

Compression Sensitivity of the Bijective Burrows-Wheeler transform

Hyodam Jeon, [Dominik Köppl](#)

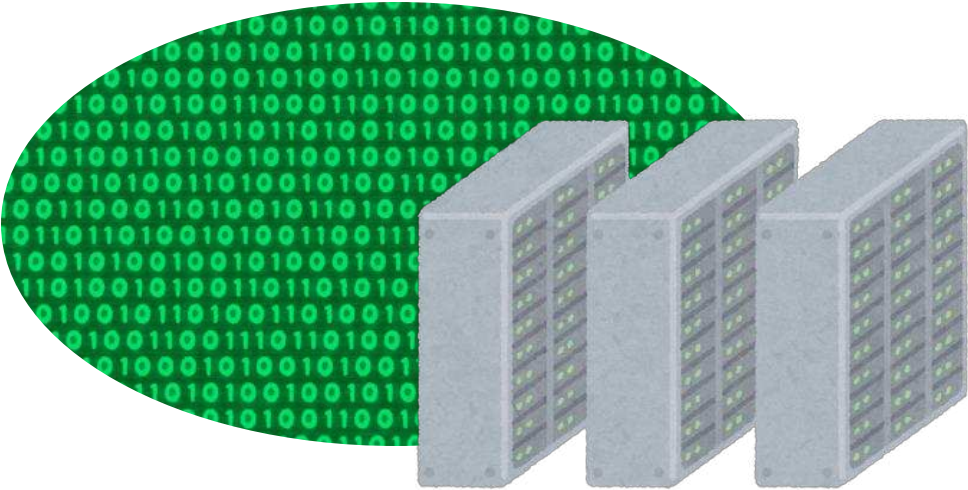
Published in:
Mathematics 2025,
13, 1070



University of Yamanashi
Computer science and engineering

International Workshop on Discrete
Mathematics and Algorithms 2025

Background



Setting: Need to store large text collections in compressed form

Examples:

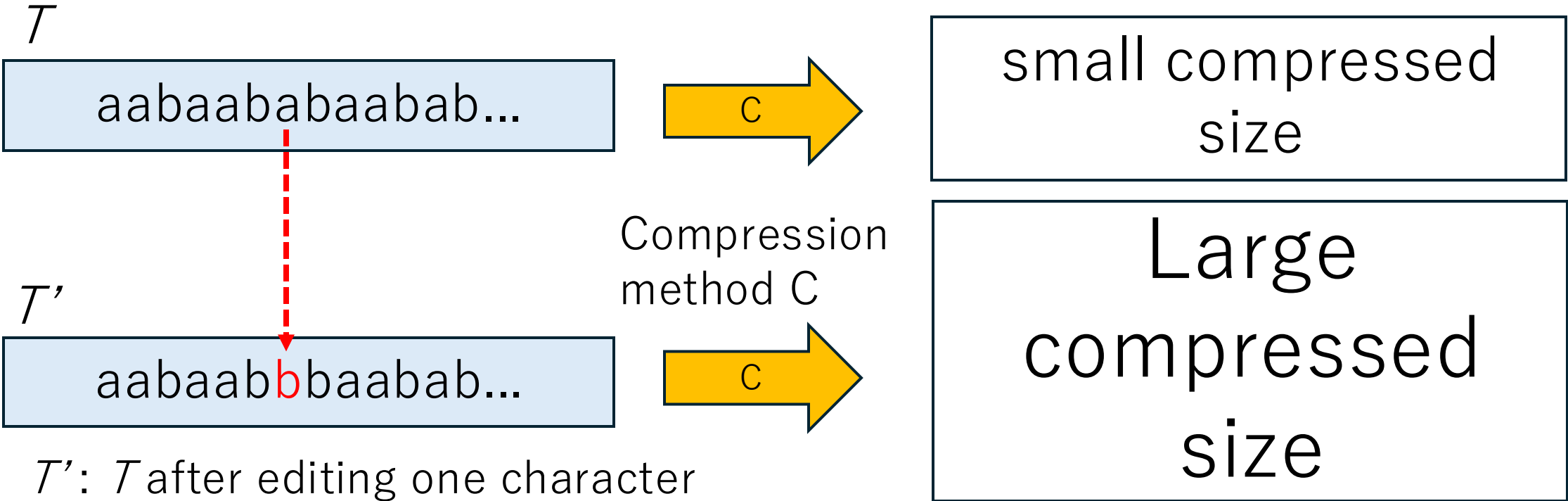
- Biological data (ncbi),
- Source code (github),
- Websites (wayback machine)

- Expect that similar data can be stored compressed with similar sizes
- However: small changes of the input can cause large difference in compressed size!
- Difference can cause economic loss!
- Question: how bad can it get?



