A separation of γ and b via Thue-Morse words

Hideo Bannai[1], Mitsuru Funakoshi[2], Tomohiro I[3],
Dominik Köppl[1], Takuya Mieno[2], Takaaki Nishimoto[4]
[1]Tokyo Medical and Dental University, [2]Kyushu University, [3]Kyushu Institute of Technology, [4]RIKEN AIP

Relations between repetitiveness measures

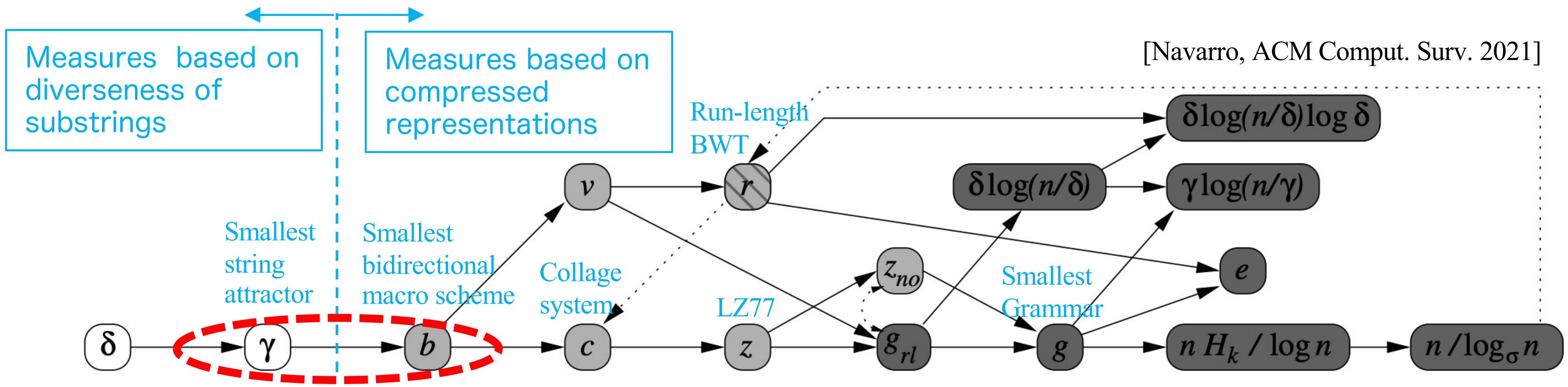


Fig 9. "Relations between the compressibility measures. A solid arrow from X to Y means that $X = O(Y)$ for all string families. For all solid and dotted arrows, there are string families where $X = o(Y)$, **with the exceptions of $\gamma \rightarrow b$** and $c \rightarrow z$. "

Our focus is on the relation $\boldsymbol{\gamma \rightarrow b}$**.**

- γ: size of smallest string attractor [Kempa&Prezza 2018]
  - String attractor: a set of positions of a string, such that any substring has an occurrence that contains one of the positions.
- b: size of smallest Bidirectional Macro Scheme (BMS) [Storer&Szymanski 1982]
  - BMS: A partitioning of a string into phrases, such that each phrase of length > 1 can be copied from another occurrence in the string, and the source of each position can be traced back to a phrase of length 1.

For Thue-Morse words, [Kutsukake et al. 2020] proved $\gamma = 4$ and conjectured $b = \Theta(\log N)$, which would imply $\gamma = o(b)$.

We prove Kutsukake et al.'s conjecture and for the first time **show the gap between $\boldsymbol{b}$ and $\boldsymbol{\gamma}$, i.e., $\boldsymbol{\gamma}$ is not always reachable by dictionary compression**.

Bidirectional Macro Scheme

[Storer&Szymanski 1982]

- $((l_1, s_1), \ldots, (l_k, s_k)) \in (\mathbf{N} \times (\Sigma \cup \mathbf{N}))^k$ is a BMS for string w if:
  - $|w| = l_1 + \ldots + l_k$ (partitioning into phrases)
  - Let $p_i = \sum_{1 \leq j < i} l_j$
    If $l_i = 1$, $s_i = w[p_i]$ (*ground* phrase)
    If $l_i > 1$, s_i is an integer (*source* of phrase)
    s.t. $w[s_i..s_i+l_i-1] = w[p_i..p_i+l_i-1]$
- Except for ground phrases, the sources of the phrases implicitly define source positions $f(i)$ for all positions i.

	0	1	2	3	4	5	6	7
	a	b	b	a	b	a	a	b
f	**3**	4	**4**	5	$\perp$	$\perp$	**3**	4

((2, 3), (2, 4), (1, b), (1, a), (2, 3))

size 5

- A BMS is *valid*, if w can be reconstructed
  $\Leftrightarrow f$ is acyclic: i.e., for any i, $f^j(i) = \perp$ for some $j \geq 1$.

