

# Bijjective Burrows Wheeler Transform - Open Problems -

Dominik Köppl  
StringMasters '21

definitions

# string transformations

Burrows-Wheeler Transform (BWT)

[Burrows,Wheeler '94]

Bijjective BWT (BBWT)

[Gil,Scott '12]

# BWT of bacabbabb

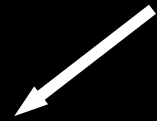
$T\$ = \text{bacabbabb}\$$

$\$ < a < b < c$

# BWT of bacabbabb

$T\$ = \text{bacabbabb\$}$

$\$ < a < b < c$



all suffixes

bacabbabb\$  
acabbabb\$  
cabbabb\$  
abbabb\$  
bbabb\$  
babb\$  
abb\$  
bb\$  
b\$  
\$

# BWT of bacabbabb

$T\$ = \text{bacabbabb\$}$

$\$ < a < b < c$

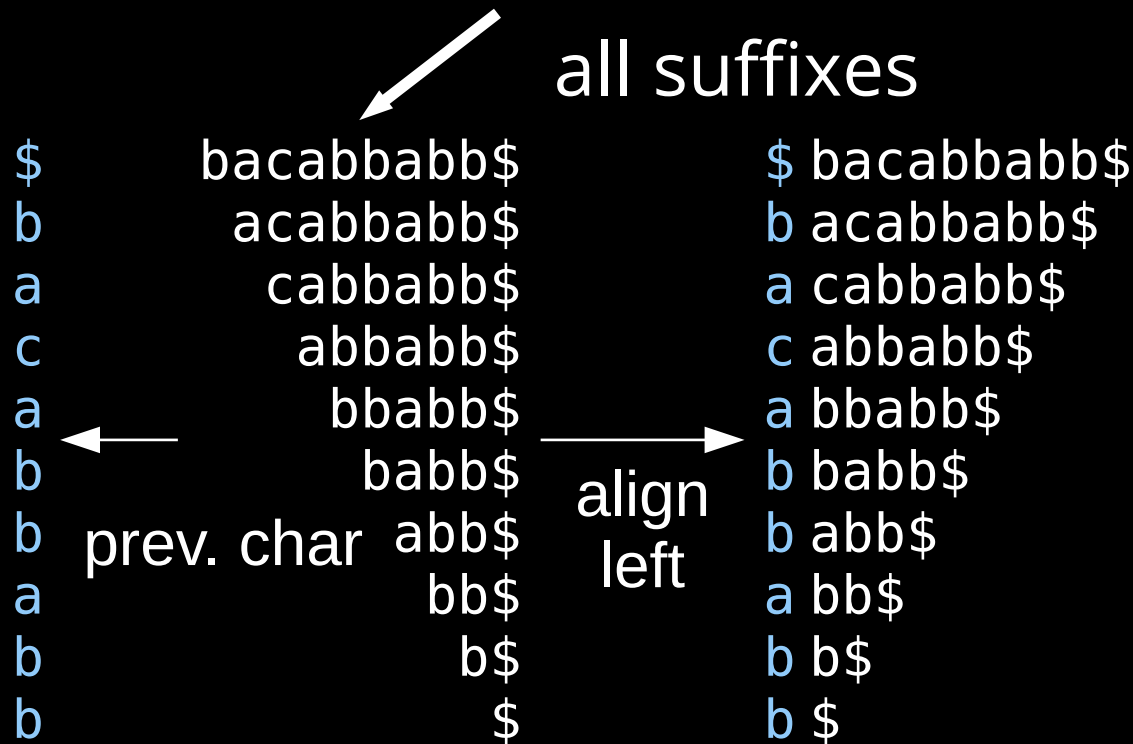
all suffixes

\$	bacabbabb\$
b	acabbabb\$
a	cabbabb\$
c	abbabb\$
a	bbabb\$
b	babb\$
b	prev. char abb\$
a	bb\$
b	b\$
b	\$