Research objective

How much impact has a single character edit of the input?



T' : T after editing one character

Edit: insert/deletion/exchange

Here: exchange a with b

compression sensitivity =

What is the max. difference between the compressed sizes of the edited text $C(T')$ and the original $C(T)$?

Related work

Compression sensitivity has been studied for various compressors

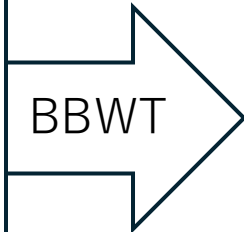
Compression method	Related work
Lempel-Ziv 78 (LZ78) (gif image compression)	Lagarde&Perifel '18
Lempel-Ziv 77 (gzip/zip/etc.)	Akagi+ '23
BWT (bzip2, compressed indexes)	Giuliani+ '23
lex-parse	Nakashima+ '24
string attractor, bidirectional macro scheme	Fujie+ '24

however: the sensitivity of the **bijjective BWT (BBWT)**
has not yet been studied

Clustering effect of the BBWT

\mathcal{T} , $r(\mathcal{T}) = 110$

```
aabaababaabaababaababaababaab  
aababaababaabaababaababaababa  
abaababaababaabaababaababaaba  
baabaababaababaabaababaababaa  
babaabaababaabab
```



$\text{BBWT}(\mathcal{T})$, $r(\text{BBWT}(\mathcal{T})) = 2$

```
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb  
bbbbbbbbbbbbbbbbbbbbbbbbbaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
aaaaaaaaaaaaaaaaaaaaaaaa
```

- BBWT arranges characters by previous context such that characters in the same context are grouped together
- $\text{BBWT}(\mathcal{T})$ is likely to be better run-length compressible than the input
- **Repetitiveness measure $r(\mathcal{T})$** : number of maximal consecutive character occurrences in text \mathcal{T} (size of the run-length encoding/compression)

Bijjective BWT (BBWT) [Gil&Scott '12]

Lyndon words and factors

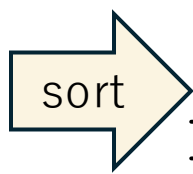
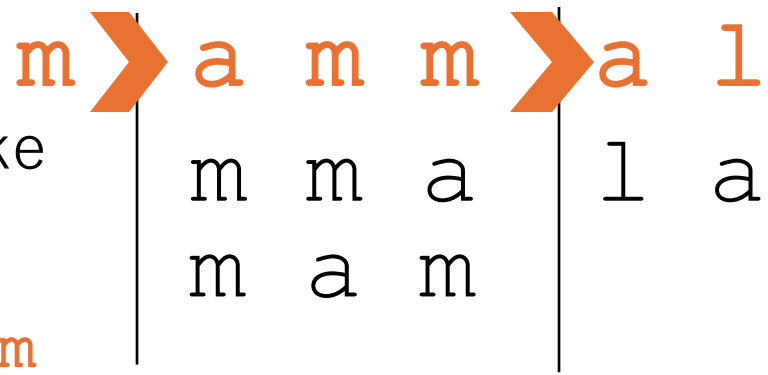
- A word S is **Lyndon** if it is smaller than all its conjugates
- If S is not Lyndon, we can factorize it uniquely into Lyndon factors $L_1, L_2, L_3 \dots L_m$ such that the Lyndon factors are in lex. decreasing order $L_1 \geq L_2 \geq \dots \geq L_m$

Order: $a < l < m$

a	l	m	a	m	m
a	m	m	a	l	m
l	m	a	m	m	a
m	a	l	m	a	m
m	a	m	m	a	l
m	m	a	l	m	a

Bijjective BWT

- sort the conjugates of all Lyndon factors of S and take the last character of each



