Vol.2019-AL-171 No.1
2019/1/29

Lyndon 分解と自己参照あり LZ 分解の項数の関係について

浦部 裕貴[1]　中島 祐人[1]　稲永 俊介[1]　坂内 英夫[1]　竹田 正幸[1]

概要：文字列の分解としてよく知られている Lyndon 分解と Lempel-Ziv 分解 (LZ 分解) は，分解の組合せ的性質が大きく異なっている．また，LZ 分解には様々な亜種が存在し，その 1 つである自己参照なし LZ 分解と Lyndon 分解の項数の関係について，Lyndon 分解の項数が LZ 分解の項数の 2 倍未満であることが近年示された．本研究では，自己参照あり LZ 分解と Lyndon 分解の関係について考え，Lyndon 分解の項数が自己参照あり LZ 分解の項数の 4 倍未満であることを示した．

キーワード：Lyndon 文字列，Lyndon 分解，Lempel-Ziv 分解

1. 序論

文字列をある規則に基づいて分解することは，その文字列が持つ性質や構造を捉えることにつながる．分解によって捉えられた性質をうまく利用した様々な効率的なアルゴリズムが知られている．このような事例の一つに，文字列中の極大な繰り返し（以下，連とよぶ）を計算する問題がある．連を計算する問題に対しては，Lempel-Ziv 分解 (以下，LZ 分解) を利用したアルゴリズム [2], [4], [8] や Lyndon 分解を利用したアルゴリズム [1], [5], [6], [9] が提案されている．LZ 分解とは，各項がそれ以前に出現している極大な文字列となる分解である [12]．LZ 分解には，以前の出現と項自身が重なることを許さない自己参照なし LZ 分解と，それらが重なることを許した自己参照あり LZ 分解がある．一方で Lyndon 分解とは，各項が Lyndon 文字列であり，項の列が辞書式順序について降順となる分解である [3]．ここで Lyndon 文字列とは，辞書式順序最小の接尾辞が自身である文字列のことである [10], [11]．

この二つの分解は定義も大きく異なるため，連計算問題を解くために利用されたことでその関係が注目された．これに対して近年，自己参照なし LZ 分解と Lyndon 分解について，それぞれの項数を z_{no}，m としたとき，$m < 2z_{no}$ であることが Kärkkäinen らによって示された [7]．これは，二つの分解の初めての直接的な関係を与えた結果である．

一方で自己参照あり LZ 分解は，自己参照なし LZ 分解に比べより一般的な定義であり，様々なアルゴリズムに利用されている．自己参照あり LZ 分解との関係を知ることが，より本質的な性質を得ることにつながるといえる．その定義からもわかるように，自己参照なし LZ 分解の項数は，自己参照あり LZ 分解の項数に対して一般に大きい．そのため，自己参照あり LZ 分解の項数と Lyndon 分解の項数の関係を得ることは，より難しい問題であると考えられた．

これに対し本稿では，自己参照あり LZ 分解の項数 z と Lyndon 分解の項数 m について，$m < 4z$ が成り立つことを示す．証明は，基本的には Kärkkäinen らの手法 [7] に基づいている．

2. 準備

アルファベット（文字の集合）を Σ とすると，Σ^* は文字列の集合である．任意の文字列について，$w = xyz$ $(x, y, z \in \Sigma^*)$ と書けるとき，x を w の接頭辞，y を w の部分文字列，z を w の接尾辞という．文字列 w の長さを $|w|$，文字列 w を k 個連結した文字列を w^k と表記する．2 つの文字列 x, y について，辞書式順序において x が y より小さいことを $x \prec y$ で表す．

Lyndon 分解

Lyndon 文字列及び **Lyndon 分解**の定義を示す．

定義 1 (Lyndon 文字列 [10], [11])．文字列 w が Lyndon 文字列であるとは，すべての真の接尾辞よりも w のほうが辞書式順序が小さいことである．

定義 2 (Lyndon 分解 [3])．文字列の列 $f_1^{q_1}, \ldots, f_m^{q_m}$ が文字列 w の Lyndon 分解であるとは以下を満たすことである．

- $f_1, \ldots, f_m$ が Lyndon 文字列であり，$f_1 \succ \cdots \succ f_m$ である．

[1] 九州大学
Kyushu University

- $w = f_1^{q_1} \cdots f_m^{q_m}$ $(q_1, \ldots, q_m \geq 1)$.

