全単射 Burrows–Wheeler 変換の圧縮感度について

ジョン ヒョダム and クップル ドミニク

概 要

Burrows–Wheeler 変換 (BWT) は，可逆データ圧縮手法として広く利用されており，さまざまな圧縮アルゴリズムや索引構造の基盤を形成している．しかし，BWT の全単射変種である BBWT におけるこのような変更の影響は，十分に解明されていない．本研究では，BWT と BBWT の感度には大幅に違いがあり，圧縮サイズの変化は編集の種類に応じて対数的増加や平方根的増加を示すことを明らかにした．

キーワード： 可逆データ圧縮，Burrows–Wheeler 変換，圧縮感度

1 はじめに

Burrows–Wheeler Transform (BWT) [3] は，bzip2 や索引構造 [4] などの様々なデータ圧縮技術の基盤として利用される手法である．Akagi らは圧縮手法の「感度」について研究し [1]，1 文字の置換・挿入・削除が圧縮サイズに与える影響を指標化した．感度は通常，圧縮サイズの増加量で定義されるが，サイズの比率を用いる場合もあり，指標解析について，LZ78 [7], lex-parse [10], BWT [6] などがある．しかし，全反射 BWT (BBWT) の差分感度については未解明の部分が多い．

2 研究目的

データはハードディスクの容量やコストを削減するために圧縮して保存される．その際，圧縮後のサイズを予測することで効率的に管理できる．しかし，中のわずかな違いで圧縮サイズが大幅に増加することがあり，こうしたパターンの把握はリスク回避に役立つ．本研究では，先行研究 [6] が扱った二進数アルファベット $\{\mathtt{a}, \mathtt{b}\}$ に対する一文字編集による圧縮サイズの増加を拡張し，#や c を含む 4 文字に適用する．さらに，アルファベット順序を変更しても適用し，BWT に加えて BBWT も応用する．本研究の目的は，圧縮サイズの変化が大きいパターンを解析し，BWT および BBWT の圧縮感度を特定することである．

3 BWT と圧縮指標 r

BWT とは，文字列 W の全ての巡回文字列を辞書順にソートし，各行の最後の文字を連結して得られる可逆圧縮手法である．巡回文字列とは，文字列の先頭を末尾に移動して生成される．辞書順は $\mathtt{\#} < \mathtt{a} < \mathtt{b} < \mathtt{c}$ で定義し，# が最小である．圧縮サイズの指標 $r(W)$ は $\mathrm{runs}(\mathrm{BWT}(W))$ で表され，$\mathrm{runs}(W)$ は W の最大の同一文字の連続個数を指す．なお，全ての巡回文字列の r の値は等しい [9].

3.1 r の $\Omega(\lg n)$ 倍増加

Fibonacci 文字列は $F_0 = \mathtt{b}$, $F_1 = \mathtt{a}$ の時，$F_k = F_{k-1}F_{k-2}$ で定義される．例えば，$F_2 = \mathtt{ab}$，$F_3 = \mathtt{aba}$，$F_4 = \mathtt{abaab}$．Fibonacci 数列 f_k は F_k の長さを表し，$f_0 = 1$，$f_1 = 1$，$f_k = f_{k-1} + f_{k-2}$ で与えられる．BWT 変換後の圧縮サイズ $r(F_k)$ は常に 2 であり，高い圧縮効率を持つ．また，F_k は回文 X_k を用いて表せる．偶数 k の場合，$F_k = X_k\mathtt{ab}$ であり，X_2 は空文字列，$X_4 = \mathtt{aba}$．さらに，X_k は k が偶数なら $X_k = X_{k-1}\mathtt{ba}X_{k-2}$，奇数なら $X_k = X_{k-2}\mathtt{ba}X_{k-1}$ が成り立つ．

定理 1 $r(F_{2k}^b\#) = 2k + 2$.

先行研究 [6] は，F_{2k} の最後の文字を削除した F_{2k}^b の圧縮サイズが $r(F_{2k}^b) = 2k$ であると証明した．本研究では，F_{2k} の最後の文字を辞書順で小さい # に置換した $F_{2k}^b\mathtt{\#}$ の圧縮サイズを示す．F_{2k}^b 内で最小の接頭辞を持つ文字列 v を用いると，v の巡回文字列間の順序は # を追加しても変わらない．新たな r は $\mathtt{\#}v$ と $v\mathtt{\#}$ の巡回文字列のみで発生するため，$r(F_{2k}^b\mathtt{\#}) = r(v) + 2 = r(F_{2k}^b) + 2 = 2k + 2$ となる．