NP hard to compute smallest valid BMS of string

Thue-Morse words

[Prouhet 1851][Thue 1906][Morse 1921]

Definition:

The n-th Thue-Morse word t_n ($n \geq 1$) is:

$$t_n = \mu^n(\mathsf{a})$$

where μ is a morphism defined by
$\mu(\mathsf{a}) = \mathsf{ab}$, $\mu(\mathsf{b}) = \mathsf{ba}$.

$t_1 = \mathsf{ab}$

$t_2 = \mathsf{abba}$

$t_3 = \mathsf{abbabaab}$

$t_4 = \mathsf{abbabaabbaababba}$

$t_5 = \mathsf{abbabaabbaababbabaababbaabbabaab}$

…

$|t_n| = 2^n$

Main results

<u>Theorem</u>

For any $n \geq 2$, the size of a smallest BMS for the n-th Thue-Morse word t_n is $n+2$.

<u>Corollary</u>

For any $\gamma \geq 4$, there exists a family of strings with smallest string attractor size γ s.t. the size b of a smallest BMS of the string and its length N satisfies

$$b = \Theta(\gamma \log (N/\gamma))$$

where N is the length of each string.

Theorem 1 (Upper Bound)

For any $n \geq 2$, there exists a BMS of size n+2 for the n-th Thue-Morse word t_n.

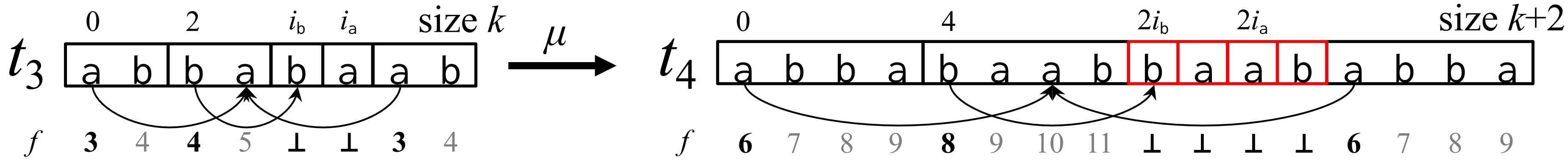
<u>Proof:</u>

Proof by induction.

There exists a BMS of size 2+2 = 4 for t_2 = abba.

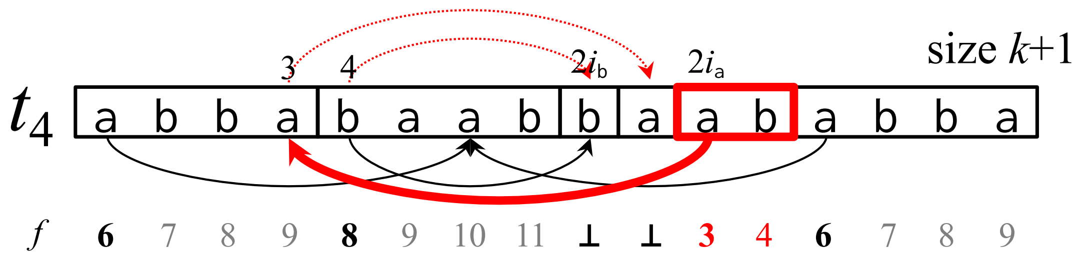
Given a BMS of size k for t_n, we show how to construct a BMS of size k+1 for t_{n+1}.

Size k BMS for t_n → Size k+1 BMS for t_{n+1}



1. Apply μ to each phrase, and double source positions.
   - with exception: for two ground phrases a, b, make 2 ground phrases each from μ(a) *and* μ(b): total 4 ground phrases.
   - In t_{n+1}, the parity (even/odd) of a position i and its source $f(i)$ are equal:
     - even position → even position
     - odd position → odd position

2. Merge ground phrases a, b created from μ(a), into a new phrase with source position 3.
   - This does not introduce cycles because a at pos 3 → $2i_b+1$
     b at pos. 4 → $2i_b$
   - # of phrases is $k + 1$.



Theorem 2 (Lower Bound)

For any $n \geq 2$, a smallest BMS of t_n has size $\geq n + 2$.

<u>Proof Idea:</u>

Go in the "opposite" direction as Theorem 1.

Proof by induction:

- Smallest BMS for t_2 has size $2 + 2 = 4$.
- Assume Theorem 2 holds for all integers up to some $n \geq 2$.

Seems difficult to do (if not impossible)

Given a BMS of size k for t_{n+1}, IF we can construct a BMS of size $k - 1$ for t_n, then,

$k \geq n + 3$ must hold, since $k - 1 \geq n + 2$.

Theorem 2.

