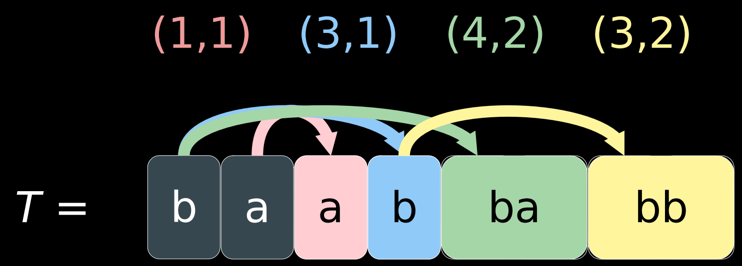
German for *wood*

HOLZ: High-Order Entropy Encoding of Lempel-Ziv Factor Distances

Dominik Köppl
Tokyo Medical and Dental University
Gonzalo Navarro
University of Chile
Nicola Prezza
Ca' Foscari University



Lempel-Ziv 77 (LZ)

- text factorization $T = $ | F_1 | F_2 | ... |
- used for lossless compression like in gzip, zip, 7zip, etc.
- LZ reads a text from left to right while
  - maintaining the read text in a dictionary and
  - replacing the remaining text with references into the dictionary

soundness

need always a suitable reference in the dictionary

$T =$ a b b a b b

1 2 3 4 5 6

soundness

need always a suitable reference in the dictionary

pre-handling:

- prepend all distinct characters to T

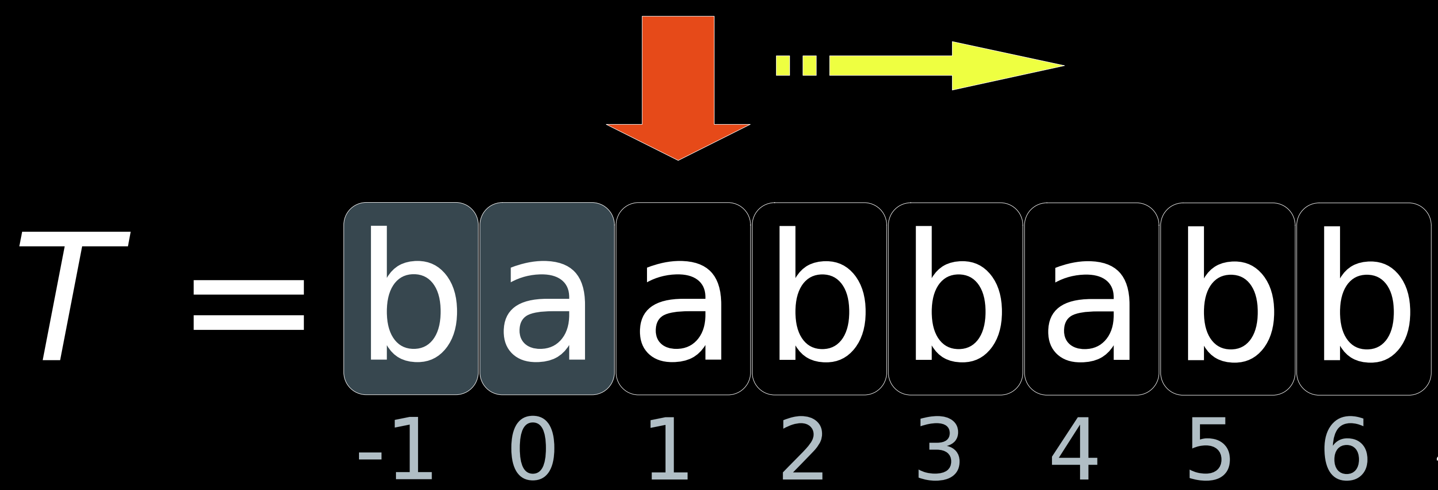
⇒ have a reference with length ≧ 1

$T =$	b	a	a	b	b	a	b	b
	-1	0	1	2	3	4	5	6

computing LZ

- take longest candidate as reference
- factorize T into $T = F_1 \cdots F_z$,

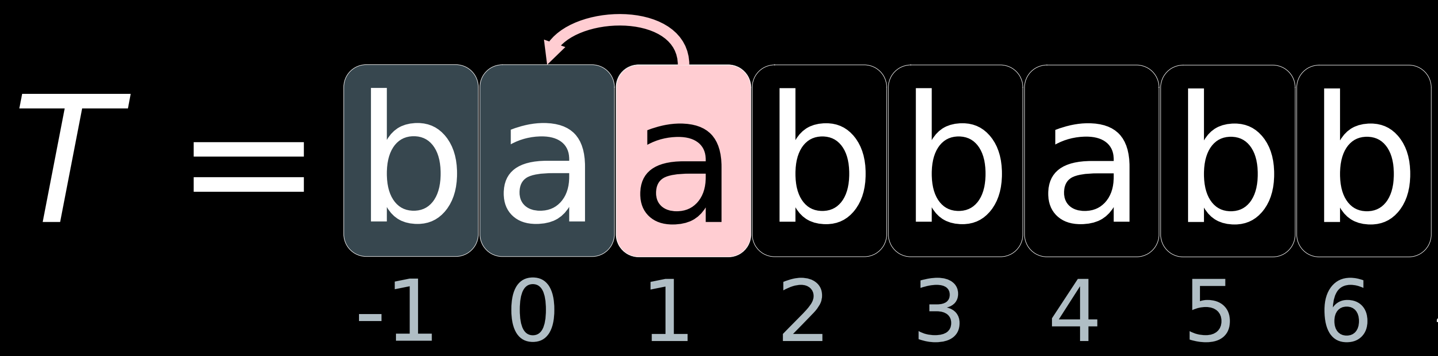
where F_x starting at a text position j refers to a suffix starting in T[-1..j-1] and having F_x as a prefix



computing LZ

- take longest candidate as reference
- factorize T into $T = F_1 \cdots F_z$,

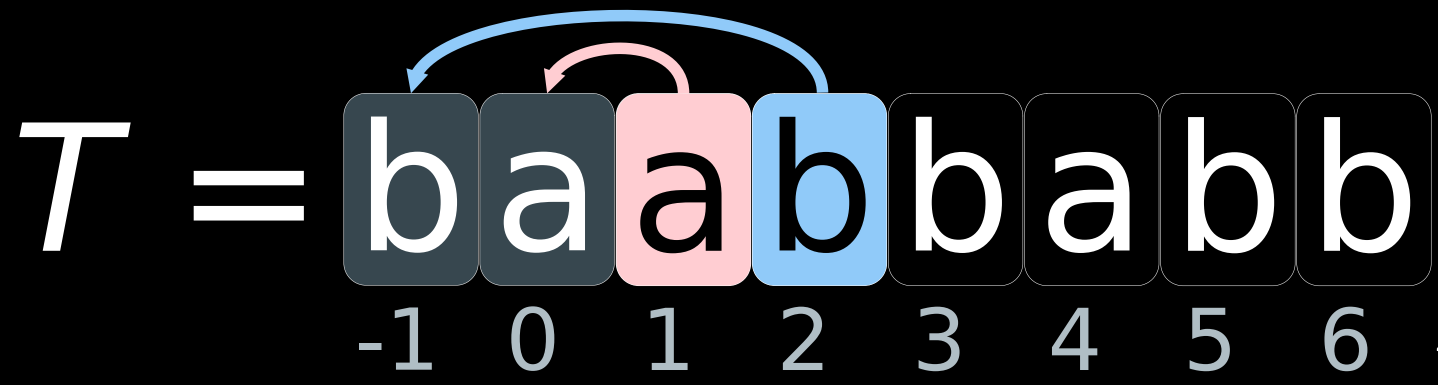
where F_x starting at a text position j refers to a suffix starting in T[-1..j-1] and having F_x as a prefix