$f_i^{q_i}$ を F_i と記述し，Lyndon 項とよぶ（つまり，$F_1, \ldots, F_m$ は w の Lyndon 分解である）.

Lempel-Ziv 分解

自己参照あり LZ 分解を次に示す.

定義 3 (自己参照あり LZ 分解). 文字列の列 $p_1, \ldots, p_z$ が文字列 $w = p_1 \cdots p_z$ の自己参照あり LZ 分解であるとは，各項 p_i が以下のいずれか一方を満たすことである.

- $p_1 \cdots p_{i-1}$ 中に開始位置をもつ $p_i \cdots p_z$ の最長接頭辞.
- $p_1 \cdots p_{i-1}$ に出現しない文字.

以降，自己参照あり LZ 分解を単に **LZ 分解**とよぶこととする．LZ 分解の各項 p_i を LZ 項とよぶ．文字列 w の任意の区間 $[i, j]$ $(1 \leq i \leq j \leq |w|)$ について，$[i, j]$ 中にある LZ 項の開始位置が存在するとき，$[i, j]$ は w の **LZ 分解の境界を含む**とよぶ.

3. 自己参照なし LZ 分解に対する証明

我々が与える主結果の証明は 4 章で行う．本証明は，Kärkkäinen らの自己参照なし LZ 分解に対する証明 [7] のテクニックを応用している．この章では，Kärkkäinen らの証明で導入された domain やそれに伴う性質，証明の概要などについて述べる.

はじめに，2 つの分解それぞれと部分文字列の最左出現に関する性質について述べる．これらの性質に基づいて，2 つの分解の関係性を議論することができる．補題 1 は，LZ 分解の定義より簡単に得られる.

補題 1. 任意の部分文字列について，その最左出現となる区間は LZ 分解の境界を含む.

次の補題は Lyndon 分解についてであるが，証明は引用先を参照されたい.

補題 2 ([7]). 連続する d 個の Lyndon 項の連結 $F_i \cdots F_{i+d-1}$ が自身よりも左側に出現するとき以下を満たす.

- $F_i \cdots F_{i+d-1}$ はその最左出現において，ある Lyndon 項 F_j $(j < i)$ 及び f_j の接頭辞である.
- すべての Lyndon 文字列 f_k $(j \leq k < i)$ は接頭辞に $F_i \cdots F_{i+d-1}$ をもつ.

3.1 Domain

補題 2 より，連続する 1 つ以上の Lyndon 項の連結を接頭辞としてもつ Lyndon 項が連続して出現している範囲が存在する．この範囲をその連結による **domain** とよび，次のように定義する.

定義 4 ([7]). F_j を $F_i \cdots F_{i+d-1}$ の最左出現と開始位置が一致する Lyndon 項とする．このとき，$F_j \cdots F_{i-1}$ を Lyndon 項 F_i の d-domain とよび，$\mathsf{dom}_d(F_i) = F_j \cdots F_{i-1}$ で表す．また，$i - j$ を $\mathsf{dom}_d(F_i)$ のサイズとよび，サイズが 0 のとき（つまり，$F_i \cdots F_{i+d-1}$ は左側に出現をもたないとき），$\mathsf{dom}_d(F_i) = \varepsilon$ とする．さらに，$\mathsf{extdom}_d(F_i) = \mathsf{dom}_d(F_i) \cdot F_i \cdots F_{i+d-1}$ と定義する.