For any $n \geq 2$, a smallest BMS of t_n has size $\geq n + 2$.

Proof Idea **(modified)**:

Go in the "opposite" direction as Theorem 1.

Proof by induction:

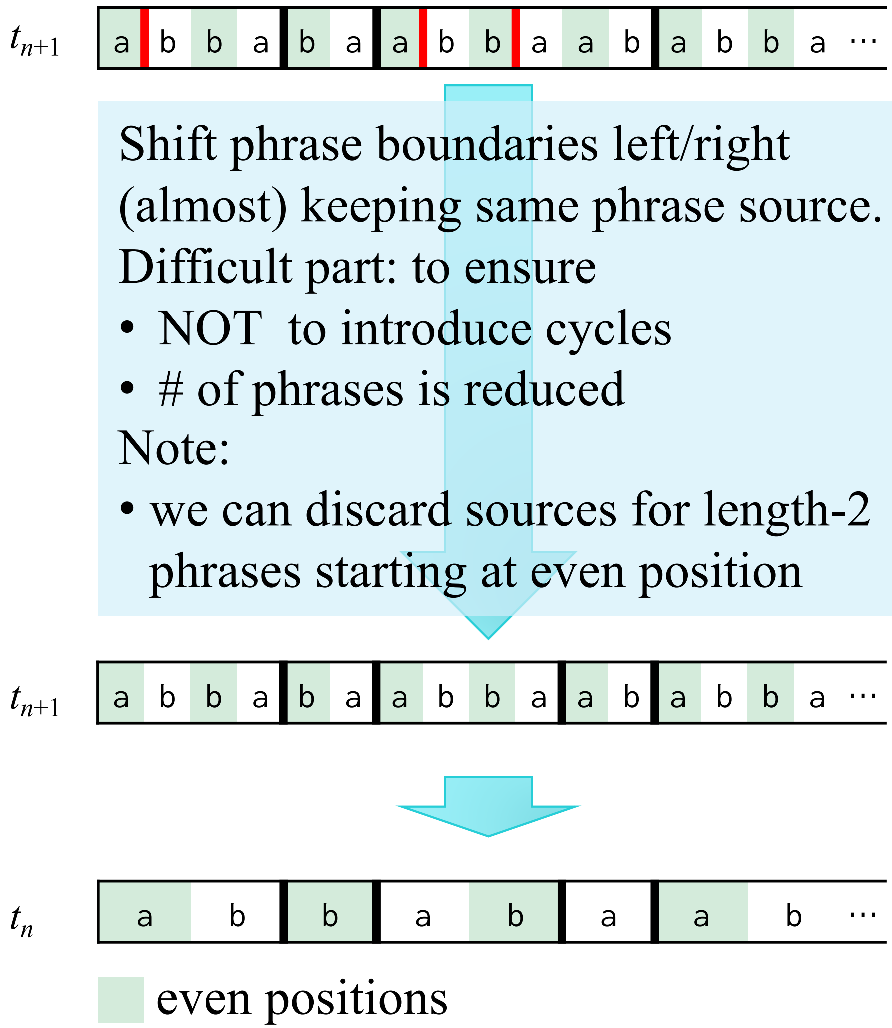
- Smallest BMSs for t_2, t_3, t_4 resp. have sizes 4, 5, 6.
- Assume Theorem 2 holds for all integers up to some $n \geq 4$.

Given a BMS of size k for t_{n+1}, ~~IF~~ we can construct a BMS of size $k - i$ for t_{n+1-i} for some $i \in \{ 1, 2, 3 \}$.

$k \geq n + 3$ must hold, since $k - i \geq (n + 1 - i) + 2 = n - i + 3$.

Size k BMS for t_{n+1} → Size k' BMS for t_n

Modify BMS for t_{n+1} in the following steps:



Shift phrase boundaries left/right (almost) keeping same phrase source.
Difficult part: to ensure

- NOT to introduce cycles
- # of phrases is reduced

Note:

- we can discard sources for length-2 phrases starting at even position

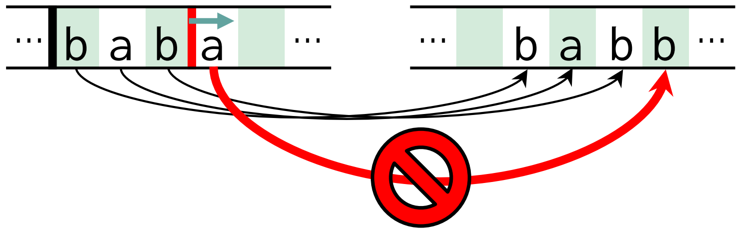
Those that don't start a phrase at an even position

1. Remove "**bad**" phrase boundaries
   - 1-1. Eliminate ground phrases
   - 1-2. Eliminate phrases aba, bab
   - 1-3. Eliminate remaining bad phrase boundaries
2. Apply inverse morphism μ^{-1} (halve phrases and sources)

… and apply this procedure at most 3 times.