3.2 r の $\Omega(\sqrt{n})$ 差分感度

$W_k = \left(\prod_{i=2}^{k-1} \mathtt{ab}^i\mathtt{aaab}^i\mathtt{aba}^{i-2}\right)\mathtt{ab}^k\mathtt{a}$ と定義され，長さ n は $\Theta(k^2)$ である．$r(W_k) = 6k - 12$ であるが，先行研究 [6] は W_k の最後の一文字を修正すると r が $6k - 12 + \Theta(\sqrt{n})$ になることを証明した．ここでは a，b を反転した $\overline{W_k} = \left(\prod_{i=2}^{k-1} \mathtt{ba}^i\mathtt{bbba}^i\mathtt{bab}^{i-2}\right)\mathtt{ba}^k\mathtt{b}$ を定義し，$r(\overline{W_k}) = 6k - 12$ であることを事前に確認した．

定理 2 $r(W_k^b\#) - r(W_k) = 2k - 5 = \Theta(\sqrt{n})$.

W_k の最後の a を#に置換した $W_k^b\mathtt{\#}$ の圧縮サイズは，$r(W_k)$ より $2k - 5$ 増加する．これは，$2 \le i \le k - 1$ の範囲で# の影響により，$\mathtt{b}^i\mathtt{\#}$ と $\mathtt{b}^i\mathtt{a}$ で始まる巡回文字列が繰り返し現れるためである．その結果，元の W_k の BWT は ba

のみを繰り返すのに対し，$W_k^b\#$ は baba を繰り返す．この違いにより $\Theta(2k)$ の追加の r が発生し，圧縮サイズが増加する．

定理 3 $r(\overline{W_k^b}\mathtt{c}) - r(\overline{W_k}) = 2k - 5 = \Theta(\sqrt{n})$.

文字を反転しても，圧縮サイズは $2k-5$ 増加する．$2 \le i \le k-1$ の範囲では，$\mathtt{a}^i\mathtt{b}$ と $\mathtt{a}^i\mathtt{c}$ で始まる巡回文字列が繰り返し出現する．その結果，$\overline{W_k}$ の BWT は ba のみを繰り返すのに対し，$\overline{W_k^b}\mathtt{c}$ は baba を繰り返すため，圧縮サイズが $\Theta(2k)$ 増加する．

4 BBWT と圧縮指標 ρ

文字列 S は S の巡回文字列の中に辞書式順で最小であると，**Lyndon 語**と呼ぶ [8]．**Lyndon 巡回文字列**とは，文字列 S の巡回文字列の中の最小の文字列を指す．S が Lyndon 語なら $r(S) = \rho(S)$ [9] が成り立つが，そうでない場合 S は Lyndon 要素に分解でき，その操作を **Lyndon 分解**と呼ぶ．**全単射 BWT**(**BBWT**) [5] は，Lyndon 分解後，それぞれの Lyndon 要素の巡回文字列を作って，それらをソートし，最後の文字を連結して得られる．BBWT の圧縮サイズの指標は $\rho(S) = \mathrm{runs}(\mathrm{BBWT}(S))$ である．S が Fibonacci 文字列であり異なる m 個の Lyndon 要素を持つと，$\rho(S) \ge m$ である [2].

4.1 ρ の $\Omega(\lg n)$ 倍増加

F_k の Lyndon 巡回文字列は偶奇に関わらず $\mathtt{a}X_k\mathtt{b}$ [11] であり，これを L_k とする．L_{2k} の末尾を削除したものを L_{2k}^b，末尾を x に置換したものを $L_{2k}^b x$ と定義する．L_{2k} は F_{2k} の巡回文字列なので，$\rho(L_{2k}) = 2$ であるが，一文字編集で圧縮サイズが増加する．そのときの BBWT の圧縮サイズの下限を証明する．

定理 4 $\rho(L_{2k}^b) \ge k$.

$L_{2k}^b = \mathtt{a}X_{2k}$ であり，X_{2k} の分解を用いて Lyndon 分解する．展開すると，$L_{2k}^b = \mathtt{a}X_{2k-1}\mathtt{ba}X_{2k-3}\mathtt{ba}X_{2k-4}$．ここで，$\mathtt{a}X_{2k-1}\mathtt{b}$ は L_{2k-1}，$\mathtt{a}X_{2k-3}\mathtt{b}$ は L_{2k-3} になる．残りの $\mathtt{a}X_{2k-4}$ も同様に分解され，新たな Lyndon 要素 L_{2k-5} を得る．この過程を繰り返すと，最終に $\mathtt{a}X_4 = \mathtt{a}X_3\mathtt{ba}X_2$ となり，$\mathtt{a}X_3\mathtt{b}$ は L_3，X_2は 空文字列なので $\mathtt{a} = L_1$ も Lyndon 要素になる．よって，L_{2k}^b の Lyndon 分解は $L_{2k-1}, L_{2k-3}..L_1$ となり，異なる Lyndon 要素の個数が ρ の下限なので，圧縮サイズの下限は k.