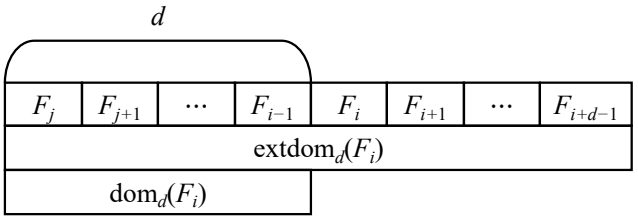

図 1 以降の図では，上記のように domain を弧線で表すこととする．この図では，弧線の終了位置の右側に続いている Lyndon 項 F_i の d-domain $\mathsf{dom}_d(F_i)$ を図示している．（この図は参考文献 [7] の Figure 1 を再現したものである.）

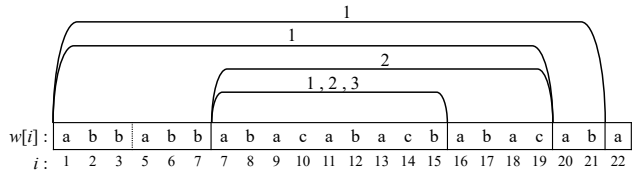

図 2 図中の文字列 w の空でない domain を全て図示している．2 つの Lyndon 項の連結 $w[20 \ldots 22]$ の w 中での最左出現は区間 $[7, 9]$ である．よって，Lyndon 項 $w[20 \ldots 21]$ の 2-domain は，$w[7 \ldots 19]$ である．また，この domain に紐付けられた区間は $[7, 9]$ である．（この図は参考文献 [7] の Figure 1 を再現したものである.）

上の定義と補題 1 より，任意の domain は最左出現である区間において LZ 分解の境界を含むことがわかる．任意の domain $\mathsf{dom}_d(F_i)$ について，$F_i \cdots F_{i+d-1}$ の最左出現である区間を，この **domain に紐付けられた区間**とよぶこととする.

補題 3 ([7]). domain に紐付けられた区間は，LZ 分解の境界を少なくとも 1 つ含む.

図 2 に，domain 及び domain に紐付けられた区間の例を示している.

3.2 Tandem domain

以下のように，特徴的な 2 つの domain の組について **tandem domain** を定義する.

定義 5 ([7]). 2 つの domain $\mathsf{dom}_{d+1}(F_i), \mathsf{dom}_d(F_{i+1})$ が，$\mathsf{dom}_{d+1}(F_i) \cdot F_i = \mathsf{dom}_d(F_{i+1})$ を満たすとき，この 2 つの domain の組を tandem domain とよぶ.

任意の tandem domain $\mathsf{dom}_{d+1}(F_i), \mathsf{dom}_d(F_{i+1})$ について，補題 2 より，ある文字列 x を用いて $F_i = F_{i+1} \cdots F_{i+d} \cdot x$ と書ける．同様に，$F_i \cdots F_{i+d} = F_{i+1} \cdots F_{i+d} \cdot x \cdot F_{i+1} \cdots F_{i+d}$ と書ける．このとき，$F_i \cdots F_{i+d}$ の最左出現の接尾辞である $x \cdot F_{i+1} \cdots F_{i+d}$ に対応する区間をこの **tandem domain に紐付けられた区間**とよぶ.

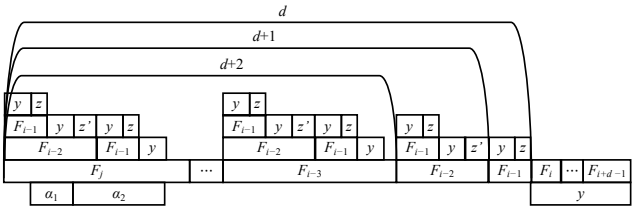

図 3 3-group $\mathsf{dom}_{d+2}(F_{i-2}), \mathsf{dom}_{d+1}(F_{i-1}), \mathsf{dom}_{d+2}(F_i)$ を図示している．α_1 が tandem domain $\mathsf{dom}_{d+1}(F_{i-1})$, $\mathsf{dom}_{d+2}(F_i)$ に紐付けられた区間の文字列であり，α_2 が tandem domain $\mathsf{dom}_{d+2}(F_{i-2}), \mathsf{dom}_{d+1}(F_{i-1})$ に紐付けられた区間の文字列である．また，この 3-group に紐付けられた区間の文字列は，連結 $\alpha_1\alpha_2$ と一致する．（この図は参考文献 [7] の Figure 2 を参考にしている．）

図 2 において，Lyndon 項 $w[16\ldots19]$ の 3-domain $w[7\ldots15]$ と Lyndon 項 $w[20\ldots21]$ の 2-domain $w[7\ldots19]$ の組は tandem domain であり，その紐付けられた区間は $[10, 13]$ である．