# BWT of bacabbabb

$T\$ = \text{bacabbabb\$}$

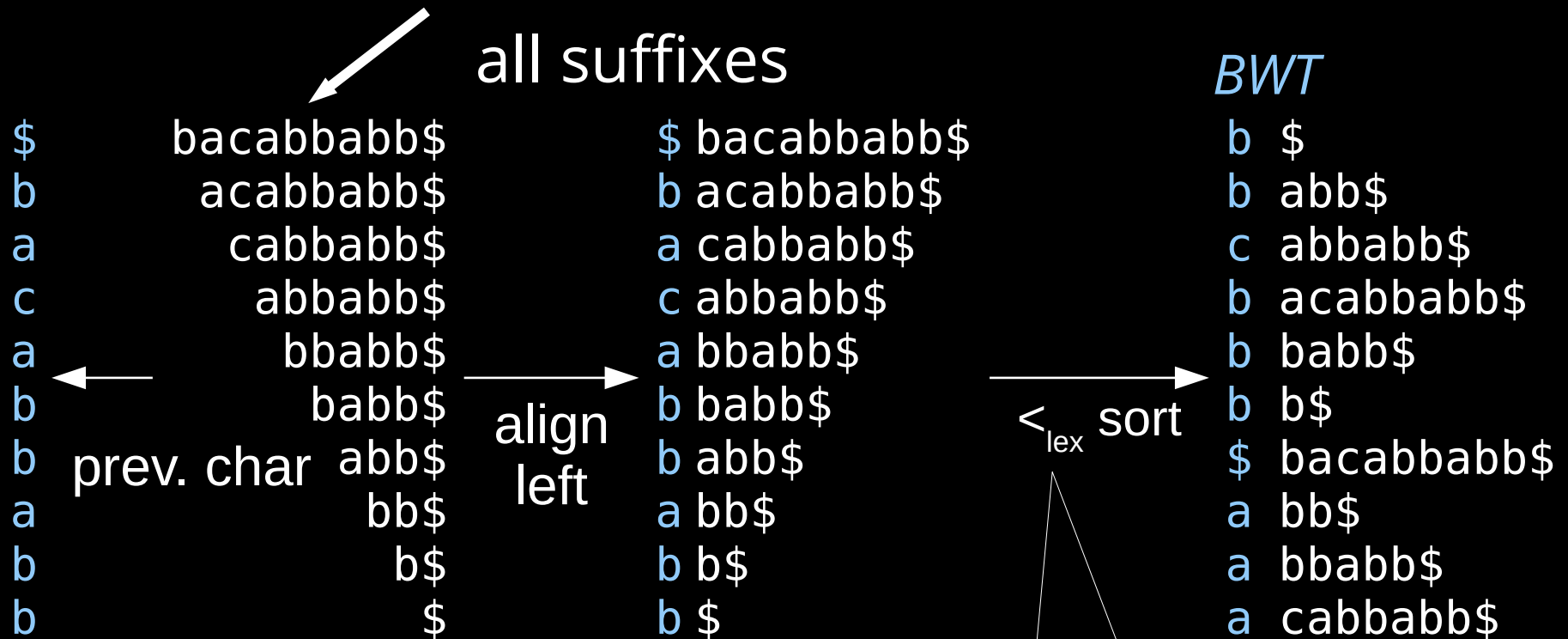
$\$ < a < b < c$



# BWT of bacabbabb

$T\$ = \text{bacabbabb\$}$

$\$ < a < b < c$





the BBWT is  
the BWT of  
the Lyndon factorization

with respect to  $\prec_{\omega}$

the BBWT is  
the BWT of  
the Lyndon factorization **1.**

with respect to  $\prec_{\omega}$  **2.**

# conjugates

- $T = T[1] T[2] \cdots T[n]$
- conjugates = cyclic shifts:
  - $T[1] T[2] \cdots T[n]$
  - $T[2] T[3] \cdots T[n] T[1]$
  - $\vdots$
  - $T[n] T[1] \cdots T[n-1]$

# Lyndon words

- a
- aabab

Lyndon word is smaller than

- every proper suffix
- every rotation

# Lyndon words

- a
- aabab

Lyndon word is smaller than

- every proper suffix
- every rotation

not Lyndon words:

- abaab (rotation aabab smaller)
- abab (abab not smaller than suffix ab)

# Lyndon factorization [Chen+ '58]

- input: text  $T =$ 

$T_1$	$T_2$	...	$T_t$
-------	-------	-----	-------
- output: factorization  $T_1 \dots T_t$  with
  - $T_x$  is Lyndon word
  - $T_x \geq_{\text{lex}} T_{x+1}$
  - factorization uniquely defined
  - linear time [Duval '88]

(Chen-Fox-Lyndon Theorem)

# example

$T = \text{bacabbabb}$

Lyndon factorization:  $\text{b} | \text{ac} | \text{abb} | \text{abb}$

–  $\text{b}$ ,  $\text{ac}$ ,  $\text{abb}$ , and  $\text{abb}$  are Lyndon

–  $\text{b} >_{\text{lex}} \text{ac} >_{\text{lex}} \text{abb} \geq_{\text{lex}} \text{abb}$

# $\prec_{\omega}$ order

- $u \prec_{\omega} w \iff uuuu\dots \prec_{\text{lex}} wwww\dots$
- $ab \prec_{\text{lex}} aba$
- $aba \prec_{\omega} ab$



# $\prec_{\omega}$ order

•  $u \prec_{\omega} w \iff uuuu\dots \prec_{\text{lex}} wwww\dots$

•  $ab \prec_{\text{lex}} aba$

ab**a**babab...

•  $aba \prec_{\omega} ab$

aba**a**baaba...

