Thus, by Lemma 5, there exists an LZ phrase boundary in $F_j \cdots F_{j+l-1}$ which was not counted in S. Let n_h be the number of LZ phrase boundaries which is contained in $\mathsf{extdom}_{d_h}(F_{i_h})$. It is clear that these boundaries are not in $F_j \cdots F_{j+l-1}$. Thus, they do not overlap the substring associated with the group and $\mathsf{dom}_d(F_i)$, respectively. Finally, we can discuss the number $N_{i,d}$ of LZ phrase boundaries in $\mathsf{extdom}_d(F_i)$ by using Equality (2):

$$N_{i,d} \geq 1 + \sum_{h=1}^{t} n_h + S. \tag{2}$$

4 Proof for overlapping LZ factorization

In this section, we prove Theorem 1. Our proof follows a general scheme introduced by Kärkkäinen et al. [15]. However, our analysis leading to the inequality $m < 4z$ is original and seems to be interesting.

4.1 Number of LZ phrase boundaries in groups

In the proof for non-overlapping version, Corollary 17 of [15] is one of the important properties. However, the corollary does not hold for overlapping version of LZ factorization. We want to introduce a new lemma as Lemma 13 for our problem. We start from the following lemma.

▶ **Lemma 12.** *Each substring associated with a 3-group contains an LZ phrase boundary.*

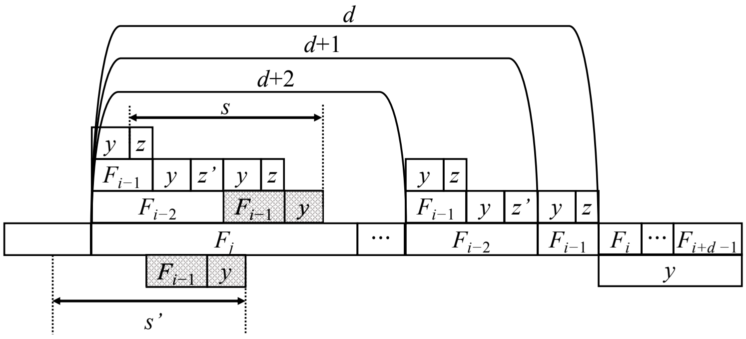

Figure 4 An illustration of the first case of proof for Lemma 12.

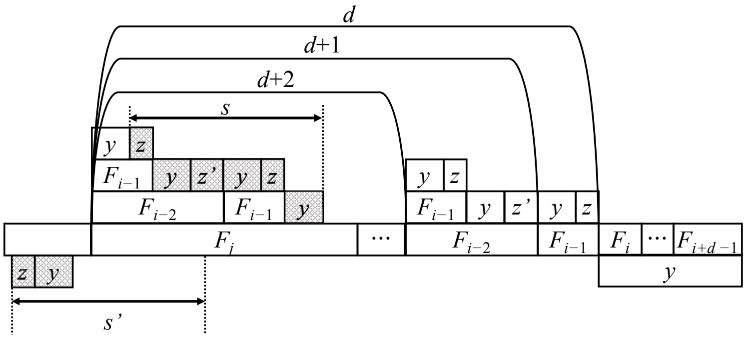

Figure 5 An illustration of the second case of proof for Lemma 12.

Proof. Let $\mathsf{dom}_{d+2}(F_{i-2}), \mathsf{dom}_{d+1}(F_{i-1}), \mathsf{dom}_d(F_i)$ be a 3-group. By the definition of groups, F_{i-1} can be written as $F_i \cdots F_{i+d-1} \cdot z$ for some $z \in \Sigma^+$, and F_{i-2} can be written as $F_{i-2} = F_{i-1} \cdots F_{i+d-1} \cdot z' = F_i \cdots F_{i+d-1} \cdot z \cdot F_i \cdots F_{i+d-1} \cdot z'$ for some $z' \in \Sigma^+$. For convenience, $y = F_i \cdots F_{i+d-1}$. Then, $F_{i-2} \cdots F_{i+d-1} = y \cdot z \cdot y \cdot z' \cdot F_{i-1} \cdot y$.