3.3 Group

tandem domain を以下のように一般化した概念が **group** である．

定義 6. p 個の domain の列 $\mathsf{dom}_{d+p-1}(F_i), \ldots, \mathsf{dom}_d(F_{i+p-1})$ が，任意の t $(0 \le t \le p-2)$ について，$\mathsf{dom}_{d+p-1-t}(F_{i+t}), \mathsf{dom}_{d+p-2-t}(F_{i+t+1})$ が tandem domain であるとき，この domain の列を p-group とよぶ．

p-group $\mathsf{dom}_{d+p-1}(F_i), \ldots, \mathsf{dom}_d(F_{i+p-1})$ について，補題 2 より，F_i は $F_{i+p-1}\cdots F_{i+p+d-2}$ を接頭辞としてもつ．よって，ある文字列 x を用いて，$F_i\cdots F_{i+p+d-2} = F_{i+p-1}\cdots F_{i+p+d-2}\cdot x\cdot F_{i+1}\cdots F_{i+p+d-2}$ と書ける．$F_i\cdots F_{i+p+d-2}$ の最左出現での接尾辞である $x\cdot F_{i+1}\cdots F_{i+p+d-2}$ をこの **group に紐付けられた区間**とする．

同じ group に属している隣り合う domain の組は tandem domain である．group に紐付けられた区間は，その group に含まれている tandem domain に紐付けられた区間を連結した区間となっている．

補題 4 ([7]). p-group に紐付けられた区間は $p-1$ 個の tandem domain に紐付けられた区間を連結したものである．

p-group $\mathsf{dom}_{d+p-1}(F_i), \ldots, \mathsf{dom}_d(F_{i+p-1})$ と p'-group $\mathsf{dom}_{d'+p'-1}(F_k), \ldots, \mathsf{dom}_{d'}(F_{k+p'-1})$ ついて，$i+p-1 < k$ または $k+p'-1 < i$ を満たすとき，この 2 つの group は異なる group であるという．任意の異なる group に紐付けられた区間について以下が成り立つ．

補題 5 ([7]). 異なる group に紐付けられた区間は重なり合わない．

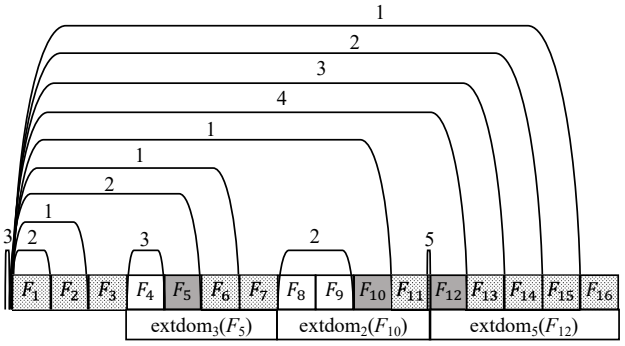

図 4 $\mathsf{dom}_1(F_{16}) = F_1\cdots F_{15}$ の canonical subdomain を例に示す．$F_{13}, \ldots, F_{15}$ に関する domain は group であり，cluster とよぶ．F_5, F_{10}, F_{12} に関する domain が loose subdomain であり，loose subdomain を用いて，$\mathsf{extdom}_1(F_{16}) = F_1\cdots F_3\cdot\mathsf{extdom}_3(F_5)\mathsf{extdom}_2(F_{10})\mathsf{extdom}_5(F_{12})$ と書き表せる．（この図は参考文献 [7] の Figure 3 を参考にしている．）

3.4 Subdomain

以下のように **subdomain** を定義する．subdomain を用いることで，domain と group に紐付けられた区間の関係を議論できる．

定義 7 ([7]). $\mathsf{dom}_e(F_k)$ が $\mathsf{dom}_d(F_i) = F_j\cdots F_{i-1}$ の subdomain であるとは，次のいずれかを満たすことである．

- $k = i$ かつ $e = d$.
- $j \le k < i$ かつ $\mathsf{extdom}_e(F_k)$ が $\mathsf{extdom}_d(F_i)$ の部分文字列．