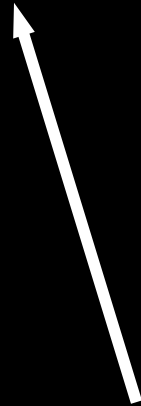
# BBWT of bacabbabb

b | ac | abb | abb

# BBWT of bacabbabb

b | ac | abb | abb

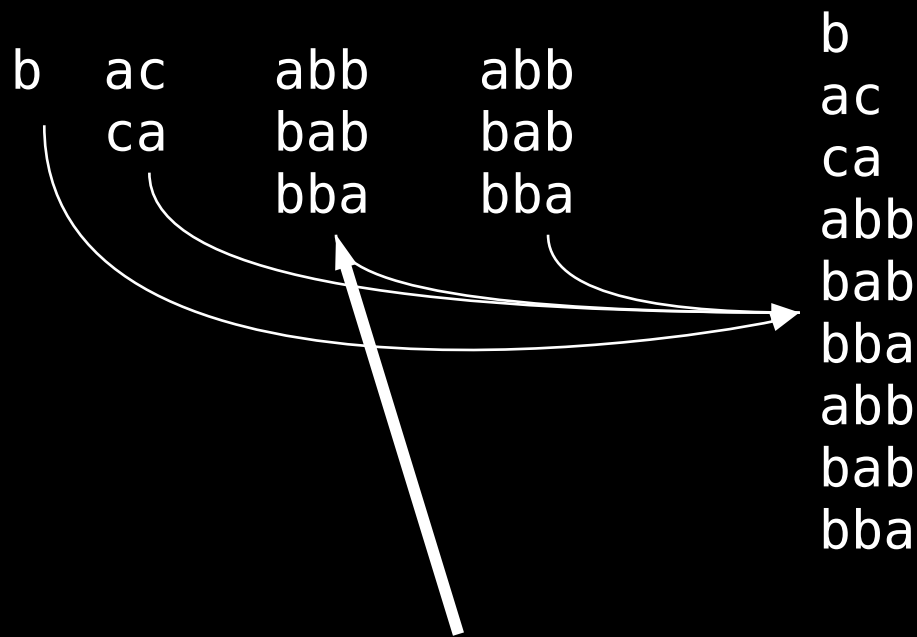
b	ac	abb	abb
	ca	bab	bab
		bba	bba



conjugates of all Lyndon factors

# BBWT of bacabbabb

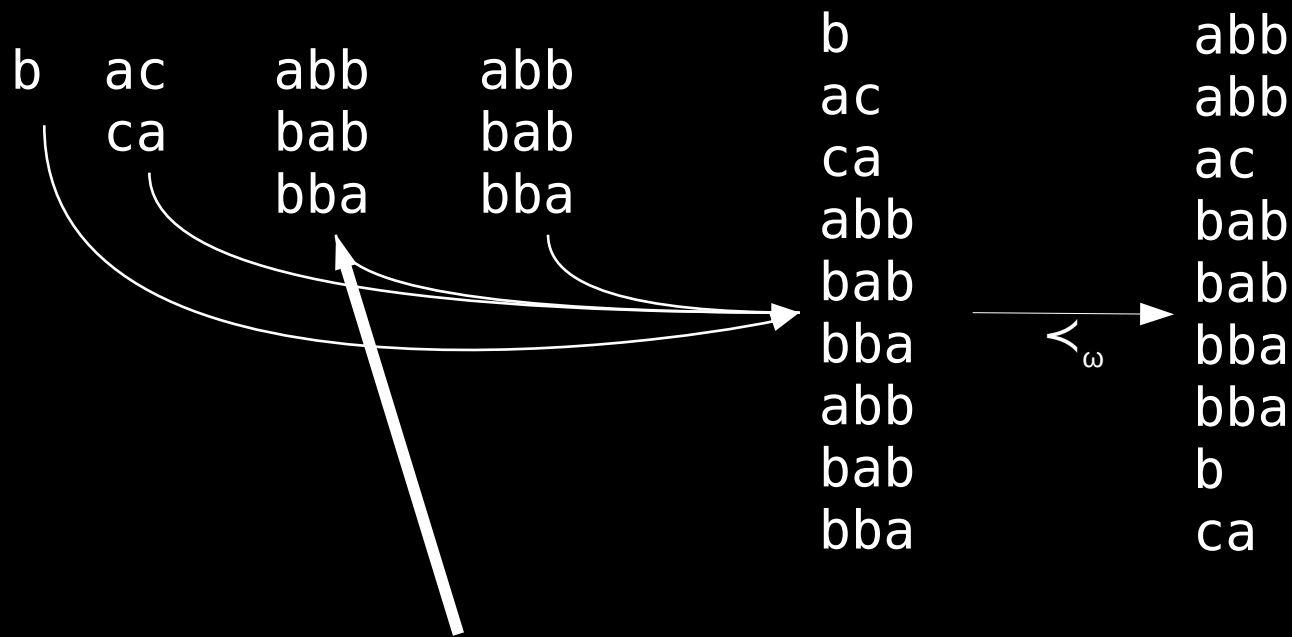
b | ac | abb | abb



conjugates of all Lyndon factors

# BBWT of bacabbabb

b | ac | abb | abb

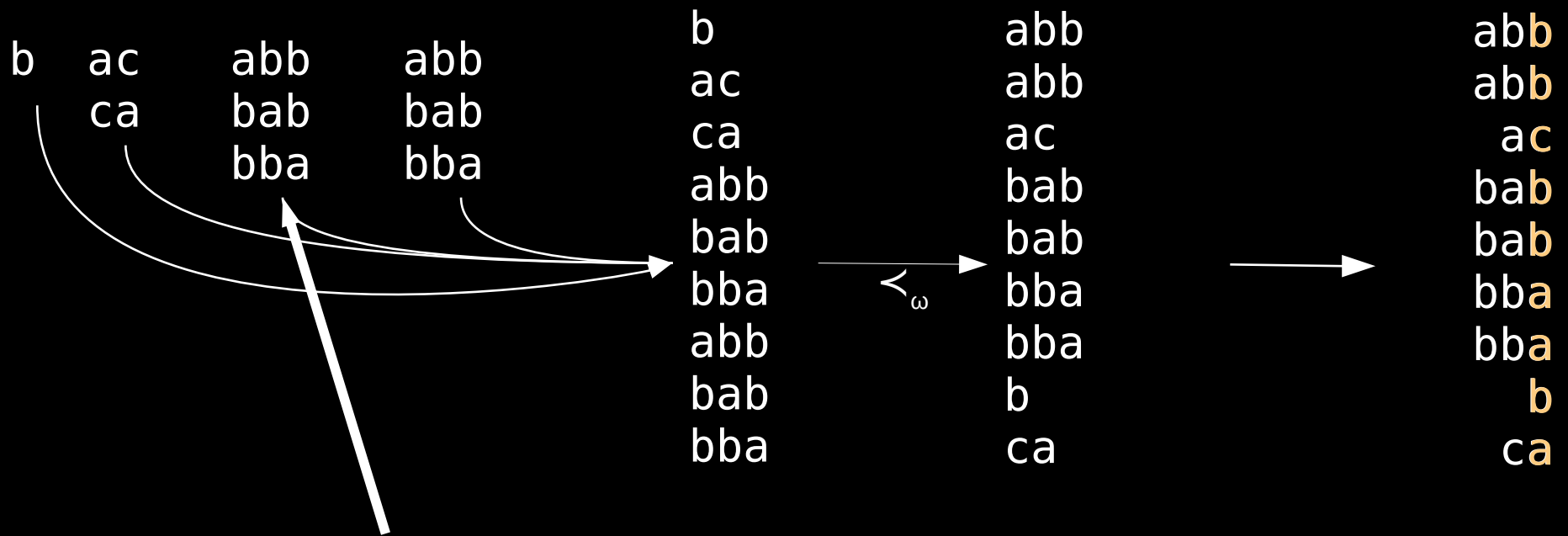


conjugates of all Lyndon factors

# BBWT of bacabbabb

b | ac | abb | abb

*BBWT*



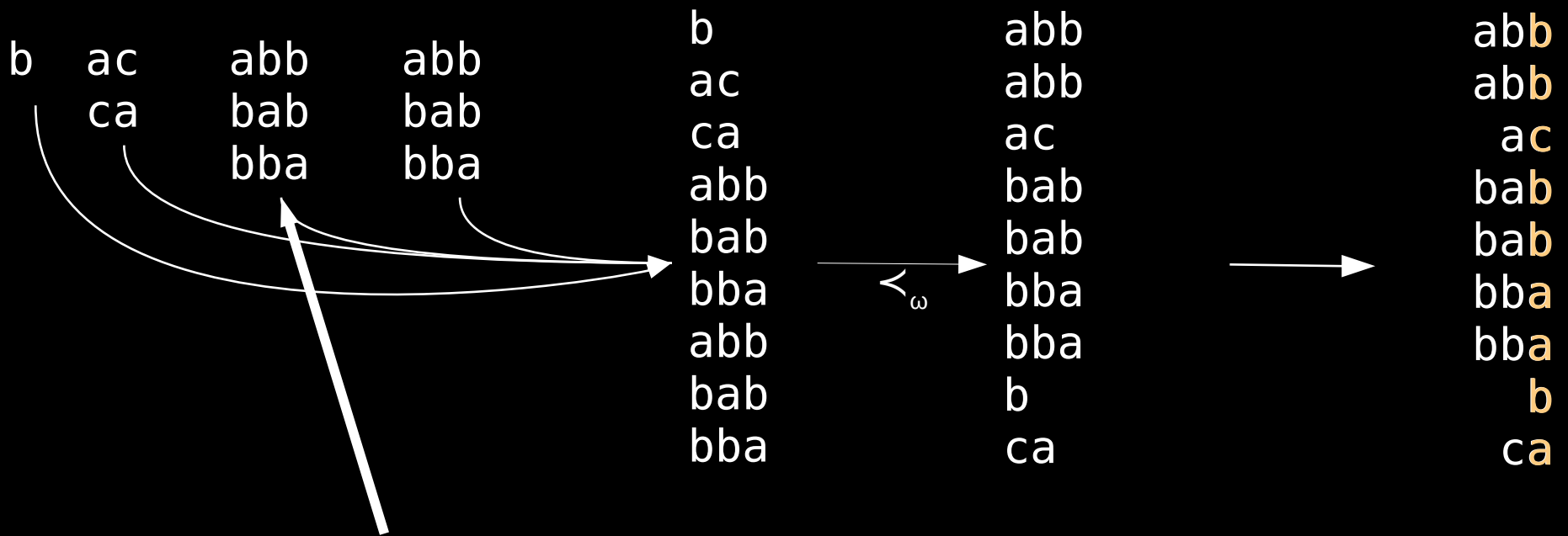
conjugates of all Lyndon factors

**BBWT(T) = bbcbbaaba**

# BBWT of bacabbabb

b | ac | abb | abb

*BBWT*



conjugates of all Lyndon factors

**BBWT(T) = bbcbbbaaba**

**BWT(T\$) = bbcbbb\$aaa**

# background

properties of BBWT :

- no \$ necessary
- BBWT seems to be more compressible than BWT for some inputs

[Scott and Gill '12]

- BBWT is indexible [Bannai+ '19]:
  - $O(m \lg m \lg \sigma)$  query time for  $m$  : pattern length, ( $\lg \sigma$  for the wavelet tree)
- is computable in