a	l	a	l	a	l
a	m	m	a	m	m
l	a	l	a	l	a
m	a	m	m	a	m
m	m	a	m	m	a
m	m	m	m	m	m

- $BBWT(mamma1) = lmamam$
- compression measure : $\rho(S) = r(BBWT(S))$

Inverting the BBWT

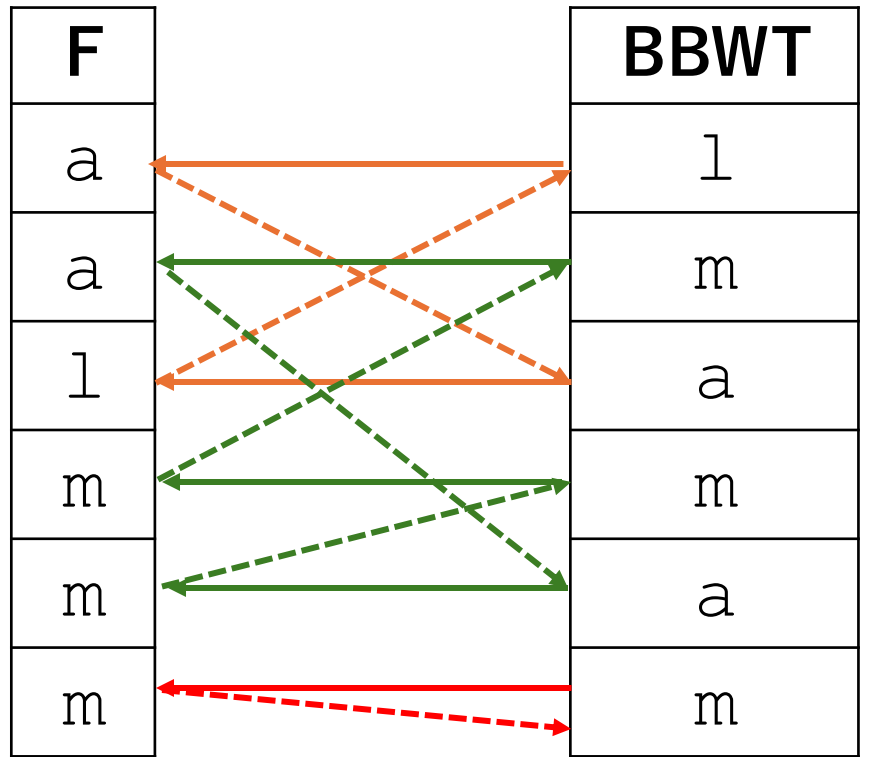
To obtain S from BBWT(S), we follow the cycles in the BBWT → obtain S's Lyndon factors

First sorted characters **F**

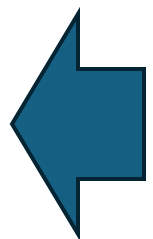
BBWT(mamma l)

a	l	a	l	a	l	a	l	a	l	a	l
a	m	m	a	m	m	a	m	m	a	m	m
l	a	l	a	l	a	l	a	l	a	l	a
m	a	m	m	a	m	m	a	m	m	a	m
m	m	a	m	m	a	m	m	a	m	m	a
m	m	m	m	m	m	m	m	m	m	m	m

Loop to retrieve Lyndon factors



m
a m m
a l



a l
a m m
m
Sort in descending order

Sensitivity : Results

Red: new results, (*x) : edit operation with character x
 Black: known results due to Giuliani+ '23

Input text \ method	Edit operation	BWT	BBWT
Fibonacci word • $r(\text{BWT}(F_{2k}))=2$ • $\rho(\text{Lyndon conjugate of } F_{2k})=2$	remove last letter	$2k$	$\geq k$
	change last letter	$2k + 2(*a)$	$\geq k + 1(*\#)$
		$2k + 2(*\#)$	$\geq k(*c)$
	insert at specific positions	-	$\geq k$ $\geq k + 1$

Fibonacci

Fibonacci words


- $F_0 = b, F_1 = a, F_k = F_{k-1}F_{k-2}$
- $F_2 = ab$
- $F_3 = aba$
- $F_4 = abaab$
- $F_5 = abaababa$
- $F_6 = abaababaabaab$
- $F_7 = abaababaabaababaabaababa$

Fibonacci numbers

- $f_0 = 1, f_1 = 1, f_k = f_{k-1} + f_{k-2}$
- $f_2 = 2$
- $f_3 = 3$
- $f_4 = 5$
- $f_5 = 8$
- $f_6 = 13$
- $f_7 = 21$

Fibonacci

Fibonacci words

- $F_0 = b, F_1 = a, F_k = F_{k-1}F_{k-2}$
- $F_2 = ab$
- $F_3 = \boxed{a}ba$ X_k : palindrome
- $F_4 = \boxed{aba}ab$
- $F_5 = \boxed{abaab}aba$ 
- $F_6 = \boxed{abaababa}ab$
- $F_7 = \boxed{abaababaabaaba}ba$

$$F_k = X_k ab \text{ if } k \text{ is even}$$
$$= X_k ba \text{ if } k \text{ is odd}$$

Fibonacci numbers

- $f_0 = 1, f_1 = 1, f_k = f_{k-1} + f_{k-2}$
- $f_2 = 2$
- $f_3 = 3$
- $f_4 = 5$
- $f_5 = 8$
- $f_6 = 13$
- $f_7 = 21$

Logarithmic increase of ρ

What we did: Remove last character of Lyndon conjugate $aX_{2k}b$ of F_{2k} .