定理 5 $\rho(L_{2k}^b\mathtt{c}) \ge k$.

$L_{2k}^b\mathtt{c}$ は $L_{2k-1}, L_{2k-3}, \ldots, L_3$ に分解され，最後に $\mathtt{ac}$ が残るが，$\mathtt{ac}$ は Lyndon 要素であるので，全ての Lyndon 要素の個数は k であり，$L_{2k}^b\mathtt{c}$ の圧縮サイズの下限は k になる．

定理 6 $\rho(L_{2k}^b\mathtt{\#}) \geq k+1$.

$L_{2k}^b\mathtt{\#}$ は $L_{2k-1}, L_{2k-3}, \ldots, L_3$ に分解され，最後に $\mathtt{a}X_2\mathtt{\#} = \mathtt{a\#}$ が残る．ここで，$\mathtt{a} = L_1$ と $\mathtt{\#}$ が Lyndon 要素に追加され，圧縮サイズの下限は $k+1$.

定理 7 特定の位置に $\mathtt{\#}$ を追加すると $\rho \geq k$.

二つの異なる位置に $\mathtt{\#}$ を挿入すると ρ が増加する．Case 1: $L_{2k-1}, L_{2k-3}, \ldots, L_3$ を Lyndon 要素とし，$\mathtt{a}X_2\mathtt{b} = \mathtt{ab}$ の間に $\mathtt{\#}$ を挿入することで，$\mathtt{a} = L_1$ と $\mathtt{\#b}$ の Lyndon 要素が得られる．挿入位置は $f_{2k}-2$ であり，圧縮サイズは $k+1$ 以上である．Case 2: X_{2k-1} を分解し，$\mathtt{a}X_{2k-3}\mathtt{ba}X_{2k-2}\mathtt{b}$ より L_{2k-3} を Lyndon 要素に追加する．分解を繰り返して L_3 まで Lyndon 要素を追加する．残りの $\mathtt{a}X_4\mathtt{b}$ の $\mathtt{a}$ と X_4 の間に挿入すると，$\mathtt{a} = L_1$ になり $\mathtt{\#}X_4$ 以降は Lyndon 要素なので圧縮サイズの下限は k である．

4.2 ρ の $\Omega(\sqrt{n})$ 差分感度

W_k の Lyndon 巡回文字列 C_k は $\mathtt{a}^{k-2}\mathtt{b}^k\mathtt{a}\cdot\left(\prod_{i=2}^{k-2}\mathtt{ab}^i\mathtt{aaab}^i\mathtt{aba}^{i-2}\right)\cdot\mathtt{ab}^{k-1}\mathtt{aaab}^{k-1}\mathtt{ab}$ であり，$\rho(C_k) = 6k-12$ になる．

定理 8 $\rho(C_k^b) - \rho(C_k) = 2k-5 = \Theta(\sqrt{n})$.

文字列 C_k^b は次のように定義される： $\mathtt{a}^{k-2}\mathtt{b}^k\mathtt{a}\cdot\left(\prod_{i=2}^{k-2}\mathtt{ab}^i\mathtt{aaab}^i\mathtt{aba}^{i-2}\right)\cdot\mathtt{ab}^{k-1}\mathtt{aaab}^{k-1}\mathtt{a}$. C_k^b は C_k の最後の $\mathtt{b}$ を削除した文字列なので，$\mathtt{a}$ の最大連続長は $k-1$ であるが，C_k^b の接頭辞とは違うので，C_k^b は Lyndon 分解される：

1. D_k = 前半部分（Lyndon 語），
2. $\mathtt{a}$（単独の Lyndon 要素).

$\rho(D_k) = 8k-18$ である．$\mathtt{a}$ は D_k より小さいため，$\rho(C_k^b) = \rho(\mathtt{a}) + \rho(D_k) = 1+8k-18 = 8k-17$．最後の文字を削除すると ρ は $2k-5$ 増加する．

定理 9 $\rho(C_k^b\#) - \rho(C_k) = 2k-4 = \Theta(\sqrt{n})$.