  - $O(n)$  time with  $n \lg n + n \lg \sigma$  bits [Bannai+ '21]
  - $O(n^2)$  time with  $O(\lg n)$  bits [Köppl+ '20]
  - $O(n \lg n / \lg \lg n)$  time with  $O(n \lg \sigma)$  bits [Bonomo' +14]



open problems

# number of runs $r_{\text{BBWT}}$

connection of  $r_{\text{BWT}}$  and  $r_{\text{BBWT}}$

(where  $r_S$  : number of character runs in string  $S$ )

- if  $T$  is Lyndon, then  $\text{BWT}(T) = \text{BBWT}(T)$

$\Rightarrow r_{\text{BWT}(T)} = r_{\text{BBWT}(T)} = 2$  for  $T$ : lower Christoffel words

[Mantaci+ '03]

- all conjugates of a text have the same BWT, but what about BBWT?
- empirical observation:  
# Lyndon factors is low  $\Rightarrow r_{\text{BWT}(T)} \approx r_{\text{BBWT}(T)}$

$$r_{\text{BBWT}(T[1]..T[n])} \langle \rangle r_{\text{BBWT}(T[n]..T[1])}$$

what is the relationship between the runs of  $\text{BBWT}(T)$  and  $\text{BBWT}(T_r)$ , where  $T_r$  is the inverted text

(for BWT: [Giuliani'+ 21])

# improve # Lyndon factors

- finding the alphabet ordering that maximizes/minimizes # Lyndon factors is NP-complete [Gibney+ '21]