補題 6 ([7]). domain $\mathsf{dom}_d(F_i)$ と tandem domain $\mathsf{dom}_e(F_{k+1}), \mathsf{dom}_{e+1}(F_k)$ について，$\mathsf{dom}_e(F_{k+1})$ と $\mathsf{dom}_{e+1}(F_k)$ がともに $\mathsf{dom}_d(F_i)$ の subdomain であるならば，$\mathsf{dom}_d(F_i)$ に紐付けられた区間と tandem domain に紐付けられた区間は重ならない．

補題 6 より，domain $\mathsf{dom}_d(F_i)$ と group についてある group に含まれる全ての domain が $\mathsf{dom}_d(F_i)$ の subdomain であるとき，$\mathsf{dom}_d(F_i)$ に紐付けられた区間とその group に紐付けられた区間は重ならないことがわかる．

3.5 Canonical subdomain

任意の domain $\mathsf{dom}_d(F_i) = F_j\cdots F_{i-1}$ について，**canonical subdomain** $C_{i,d}$ を以下のように定義する．$C_{i,d}$ は，$\mathsf{dom}_d(F_i)$ の subdomain の列であり，(i)，(ii) の条件により $C_{i,d}$ の先頭に要素を追加していくことで得られる．以下に現れる δ 及び l はそれぞれ初期値を $\delta = d+1, l = i-1$ し，$l = j$ となったとき終了する．

(i) $\mathsf{dom}_\delta(F_l) = F_j\cdots F_{l-1}$ のとき，$\mathsf{dom}_\delta(F_l)$ を $C_{i,d}$ の先頭に追加し，$\delta = \delta+1$，$l = l-1$ とする．

(ii) $\mathsf{dom}_\delta(F_l) = F_{j'}\cdots F_{l-1}$ $(j < j')$ のとき，$\mathsf{dom}_\delta(F_l)$ を $C_{i,d}$ の先頭に追加し，$\delta = 1, l = j'-1$ とする．このとき，$\mathsf{dom}_\delta(F_l)$ を **loose subdomain** とよぶ．

言い換えると上に示した操作は，$F_j, \ldots, F_i$ について，右

から左へ group となる極大な domain の列に分割する操作であるといえる．この極大な domain の列（loose subdomain でない連続した domain の列）のことを **cluster** とよぶ．このとき，[loose subdomain の数]+1 個の cluster が存在する．$\mathsf{dom}_{d'}(F_j) = \varepsilon$ より，F_j に関する domain は必ず cluster となることがわかる．

domain $\mathsf{dom}_d(F_i)$ の canonical subdomain $C_{i,d}$ 中の loose subdomain の数を t 個とし，$\mathsf{extdom}_d(F_i)$ に対応する区間に含まれる LZ 分解の境界の数について議論する．loose subdomain の列を $\mathsf{dom}_{d_1}(F_{i_1}), \ldots, \mathsf{dom}_{d_t}(F_{i_t})$ $(i_1 < \ldots < i_t)$ とし，最左である cluster のサイズを l (≥ 1) とする．loose subdomain の定義より，以下の式が得られる．

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

$C_{i,d}$ に含まれる domain において，cluster である group に紐付けられた区間に含まれる LZ 分解の境界の数の和を S とする．これらは補題 5 より互いに重ならず，group に紐付けられた区間の定義よりすべて $F_j \cdots F_{j+l-1}$ 中に出現する．また，それらは $\mathsf{dom}_d(F_i)$ の subdomain であるので，補題 6 より，$\mathsf{dom}_d(F_i)$ に紐付けられた区間とは重ならない．よって，補題 3 より，LZ 分解の境界は S に数えられたものとは異なる境界が $F_j \cdots F_{j+l-1}$ 中に少なくとも 1 つ存在する．$\mathsf{extdom}_{d_h}(F_{i_h})$ に含まれる LZ 分解の境界の数を n_h とする．これらの境界は明らかに $F_j \cdots F_{j+l-1}$ 以外の区間に存在するため，group に紐付けられた区間や $\mathsf{dom}_d(F_i)$ に紐付けられた区間と重ならない．以上より，$\mathsf{extdom}_d(F_i)$ に含まれる LZ 分解の境界の数を式 (2) のように区間ごとに分けて数え上げる．