$C_k^b\mathtt{\#}$ は定理 8 に $\mathtt{\#}$ を追加した文字列である．$\mathtt{\#} < \mathtt{a} < \mathtt{b}$ であるため，定理 8 の Lyndon 分解に，$\mathtt{\#}$ が追加される形である．$\rho(C_k^b\#) = \rho(\mathtt{\#}) + \rho(\mathtt{a}) + \rho(D_k) = 1+1+8k-18 = 8k-16$．最後の文字を $\mathtt{\#}$ に置換すると ρ は $2k-4$ 増加する．

定理 10 $\rho(C_k\mathtt{c}) - \rho(C_k) = 2k = \Theta(\sqrt{n})$.

$C_k\mathtt{c}$ は C_k の最後に $\mathtt{c}$ を追加した文字列である．$\mathtt{a}$ の最大連続長は $\mathtt{c}$ の追加より $k-2$ に減り，$C_k\mathtt{c}$ の接頭辞なので $C_k\mathtt{c}$ は Lyndon 語になる．よって，$\rho(C_k\mathtt{c}) = r(C_k\mathtt{c})$ が成り立つ．ρ を増加させる巡回文字列は $\mathtt{b}^i\mathtt{a}$ $(2 \le i \le k-2)$ から始まるものであり，それらの末尾の $\mathtt{abab}$ の繰り返しにより，ρ は $2k = \Theta(\sqrt{n})$ 増加する．

定理 11 $\rho(C_k^b\mathtt{c}) - \rho(C_k) = 2k - 1 = \Theta(\sqrt{n})$.

$C_k^b\mathtt{c}$ は C_k の最後の文字を $\mathtt{c}$ に置換した文字列である．定理 10 と同様に，$C_k^b\mathtt{c}$ は Lyndon 語であるが，$\mathtt{c}$ から始まる巡回文字列が存在しないため，ρ は $2k - 1 = \Theta(\sqrt{n})$ 増加する．

5 おわりに

本研究では，BBWT の圧縮感度に注目した．文字の編集が圧縮サイズに与える影響を解析した．圧縮サイズの増加に対して対数的あるいは平方根的な影響を持つことを示した．これにより，BWT や BBWT を用いたデータ圧縮において，小さな編集が圧縮効率に及ぼすリスクをより正確に評価できるようになった．今後の研究では，より実用的なデータセットへの適用を進めることで，圧縮方法の性質をより理解できると考えられる．

謝辞

本研究は JSPS 科研費 JP23H04378 の助成と山梨県若手研究者奨励事業費補助金 2291 の支援を受けたものである．

参考文献

[1] Tooru Akagi, Mitsuru Funakoshi, and Shunsuke Inenaga. Sensitivity of string compressors and repetitiveness measures. *Inf. Comput.*, 291:104999, 2023.

[2] Christina Boucher, Davide Cenzato, Zsuzsanna Lipták, Massimiliano Rossi, and Marinella Sciortino. r-indexing the eBWT. In *Proc. SPIRE*, volume 12944 of *LNCS*, pages 3–12, 2021.

[3] Michael Burrows and David J. Wheeler. A block sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation, Palo Alto, California, 1994.

[4] Paolo Ferragina and Giovanni Manzini. Opportunistic data structures with applications. In *Proc. FOCS*, pages 390–398, 2000.

[5] Joseph Yossi Gil and David Allen Scott. A bijective string sorting transform. *ArXiv 1201.3077*, 2012.

[6] Sara Giuliani, Shunsuke Inenaga, Zsuzsanna Lipták, Giuseppe Romana, Marinella Sciortino, and Cristian Urbina. Bit catastrophes for the Burrows–Wheeler transform. In *Proc. DLT*, volume 13911 of *LNCS*, pages 86–99, 2023.

[7] Guillaume Lagarde and Sylvain Perifel. Lempel-Ziv: a "one-bit catastrophe" but not a tragedy. In *Proc. SODA*, pages 1478–1495, 2018.

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

[9] Sabrina Mantaci, Antonio Restivo, Giovanna Rosone, and Marinella Sciortino. An extension of the Burrows–Wheeler transform. *Theor. Comput. Sci.*, 387(3):298–312, 2007.

[10] Yuto Nakashima, Dominik Köppl, Mitsuru Funakoshi, Shunsuke Inenaga, and Hideo Bannai. Edit and alphabet-ordering sensitivity of lex-parse. In *Proc. MFCS*, volume 306 of *LIPIcs*, pages 75:1–75:15, 2024.

[11] Kalle Saari. Lyndon words and Fibonacci numbers. *J. Comb. Theory, Ser. A*, 121:34–44, 2014.