- efficient approximation algorithm feasible?

use different orderings

- generalized lexicographic order, etc.
- but: then still index-able?

# Lyndon words  $< r_{\text{BBWT}}$  ?

is the number of distinct Lyndon words of  $T$   
bounded by  $r_{\text{BBWT}(T)}$  ?

if so, we gain:

$O(r_{\text{BBWT}(T)})$  words run-length compressed BBWT-index  
for  $r_{\text{BBWT}(T)} = o(n)$  [Bannai' +19]

# size of bijection cycles $k$

- since BBWT is a bijection, there exists a  $k$  such that

$$\text{BBWT}^k(T) = \text{BBWT}^{k-1}(\text{BBWT}(T)) = T \text{ with } k \geq 1$$

- we can compute  $k$  by constructing BBWT  $k$  times  $\rightarrow O(kn)$  time
- $O(n)$  time possible?

# BBWT construction algorithms

trade-off ?

- in  $O((n^2/\tau + n) \lg \tau)$  time with

-  $O(\tau \sigma_\tau)$  words of space? with

$\sigma_\tau: \max |\{ |\{T[i], \dots, T[i+\tau-1]\} | : i \in [1..n] \} |$

(result for BWT: [Crochemore+ '15])

• run-length encoded BBWT

-  $O(n \lg r)$  time with

-  $o(r)$  words of extra space

(result for BWT: [Bannai+ '20])