$$1 + \sum_{h=1}^{t} n_h + S \tag{2}$$

4. 自己参照あり LZ 分解に対する証明

本章では，自己参照あり LZ 分解に対する以下の結果を証明する．

定理 1. 任意の文字列について，その Lyndon 分解の項数 m，（自己参照あり）LZ 分解の項数 z とすると，$m < 4z$ が成り立つ．

自己参照なし LZ 分解に対する証明と同様に，式 (2) に基づいて，$\mathsf{extdom}_d(F_i)$ に対応する区間に含まれる LZ 分解の境界の数について議論する．

4.1 group と LZ 分解の境界の数

group に紐付けられた区間と LZ 分解の境界の関係を補題 8 示す．この補題を利用することで，式 (2) の第 3 項 S で捉えている LZ 分解の境界を議論できる．この補題を示すために，はじめに以下の補題を示す．

補題 7. 3-group に紐付けられた区間は LZ 分解の境界を少なくとも 1 つ含む．

証明. 3-group $\mathsf{dom}_{d+2}(F_{i-2}), \mathsf{dom}_{d+1}(F_{i-1}), \mathsf{dom}_d(F_i)$ について考える．group の定義より，$F_{i-1} = F_i \cdots F_{i+d-1} \cdot z$ $(z \in \Sigma^+)$ であり，$F_{i-2} = F_{i-1} \cdots F_{i+d-1} \cdot z' = F_i \cdots F_{i+d-1} \cdot z \cdot F_i \cdots F_{i+d-1} \cdot z'$ $(z' \in \Sigma^+)$ と書ける．簡単のため，$y = F_i \cdots F_{i+d-1}$ とおくと，$F_{i-2} \cdots F_{i+d-1} = y \cdot z \cdot y \cdot z' \cdot F_{i-1} \cdot y$ である（図 3 参照）．

3-group に紐付けられた区間は $F_{i-2} \cdots F_{i+d-1}$ の最左出現の接尾辞 $z \cdot y \cdot z' \cdot F_{i-1} \cdot y$ に対応する区間である．この区間を I とし，I は LZ 分解の境界が含まないと仮定する．LZ 分解の定義より，I に対応する文字列 $z \cdot y \cdot z' \cdot F_{i-1} \cdot y$ は，I の開始位置より左側に出現をもつ．その出現における $F_{i-1} \cdot y$ の出現について考える．この出現のある接頭辞が，F_{i-2} の接尾辞と重なるとすると Lyndon 分解であることに矛盾する．よって，I に対応する文字列の左側の出現は少なくとも $|F_{i-1} \cdot y| + 1$ 以上左側に存在しなければならない．次に $F_{i-1} \cdot y$ の最左出現（F_j の接頭辞である出現）について考える．この位置での y の出現は区間 I に含まれているため，y は $|F_{i-1} \cdot y| + 1$ 以上左側に存在しなければならない．一方で y は F_j の接頭辞であり，かつこの出現が y の最左出現であるが，先の y の出現はこの最左出現より左側であるため矛盾する．したがって，3-group に紐付けられた区間は必ず LZ 分解の境界を含む． □

補題 8. p-group に紐付けられた区間は少なくとも $\left\lfloor \frac{p-1}{2} \right\rfloor$ 個の LZ 分解の境界を含む．

証明. 補題 4 より，p-group に紐付けられた区間は $p-1$ 個の tandem domain に紐付けられた区間の連結である．補題 7 より，3-group に紐付けられた区間が LZ 分解の境界を少なくとも 1 つ含むので，連続している tandem domain に紐付けられた区間について，2 つの隣り合う区間のうち少なくとも一方の区間は LZ 分解の境界を含む．よって，$p-1$ 個の tandem domain のうち LZ 分解の境界を含むものは少なくとも $\left\lfloor \frac{p-1}{2} \right\rfloor$ 個存在ため補題が成り立つ． □