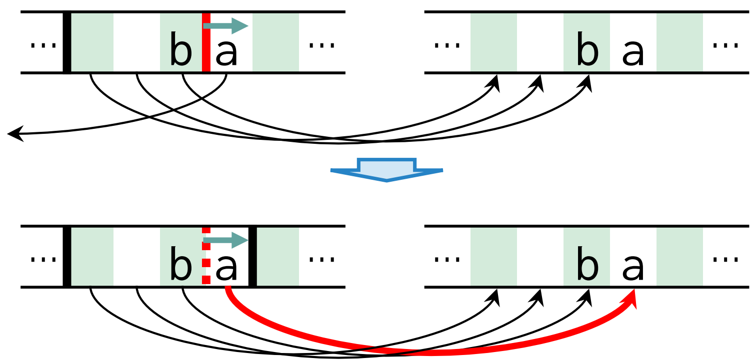
Problems when shifting a bad phrase boundary

If the parity (odd/even) of the source changes, we cannot always extend the phrase and keep the same source.

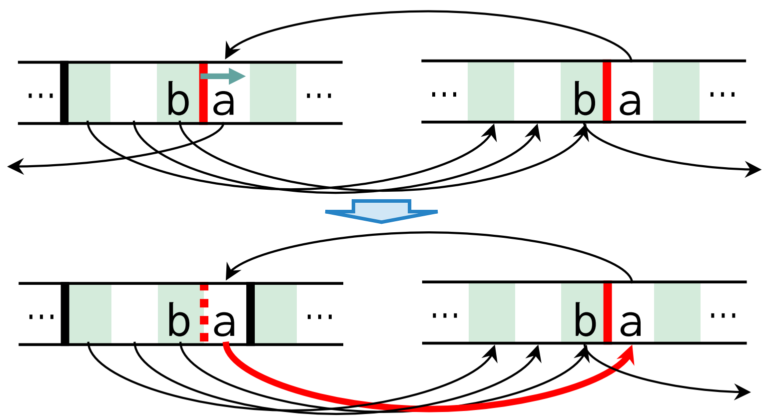


Problems when shifting a bad phrase boundary

If the parity of the source is the same,
we can keep the same phrase source, and
a bad phrase boundary can be shifted to extend the phrase.



However, cycles may be introduced.



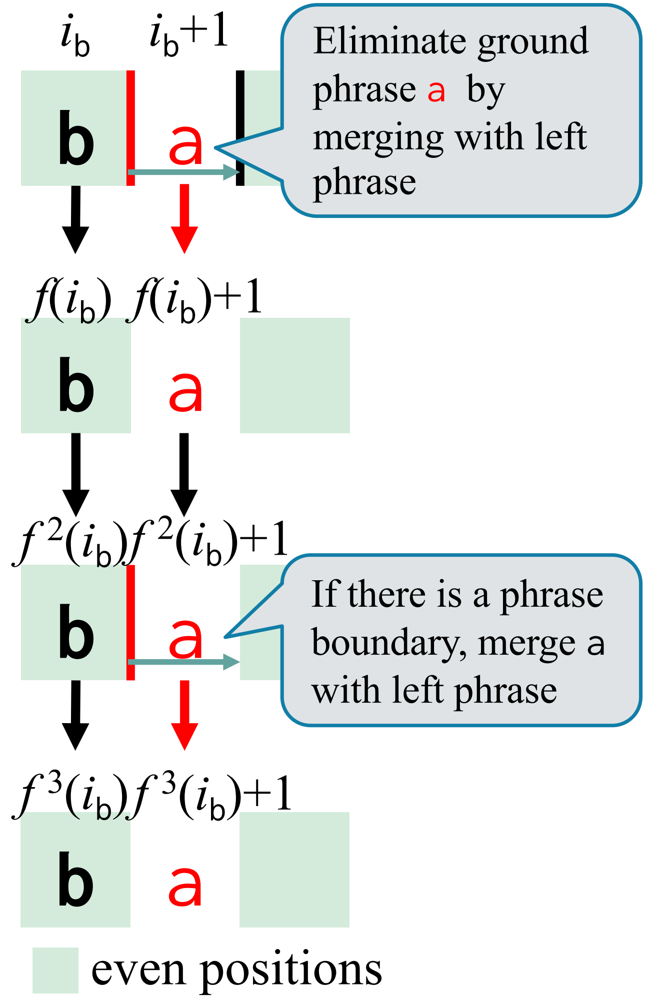
1-1. Eliminating ground phrases

Eliminating a ground phrase a at odd position ($i_b + 1$)

- If source $f(i_b)$ of left **b** is even
  - merge a with left phrase
    = update source of a to $f(i_b)+1$
- Repeat while source of **b** is even
  (There is always an a to its right)

These changes don't introduce cycles since **b** was not in a cycle.