computing LZ

- take longest candidate as reference
- factorize T into $T = F_1 \cdots F_z$,

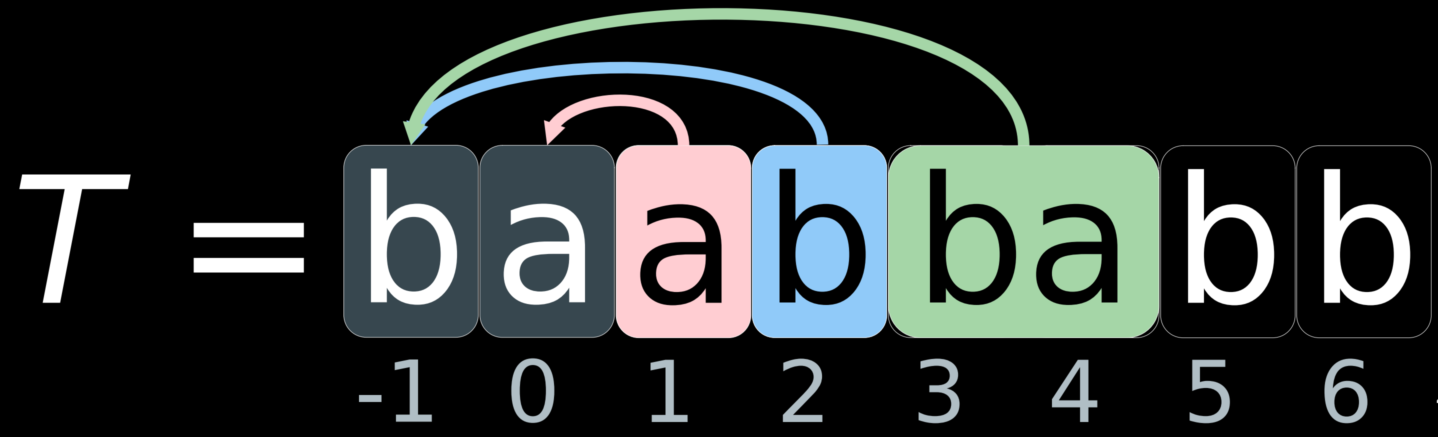
where F_x starting at a text position j refers to a suffix starting in T[-1..j-1] and having F_x as a prefix



computing LZ

- take longest candidate as reference
- factorize T into $T = F_1 \cdots F_z$,

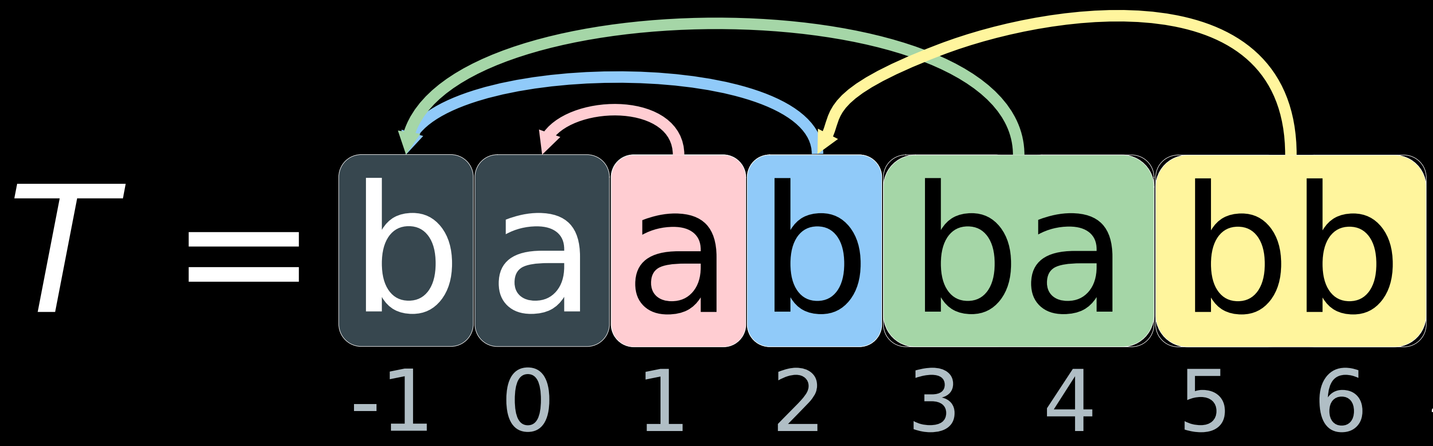
where F_x starting at a text position j refers to a suffix starting in T[-1..j-1] and having F_x as a prefix