4.2 定理 1 の証明

補題 8 と式 (2) を用いて，domain が捉えている LZ 分解の境界の数について以下の補題を示し，定理 1 を示す．

補題 9. サイズ k の domain $\mathsf{dom}_d(F_i)$ について，$\mathsf{extdom}_d(F_i)$ に対応する区間は少なくとも $\left\lceil \frac{k-1}{4} \right\rceil + 1$ 個の LZ 分解の境界を含む．

証明. サイズ k の domain $\mathsf{dom}_d(F_i) = F_j \cdots F_{i-1}$ について考える．k について，帰納的に示す．

(i) $k = 0$ のとき，$\mathsf{dom}_d(F_i)$ に紐付けられた区間が補題 3 より LZ 分解の境界を含む．domain に紐付けられた区間は $\mathsf{extdom}_d(F_i)$ に対応する区間に含まれているため，$\mathsf{extdom}_d(F_i)$ に対応する区間は LZ 分解の境界を少なくとも 1 つ含み，題意を満たす．

(ii) $k > 0$ かつ $C_{i,d}$ に loose subdomain が存在しないと

き，$\mathsf{dom}_{d+k}(F_j),\ldots,\mathsf{dom}_d(F_i)$ は $(k+1)$-group であり，$(k+1)$-group に紐付けられた区間は補題 8 より，$\lfloor\frac{k}{2}\rfloor$ 個の LZ 分解の境界を含む．また，それらはすべて $\mathsf{dom}_d(F_i)$ の subdomain であるので，補題 6 より，$\mathsf{dom}_d(F_i)$ に紐付けられた区間とは重ならない．さらに，補題 3 より，$\mathsf{dom}_d(F_i)$ に紐付けられた区間も LZ 分解の境界を含むので，合計で $\mathsf{extdom}_d(F_i)$ に対応する区間は $\lfloor\frac{k}{2}\rfloor+1$ 個の LZ 分解の境界を含む．よって $k>0$ について，$\lfloor\frac{k}{2}\rfloor+1>\lceil\frac{k-1}{4}\rceil+1$ であり，題意を満たす．

最後に $k>0$ かつ $C_{i,d}$ が t 個の loose subdomain を含む場合について考える．サイズが $k'<k$ である domain について，$\mathsf{extdom}_{d'}(F_{i'})$ に対応する区間は $\left\lceil\frac{k'-1}{4}\right\rceil+1$ 個の LZ 分解の境界を含むと仮定する．t 個の loose subdomain $\mathsf{dom}_{d_1}(F_{i_1}),\ldots,\mathsf{dom}_{d_t}(F_{i_t})$ について，$\mathsf{dom}_{d_h}(F_{i_h})$ のサイズを k_h とする $(1\le h\le t)$.

式 (2) の第 2 項について，帰納法の仮定より $n_h=\lceil\frac{k_h-1}{4}\rceil+1$ である．ここで，各 loose subdomain のサイズ k_h の和について考える．loose subdomain のサイズの和は全体の項数 k から各 cluster に含まれる domain と loose subdomain を除いたものと一致する．最左の cluster の domain の数を l とする．最左以外の cluster には，対応する loose subdomain が 1 つ存在し，それらを合計した値を考える．最右の cluster の domain の数は d_t-d-1 であり，それに loose subdomain の数を加え，全体で d_t-d である．その他の cluster については d_h が cluster の domain の数に loose subdomain を加えた値である．したがって，$\sum_{h=1}^{t}k_h=k-l-\sum_{h=1}^{t-1}d_h-(d_t-d)$ が成り立つ．

式 (2) の第 3 項について，各 cluster は group であるので，補題 8 を用いる．最左の cluster は domain の数が l 個である．最右の cluster は domain の数が d_t-d-1 であり，$\mathsf{dom}_d(F_i)$ も group として数えることができるので，全体で d_t-d 個である．その他の cluster については d_h-1 が group として数えられる domain の数であり，さらに補題 8 より，d_h が 2 以上のとき d_h-2 個の LZ 分解の境界が存在する．括弧の中が真のとき 1，偽のときに 0 を表す関数 [ ] を用いて以下のように記述できる．

$$S=\left\lfloor\frac{l-1}{2}\right\rfloor+\sum_{h=1}^{t-1}\left\lfloor\frac{1}{2}\{d_h-1-[d_h>1]\}\right\rfloor+\left\lfloor\frac{d_t-d-1}{2}\right\rfloor \tag{3}$$