Theorem : $\rho(aX_{2k}b) = 2$ but $\rho(aX_{2k}) \geq k$

Proof Idea: Number of distinct Lyndon factors is a lower bound of ρ

$$a \boxed{X_{2k}} = aX_{2k}$$

Green parts are not Lyndon, blue parts are Lyndon factors

Logarithmic increase of ρ

What we did: Remove last character of Lyndon conjugate $aX_{2k}b$ of F_{2k} .

Theorem : $\rho(aX_{2k}b) = 2$ but $\rho(aX_{2k}) \geq k$

Proof Idea: Number of distinct Lyndon factors is a lower bound of ρ

$$\begin{array}{l}
 a \text{ [green bar } X_{2k}] = aX_{2k} \\
 a \text{ [blue bar } X_{2k-1}b] a \text{ [green bar } X_{2k-2}] = a X_{2k-1} b a X_{2k-2}
 \end{array}$$

(2k-1)th Fibonacci word

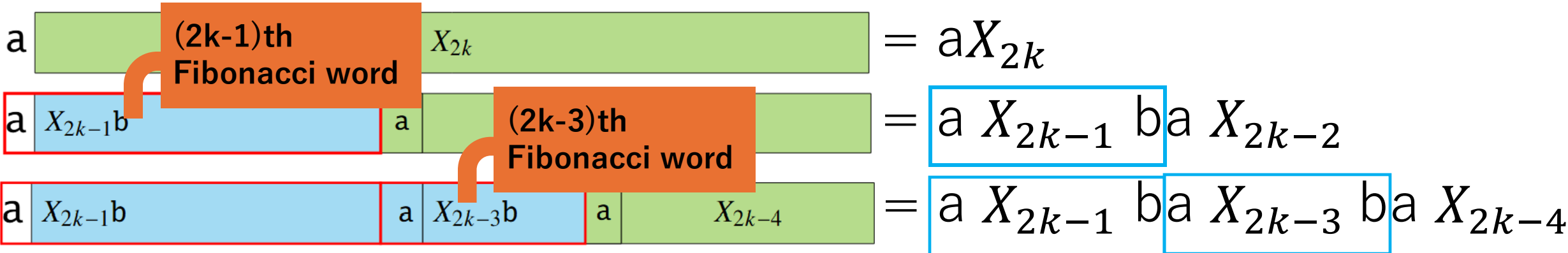
Green parts are not Lyndon, blue parts are Lyndon factors

Logarithmic increase of ρ

What we did: Remove last character of Lyndon conjugate $aX_{2k}b$ of F_{2k} .

Theorem : $\rho(aX_{2k}b) = 2$ but $\rho(aX_{2k}) \geq k$

Proof Idea: Number of distinct Lyndon factors is a lower bound of ρ



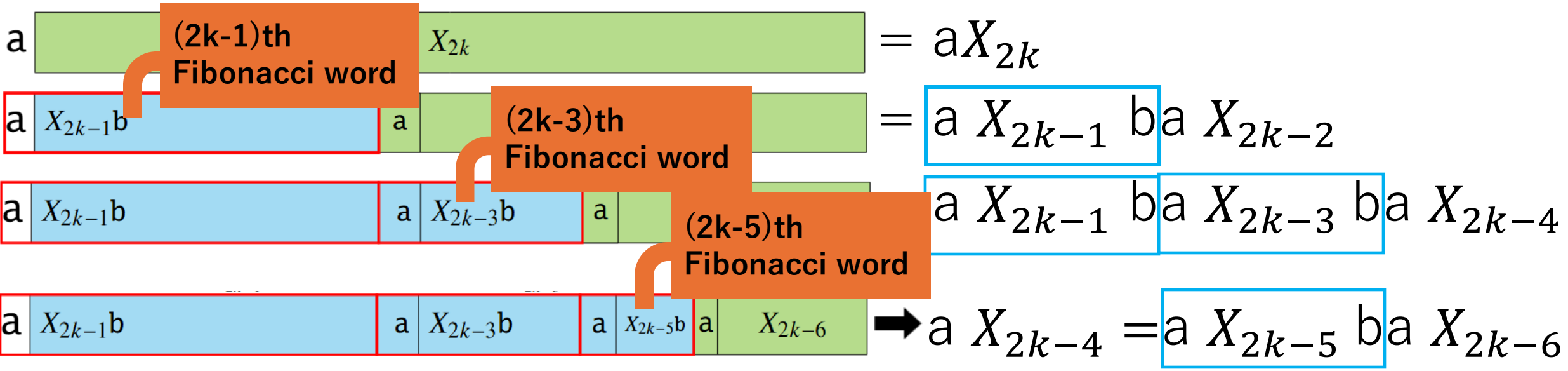
Green parts are not Lyndon, blue parts are Lyndon factors