What do we do, when the source of **b** is odd?



(Symmetric/analogous for eliminating ground phrase a/b at odd/even positions)

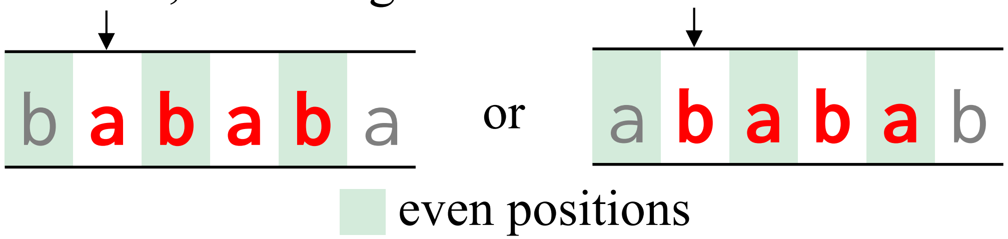
Key observation

<u>Parity Lemma</u>

The parity of occurrences in t_n can only change for the 6 strings:
a, b, ab, ba, aba, bab

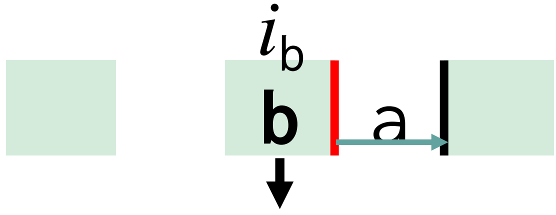
<u>Proof:</u>

1. aa and bb can occur only at odd positions.
   - Due to morphism (μ(a) = ab, μ(b) = ba), even positions start with either ab or ba.
2. abab and baba can only occur at even positions.
   - Otherwise, due to 1., the string would contain a cubes:



   - However, Thue-Morse words are known to be cube free.
3. The 6 strings are the only strings that do not contain aa, bb, abab or baba as substrings.

1-1. Eliminating ground phrases: Terminal Cases

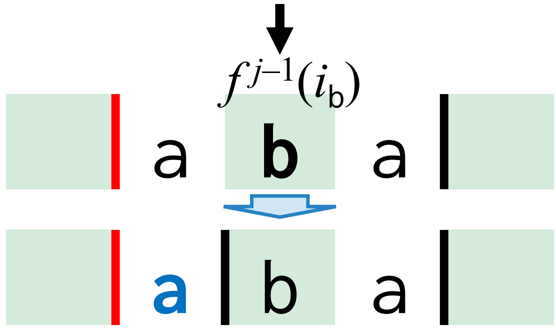


even positions

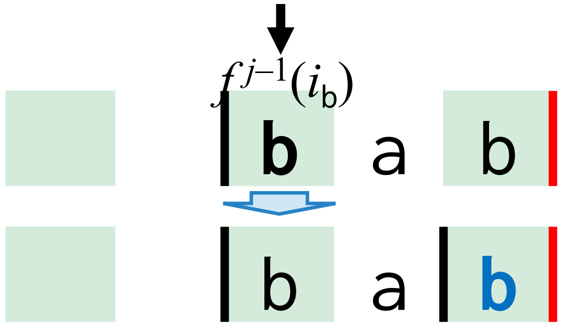
Let j be smallest integer s.t. $f^{j}(i_b)$ is odd (or is ⊥).

Only following cases due to parity lemma.
(occurrence of b in aba, bab, ba, ab, b)

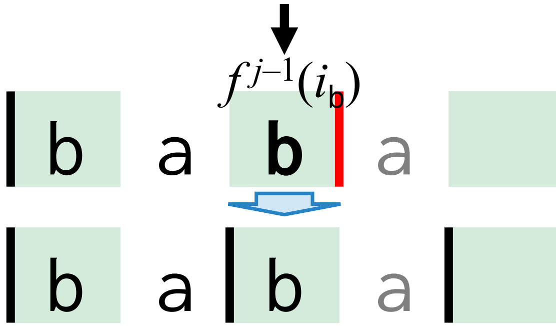
Procedure terminates, because
# of bad phrase boundaries strictly decreases.



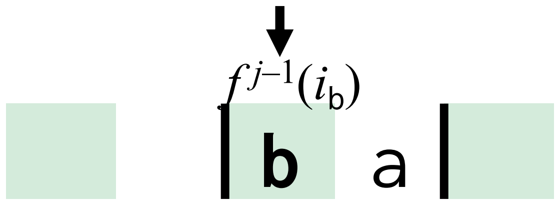
Done since phrase ba doesn't need source. Recurse to eliminate a.