The substring associated with the 3-group is the suffix $z \cdot y \cdot z' \cdot F_{i-1} \cdot y$ of the leftmost occurrence of $F_{i-2} \cdots F_{i+d-1}$. s denotes the occurrence (see Figure 4). Suppose that $z \cdot y \cdot z' \cdot F_{i-1} \cdot y$ does not have any LZ phrase boundaries at the occurrence. By the definition of LZ factorization, $z \cdot y \cdot z' \cdot F_{i-1} \cdot y$ has an occurrence to the left. Let s' be one of such occurrences of $z \cdot y \cdot z' \cdot F_{i-1} \cdot y$. We consider the suffix $F_{i-1} \cdot y$ of s'. If a prefix of this suffix $F_{i-1} \cdot y$ overlaps a suffix of F_{i-2}(see Figure 4). This fact implies that f_{i-2} has a prefix of $F_{i-1} \cdot y$ as a suffix since $F_{i-2} = f_{i-2}^{e_{i-2}}$. On the other hand, f_{i-2} has $F_{i-1} \cdot y$ as a prefix by Lemma 3. Hence, f_{i-2} has a prefix of $F_{i-1} \cdot y$ as a prefix and also a suffix. This fact contradicts that f_{i-2} is a Lyndon word. Thus, the distance between s and s' has to be at least $|F_{i-1} \cdot y| + 1$. However, this fact also contradicts the leftmost occurrence of y (the leftmost occurrence of y is a prefix of F_j in fact, see also Figure 5). Therefore, every substring associated with a 3-group contains an LZ phrase boundary. ◀

By using this lemma, we can easily obtain the following key lemma.

▶ **Lemma 13.** *Each substring associated with a p-group contains at least $\lfloor \frac{p-1}{2} \rfloor$ LZ phrase boundaries.*

Proof. From Lemma 8, the substring associated with a p-group is the concatenation of $p-1$ substrings associated with tandem domains. The substring associated with 3-group contains an LZ phrase boundary by Lemma 12. Let x and y be the consecutive substrings which are associated with two consecutive tandem domains. Then, either x or y contains an LZ phrase boundary. Therefore, there exists at least $\lfloor \frac{p-1}{2} \rfloor$ LZ phrase boundaries. ◀

4.2 Number of LZ phrase boundaries in extended domains

▶ **Lemma 14.** *Let* $\mathsf{dom}_d(F_i)$ *be a domain of size* $k \geq 0$. $\mathsf{extdom}_d(F_i)$ *contains at least* $\lceil \frac{k-1}{4} \rceil + 1$ *LZ phrase boundaries (namely* $N_{i,d} \geq \lceil \frac{k-1}{4} \rceil + 1$*).*

Proof. Let $\mathsf{dom}_d(F_i) = F_j \cdots F_{i-1}$ be a domain of size $k = i - j$. We prove this lemma by induction on k. If $k = 0$, then the substring associated with $\mathsf{dom}_d(F_i)$ contains an LZ phrase boundary and the statement holds. Now we assume that $k \geq 1$ and the lemma holds for all $k \leq k'$ for some k'.

Firstly, we consider the case when $C_{i,d}$ does not have loose subdomain. In that case, $\mathsf{dom}_{d+k}(F_j), \ldots, \mathsf{dom}_d(F_i)$ is a $(k+1)$-group. By Lemma 5, the substring associated with $\mathsf{dom}_d(F_i)$ contains an LZ phrase boundary. On the other hand, by Lemma 13, the substring associated with the $(k+1)$-group contains $\lfloor \frac{k}{2} \rfloor$ LZ phrase boundaries. Since every domain in the group is a subdomain of $\mathsf{dom}_d(F_i)$, the substring associated with $\mathsf{dom}_d(F_i)$ does not overlap each of them by Lemma 11. Thus, $\mathsf{extdom}_d(F_i)$ contains $\lfloor \frac{k}{2} \rfloor + 1$ LZ phrase boundaries. The statement of the lemma holds for this case since $\lfloor \frac{k}{2} \rfloor + 1 \geq \lceil \frac{k-1}{4} \rceil + 1$.