Logarithmic increase of ρ

What we did: Remove last character of Lyndon conjugate $aX_{2k}b$ of F_{2k} .

Theorem : $\rho(aX_{2k}b) = 2$ but $\rho(aX_{2k}) \geq k$

Proof Idea: Number of distinct Lyndon factors is a lower bound of ρ



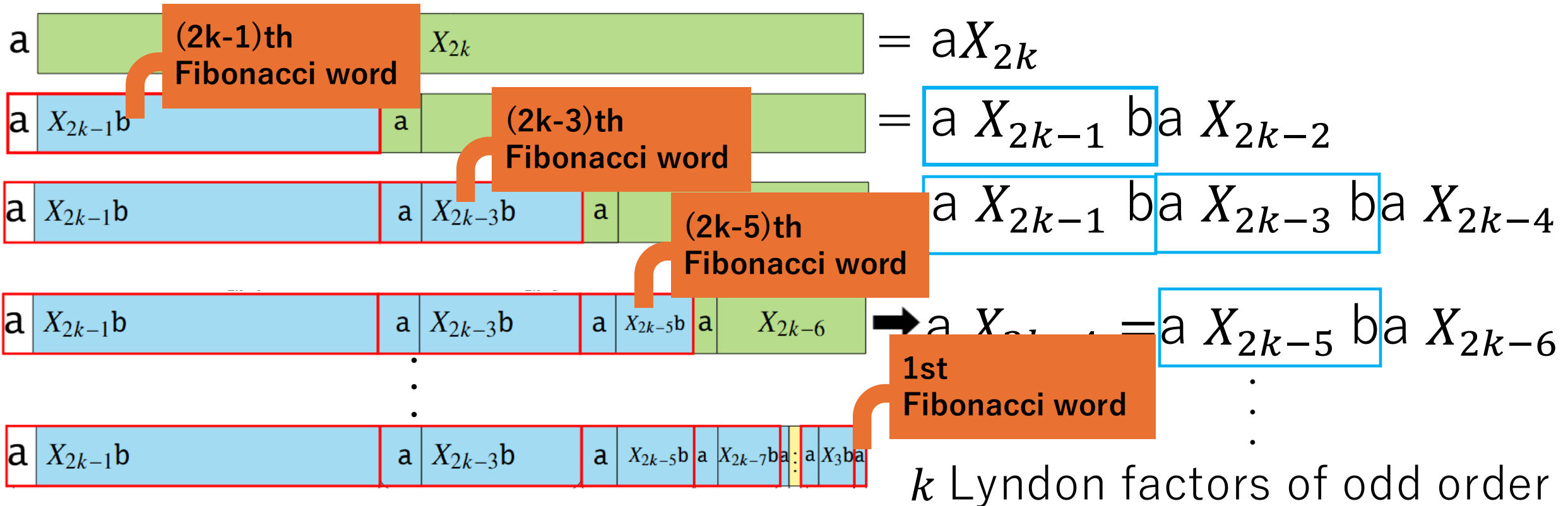
Green parts are not Lyndon, blue parts are Lyndon factors

Logarithmic increase of ρ

What we did: Remove last character of Lyndon conjugate $aX_{2k}b$ of F_{2k} .

Theorem : $\rho(aX_{2k}b) = 2$ but $\rho(aX_{2k}) \geq k$

Proof Idea: Number of distinct Lyndon factors is a lower bound of ρ



Green parts are not Lyndon, blue parts are Lyndon factors

Recap and future work

recap

1. compression measure ρ can increase by a factor of $\Theta(\log n)$
2. not shown but proved: size can increase by $+\Theta(\sqrt{n})$

→ a single edit can change the compressed size dramatically

→ **BWT and BBWT are not well-designed measures for compressibility**

future work

- experiments suggest that $\rho(S) = 2k$ in the proof of the previous slides, but we could not prove it
- improve the bounds: is there a non-trivial upper bound for ρ ?
- determine the sensitivity of other compression methods