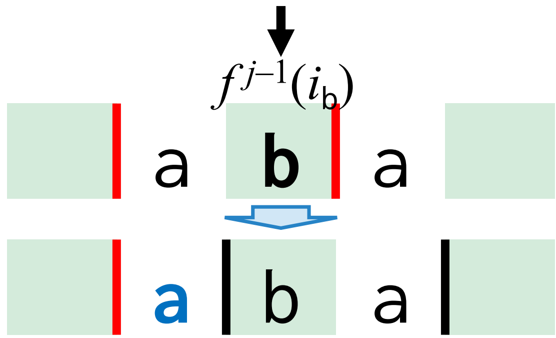
Done since phrase ba doesn't need source. Recurse to eliminate b.



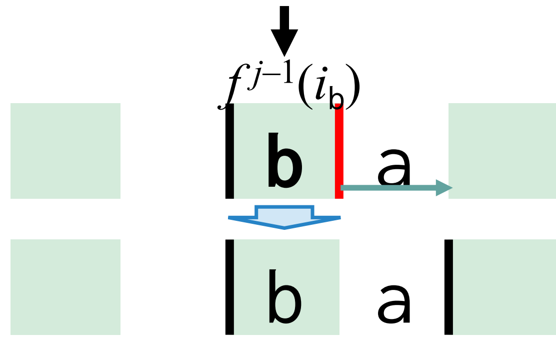
Done since phrase ba doesn't need source. Middle boundary only made the first time.



Done since phrase ba doesn't need source

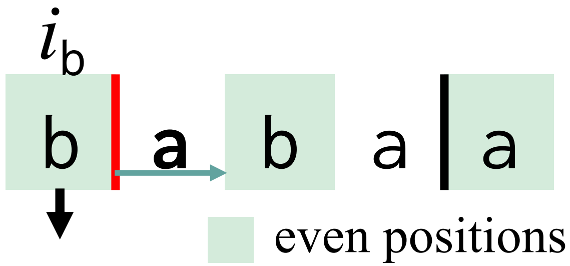


Done since phrase ba doesn't need source. Recurse to eliminate a.



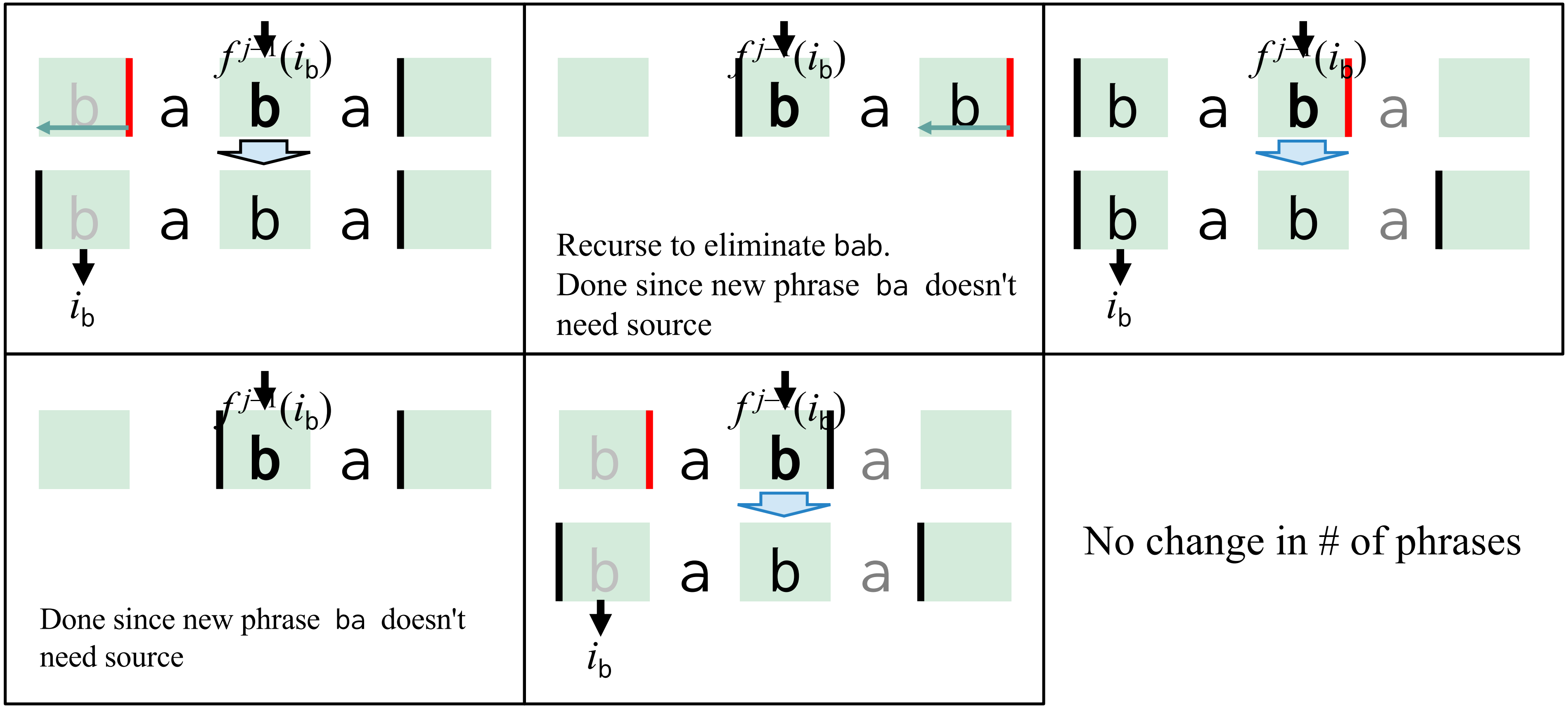
Done since phrase ba doesn't need source