ここで，$t-1$ 個の cluster を domain の個数によって 2 つの集合

$$T_1=\{h\mid d_h\ge 3,h\in[1,t-1]\}$$
$$T_2=\{h\mid d_h<3,h\in[1,t-1]\}$$

にわける．また，任意の自然数 k について $\lfloor\frac{k}{2}\rfloor>\frac{k}{2}-\frac{1}{2}$ が成り立つので，式 (3) の第 2 項を T_1,T_2 を用いて以下のように変形する．

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

よって，S は以下のように書ける．

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

以上より，式 (2) は次のように変形できる．

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

l および d_t-d の値によって場合分けを行う．

(i) $l=1$ のとき，

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

(ii) $l>1$ かつ $d_t-d-1=1$ のとき，

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

(iii) $l > 1$ かつ $d_t - d - 1 > 1$ のとき，

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

以上より，題意が示された．

上記補題を用いて，定理 1 が示される．

証明 (定理 1 の証明)．文字列 s を 1-domain の連結 $s = \mathsf{extdom}_{d_1}(F_{i_1}) \cdots \mathsf{extdom}_{d_t}(F_{i_t})$ で表せる．ただし，$i_t = m$ である．$\mathsf{dom}_1(F_{i_h})$ のサイズを k_h としたとき，$\mathsf{extdom}_1(F_{i_h})$ は補題 9 より $\left\lceil \frac{k_h - 1}{4} \right\rceil + 1$ 個の LZ 分解の境界を含む．いま，$\sum_{h=1}^{t} k_h = m - t$ であるので，

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

が成り立つ． □

5. おわりに

本稿では，Lyndon 分解の項数 m と自己参照あり LZ 分解の項数 z について $m < 4z$ を満たすことを示した．本稿で与えた証明は，Kärkkäinen らの自己参照なし LZ 分解に対する証明 [7] のテクニックを応用している．今回得られた結果は，タイトな関係であることは示されていない．よりタイトな関係を得ることが今後の課題の一つであるといえる．

参考文献

[1] H. Bannai, T. I, S. Inenaga, Y. Nakashima, M. Takeda, and K. Tsuruta. The “runs” theorem. *SIAM Journal on Computing*, 46(5):1501–1514, 2017.

[2] G. Chen, S. J. Puglisi, and W. F. Smyth. Lempel–ziv factorization using less time & space. *Mathematics in Computer Science*, 1(4):605–623, Jun 2008.

[3] 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.

[4] M. Crochemore, L. Ilie, and L. Tinta. Towards a solution to the “runs” conjecture. In P. Ferragina and G. M. Landau, editors, *Combinatorial Pattern Matching*, pages 290–302, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[5] M. Crochemore, C. S. Iliopoulos, T. Kociumaka, R. Kundu, S. P. Pissis, J. Radoszewski, W. Rytter, and T. Walen. Near-optimal computation of runs over general alphabet via non-crossing LCE queries. *CoRR*, abs/1606.08275, 2016.

[6] P. Gawrychowski, T. Kociumaka, W. Rytter, and T. Waleń. Faster longest common extension queries in strings over general alphabets. 02 2016.

[7] J. Kärkkäinen, D. Kempa, Y. Nakashima, S. J. Puglisi, and A. M. Shur. On the Size of Lempel-Ziv and Lyndon Factorizations. In *34th Symposium on Theoretical Aspects of Computer Science (STACS 2017)*, volume 66, pages 45:1–45:13, 2017.

[8] R. Kolpakov and G. Kucherov. Finding maximal repetitions in a word in linear time. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, FOCS ’99, pages 596–, Washington, DC, USA, 1999. IEEE Computer Society.

[9] D. Kosolobov. Computing runs on a general alphabet. *CoRR*, abs/1507.01231, 2015.

[10] M. Lothaire. Combinatorics on words. *Addison-Wesley*, 1983.

[11] R. C. Lyndon. On burnside’s problem. *Transactions of the American Mathematical Society*, 77(2):202–215, 1954.

[12] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, May 1977.