Suppose that $C_{i,d}$ has t (≥ 1) loose subdomains. Let $\mathsf{dom}_{d_1}(F_{i_1}), \ldots, \mathsf{dom}_{d_t}(F_{i_t})$ be the t loose subdomains of $C_{i,d}$ and k_h the size of loose subdomain $\mathsf{dom}_{d_h}(F_{i_h})$ for any $1 \leq h \leq t$. We can see a lower bound of $N_{i,d}$ by using Equation (2). For the second term of Equation (2), $n_h \geq \lceil \frac{k_h-1}{4} \rceil + 1$ holds by an induction hypothesis. Now we analyze the sum of k_h for all h. Let l be the number of domains in the leftmost cluster. Then,

$$\sum_{h=1}^{t} k_h = k - l - \sum_{h=1}^{t-1} d_h - (d_t - d) \tag{3}$$

holds. Next, we analyze the third term of Equation (2). Notice that S is the sum of the number of LZ phrase boundaries which are contained in substrings associated with each group that is a cluster in $C_{i,d}$. The leftmost cluster is a l-group, the rightmost cluster is a $(d_t - d)$-group, and each of other clusters is $(d_h - 1)$-group. For convenience, we consider 1-group as a single domain and 0-group as an empty set of domains. It is clear that substrings associated with each of them has no LZ phrase boundary. Thus, S can be written as

$$S = \left\lfloor \frac{l-1}{2} \right\rfloor + \sum_{h=1}^{t-1} \left\lfloor \frac{1}{2}(d_h - 1 - [d_h > 1]) \right\rfloor + \left\lfloor \frac{d_t - d - 1}{2} \right\rfloor \tag{4}$$

by using Knuth's notation [predicate] for the numerical value (0 or 1) of the predicate in brackets. We partition $(t-1)$ clusters (which are not the leftmost and the rightmost) into two sets as;

$$\begin{aligned} T_1 &= \{h \mid d_h \geq 3, h \in [1, t-1]\}, and \\ T_2 &= \{h \mid d_h < 3, h \in [1, t-1]\}. \end{aligned}$$

For any non-negative integer e, $\lfloor \frac{e}{2} \rfloor \geq \frac{e}{2} - \frac{1}{2}$ holds. By using this inequation, the second term in the right-hand side of Equation (4) can be written as

$$\begin{aligned}
& \sum_{h=1}^{t-1} \left\lfloor \frac{1}{2}(d_h - 1 - [d_h > 1]) \right\rfloor \\
= & \sum_{h \in T_1} \left\lfloor \frac{1}{2}(d_h - 1 - [d_h > 1]) \right\rfloor \geq \frac{1}{2} \sum_{h \in T_1} (d_h - 1 - [d_h > 1]) - \frac{|T_1|}{2} \\
= & \frac{1}{2} \sum_{h \in T_1} \left(\frac{d_h}{3} - [d_h > 1] \right) + \frac{1}{3} \sum_{h \in T_1} d_h - |T_1| \geq \frac{1}{3} \sum_{h \in T_1} d_h - |T_1|.
\end{aligned}$$

Thus, S can be also written as

$$S \geq \frac{1}{3} \sum_{h \in T_1} d_h - |T_1| + \alpha \left(\alpha = \left\lfloor \frac{l-1}{2} \right\rfloor + \left\lfloor \frac{d_t - d - 1}{2} \right\rfloor \right).$$

Moreover, Equation (2) can be written as

$$\begin{aligned}
& 1 + \sum_{h=1}^{t} \left(\left\lceil \frac{k_h - 1}{4} \right\rceil + 1 \right) + S \\
\geq & 1 + \frac{3}{4}t + \frac{1}{4} \left(k - l - \sum_{h=1}^{t-1} d_h + d - d_t \right) + S \\
\geq & 1 + \frac{3}{4}t + \frac{1}{4}(k - l + d - d_t) - \frac{1}{4} \sum_{h \in T_1} d_h - \frac{1}{4} \sum_{h \in T_2} d_h + \frac{1}{3} \sum_{h \in T_1} d_h - |T_1| + \alpha \\
\geq & 1 + \frac{3}{4}t + \frac{1}{4}(k - l + d - d_t) - \frac{|T_2|}{2} + \frac{1}{12} \sum_{h \in T_1} d_h - |T_1| + \alpha \\
\geq & 1 + \frac{3}{4}(1 + |T_1| + |T_2|) + \frac{1}{4}(k - l + d - d_t) - \frac{|T_2|}{2} + \frac{|T_1|}{4} - |T_1| + \alpha \\
\geq & \frac{7}{4} + \frac{1}{4}(k - l + d - d_t) + \left\lfloor \frac{l-1}{2} \right\rfloor + \left\lfloor \frac{d_t - d - 1}{2} \right\rfloor.
\end{aligned}$$