1-2. Eliminating Phrases: aba, bab



Eliminate aba that starts at odd position.

- move bad boundary to truncate to ba and update source of a while source of left b is even.
- Terminal cases: when next source of b is odd.

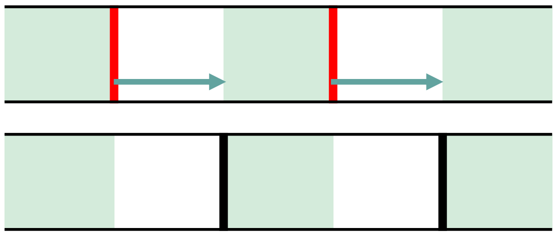


(Symmetric/analogous for eliminating phrase aba/bab at odd/even positions)

1-3. Eliminating remaining bad phrase boundaries

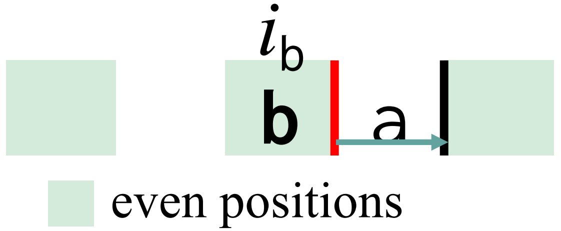
All remaining "bad" phrase boundaries can be removed by moving them to the right, keeping the sources of phrases because only cases are:

- length-2 phrases where both boundaries are bad

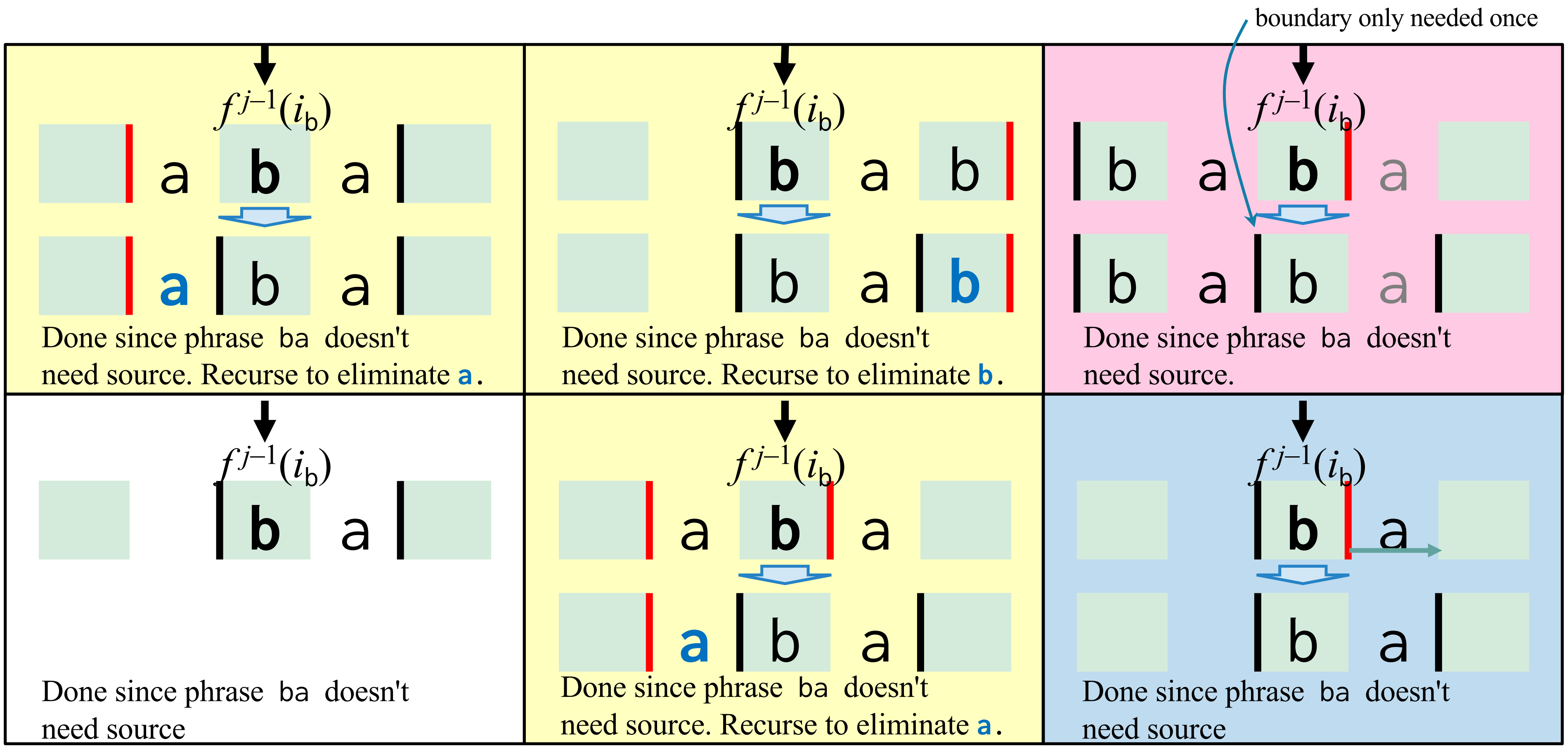


- phrases with bad boundaries whose occurrences always have the same parity (no phrases aba, bab or ground phrase)

Changes in # of phrases $\#_{\mathrm{tot}}$ in terms of # of ground phrases $\#_{\mathrm{g}}$



$\#_{\mathrm{g}}$: ± 0 $\#_{\mathrm{tot}}$: ± 0	$\#_{\mathrm{g}}$: -1 $\#_{\mathrm{tot}}$: ± 0 first time, -1 otherwise	Can also happen for abab
$\#_{\mathrm{g}}$: -1 $\#_{\mathrm{tot}}$: -1	$\#_{\mathrm{g}}$: -2 $\#_{\mathrm{tot}}$: -1	$k' \leq k - \lceil(\#_{\mathrm{g}} - 2)/2\rceil$



of phrases

	A
$\#_g$: ± 0 $\#_{tot}$: ± 0	$\#_g$: -1 $\#_{tot}$: ± 0 first time, -1 otherwise
$\#_g$: -1 $\#_{tot}$: -1	$\#_g$: -2 $\#_{tot}$: -1 (B)

Can also happen for abab

$$k' \le k - \lceil(\#_g - 2)/2\rceil$$

$$k_n \le k_{n+1} - \lceil(\#_g(n+1) - 2)/2\rceil$$

- If $k_n \le k_{n+1} - 1$, just choose $i = 1$ and we are done.
- If $k_n = k_{n+1}$, then $\#_g(n+1) = 2$ and case A was applied twice.
- Two phrases of ab and two phrases of ba are created. Therefore $\#_g(n) \ge 4$

$$k_{n-1} \le k_n - \lceil(\#_g(n) - 2)/2\rceil$$

$$\le k_n - 1 = k_{n+1} - 1$$

- If $k_{n-1} \le k_{n+1} - 2$, just choose $i = 2$ and we are done.
- If $k_{n-1} = k_n - 1$, then $\#_g(n) = 4$ and case A was applied twice, and case was B applied once. Therefore $\#_g(n-1) \ge 5$

$$k_{n-2} \le k_{n-1} - \lceil(\#_g(n-1) - 2)/2\rceil$$

$$\le k_{n-1} - 2 = k_{n+1} - 3$$

Summary

- Size of smallest BMS for t_n: $b(t_n) = n+2$
  Proof: given size k BMS for t_n, we can make
  - size $k + 1$ BMS for t_{n+1}
  - size $k - i$ BMS for t_{n-i} for some $i \in \{1, 2, 3\}$.
- Since $\gamma(t_n) = 4$ for any $n \geq 4$ [Kutsukake et al. 2020]
  $\{ t_n \mid n \geq 4 \}$ is a family of strings such that $\gamma = o(b)$
- Concatenating t_n over different binary alphabets gives, for any $\gamma \geq 4$, a family of strings such that: $b = \Theta(\gamma \log N/\gamma)$, where N is length of string.

Showed for the first time the gap between b and γ, i.e., γ is not always reachable by dictionary compression.