Let $\beta = \frac{7}{4} + \frac{1}{4}(k - l + d - d_t) + \left\lfloor \frac{l-1}{2} \right\rfloor + \left\lfloor \frac{d_t - d - 1}{2} \right\rfloor$. We can prove $\beta \geq \frac{k-1}{4} + 1$ for each of three cases as follows. If $l = 1$, then

$$\begin{aligned}
\beta \geq & \frac{7}{4} + \frac{1}{4}(k - l + d - d_t) + \frac{d_t - d - 1}{2} - \frac{1}{2} \\
= & \frac{3}{4} + \frac{k-1}{4} + \frac{d_t - d}{4} \geq \frac{k-1}{4} + 1.
\end{aligned}$$

If $l > 1$ and $d_t - d - 1 = 1$, then

$$\begin{aligned}
\beta \geq & \frac{7}{4} + \frac{1}{4}(k - l + d - d_t) + \frac{l-1}{2} - \frac{1}{2} \\
= & 1 + \frac{k-1}{4} + \frac{l - (d_t - d)}{4} \geq \frac{k-1}{4} + 1.
\end{aligned}$$

If $l > 1$ and $d_t - d - 1 > 1$, then

$$\begin{aligned}
\beta \geq & \frac{7}{4} + \frac{1}{4}(k - l + d - d_t) + \frac{l-1}{2} - \frac{1}{2} \\
& + \frac{d_t - d - 1}{2} - \frac{1}{2} \\
= & \frac{k-1}{4} + \frac{l}{4} + \frac{d_t - d}{4} \geq \frac{k-1}{4} + 1.
\end{aligned}$$

Therefore, $N_{i,d} \geq \left\lceil \frac{k-1}{4} \right\rceil + 1$ holds. ◀

4.3 Proof of Theorem 1

Now, we are ready to prove Theorem 1.

Proof of Theorem 1. A string s can be written as the sequence of 1-domains, namely $s = \mathsf{extdom}_1(F_{i_1}) \cdots \mathsf{extdom}_1(F_{i_t})$ where $i_t = m$. Let k_h be the size of $\mathsf{dom}_1(F_{i_h})$. By Lemma 14, $\mathsf{extdom}_1(F_{i_h})$ contains $\lceil \frac{k_h - 1}{4} \rceil + 1$ LZ phrase boundaries. It is clear that $\sum_{h=1}^{t} k_h = m - t$. Therefore,

$$z \geq \sum_{h=1}^{t} \left(\left\lceil \frac{k_h - 1}{4} \right\rceil + 1 \right) \geq \frac{m - 2t}{4} + t > \frac{m}{4}$$

holds. ◀

5 Conclusion

We discussed the relationship between the size z of overlapping variant of LZ factorization and the size m of Lyndon factorization of the same string. We showed that the inequality $m < 4z$ holds for any string. One of the interesting open questions is whether there exists a better bound. Finally, we conjecture that the inequality $m < 2z$ holds for any string.

References

1 Golnaz Badkobeh, Hideo Bannai, Keisuke Goto, Tomohiro I, Costas S. Iliopoulos, Shunsuke Inenaga, Simon J. Puglisi, and Shiho Sugimoto. Closed factorization. *Discrete Applied Mathematics*, 212:23–29, 2016. `doi:10.1016/j.dam.2016.04.009`.

2 Hideo Bannai, Travis Gagie, Shunsuke Inenaga, Juha Kärkkäinen, Dominik Kempa, Marcin Piatkowski, and Shiho Sugimoto. Diverse Palindromic Factorization is NP-Complete. *Int. J. Found. Comput. Sci.*, 29(2):143–164, 2018. `doi:10.1142/S0129054118400014`.

3 Hideo Bannai, Tomohiro I, Shunsuke Inenaga, Yuto Nakashima, Masayuki Takeda, and Kazuya Tsuruta. The "Runs" Theorem. *SIAM Journal on Computing*, 46(5):1501–1514, 2017. `doi:10.1137/15M1011032`.

4 Kirill Borozdin, Dmitry Kosolobov, Mikhail Rubinchik, and Arseny M. Shur. Palindromic Length in Linear Time. In *CPM 2017*, pages 23:1–23:12, 2017. `doi:10.4230/LIPIcs.CPM.2017.23`.

5 Gang Chen, Simon J. Puglisi, and W. F. Smyth. Lempel-Ziv factorization using less time & space. *Mathematics in Computer Science*, 1(4):605–623, June 2008. `doi:10.1007/s11786-007-0024-4`.

6 K. T. Chen, R. H. Fox, and R. C. Lyndon. Free Differential Calculus, IV. The Quotient Groups of the Lower Central Series. *Annals of Mathematics*, 68(1):81–95, 1958. URL: `http://www.jstor.org/stable/1970044`.

7 Maxime Crochemore, Lucian Ilie, and Liviu Tinta. Towards a Solution to the "Runs" Conjecture. In Paolo Ferragina and Gad M. Landau, editors, *Combinatorial Pattern Matching*, pages 290–302, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

8 Maxime Crochemore, Costas S. Iliopoulos, Tomasz Kociumaka, Ritu Kundu, Solon P. Pissis, Jakub Radoszewski, Wojciech Rytter, and Tomasz Walen. Near-Optimal Computation of Runs over General Alphabet via Non-Crossing LCE Queries. In *SPIRE 2016*, pages 22–34, 2016. `doi:10.1007/978-3-319-46049-9_3`.

9 Marius Dumitran, Florin Manea, and Dirk Nowotka. On Prefix/Suffix-Square Free Words. In *SPIRE 2015*, pages 54–66, 2015. `doi:10.1007/978-3-319-23826-5_6`.

10 Gabriele Fici, Travis Gagie, Juha Kärkkäinen, and Dominik Kempa. A subquadratic algorithm for minimum palindromic factorization. *J. Discrete Algorithms*, 28:41–48, 2014. `doi:10.1016/j.jda.2014.08.001`.

11 Pawel Gawrychowski, Tomasz Kociumaka, Wojciech Rytter, and Tomasz Walen. Faster Longest Common Extension Queries in Strings over General Alphabets. In *CPM 2016*, pages 5:1–5:13, 2016. `doi:10.4230/LIPIcs.CPM.2016.5`.

12 Tomohiro I, Yuto Nakashima, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Faster Lyndon factorization algorithms for SLP and LZ78 compressed text. *Theor. Comput. Sci.*, 656:215–224, 2016. `doi:10.1016/j.tcs.2016.03.005`.

13 Tomohiro I, Shiho Sugimoto, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Computing Palindromic Factorizations and Palindromic Covers On-line. In *CPM 2014*, pages 150–161, 2014. `doi:10.1007/978-3-319-07566-2_16`.

14 Hiroe Inoue, Yoshiaki Matsuoka, Yuto Nakashima, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. Computing Smallest and Largest Repetition Factorizations in $O(n \log n)$ Time. In *PSC 2016*, pages 135–145, 2016. URL: `http://www.stringology.org/event/2016/p12.html`.

15 Juha Kärkkäinen, Dominik Kempa, Yuto Nakashima, Simon J. Puglisi, and Arseny M. Shur. On the Size of Lempel-Ziv and Lyndon Factorizations. In *STACS 2017*, pages 45:1–45:13, 2017. `doi:10.4230/LIPIcs.STACS.2017.45`.

16 Roman Kolpakov and Gregory Kucherov. Finding Maximal Repetitions in a Word in Linear Time. In *FOCS 1999*, pages 596–604, Washington, DC, USA, 1999. IEEE Computer Society. URL: `http://dl.acm.org/citation.cfm?id=795665.796470`.

17 Dmitry Kosolobov. Computing runs on a general alphabet. *Inf. Process. Lett.*, 116(3):241–244, 2016. `doi:10.1016/j.ipl.2015.11.016`.

18 Yoshiaki Matsuoka, Shunsuke Inenaga, Hideo Bannai, Masayuki Takeda, and Florin Manea. Factorizing a String into Squares in Linear Time. In *CPM 2016*, pages 27:1–27:12, 2016. `doi:10.4230/LIPIcs.CPM.2016.27`.

19 Wojciech Rytter. Application of Lempel-Ziv factorization to the approximation of grammar-based compression. *Theoretical Computer Science*, 302(1):211–222, 2003. `doi:10.1016/S0304-3975(02)00777-6`.

20 Terry A. Welch. A Technique for High-Performance Data Compression. *IEEE Computer*, 17(6):8–19, 1984. `doi:10.1109/MC.1984.1659158`.

21 J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977. `doi:10.1109/TIT.1977.1055714`.

22 Jacob Ziv and Abraham Lempel. Compression of individual sequences via variable-rate coding. *IEEE Trans. Information Theory*, 24(5):530–536, 1978. `doi:10.1109/TIT.1978.1055934`.