computing LZ

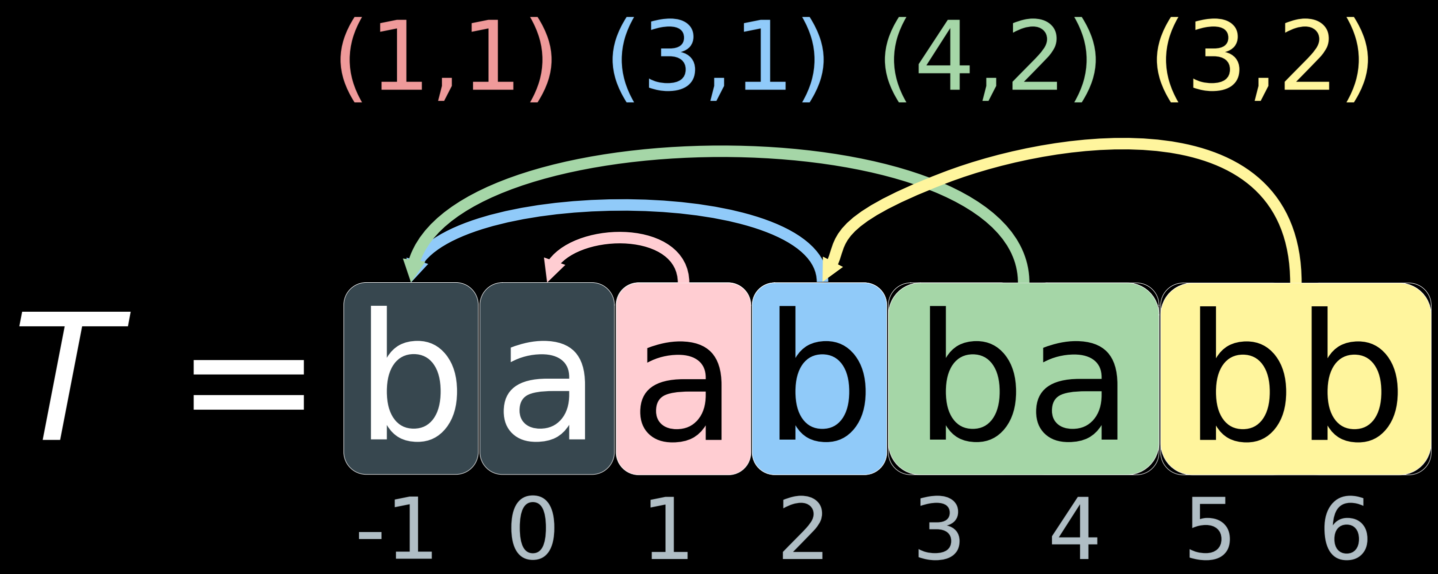
- take longest candidate as reference
- factorize T into $T = F_1 \cdots F_z$,

where F_x starting at a text position j refers to a suffix starting in T[-1..j-1] and having F_x as a prefix



pair encoding

- represent each factor as a pair of distance and length
- to obtain compression, we encode the pairs with an universal coder like Elias γ code



decompression

since a reference points always to the already read part, we can decompress the text

$(1,1)$ $(3,1)$ $(4,2)$ $(3,2)$

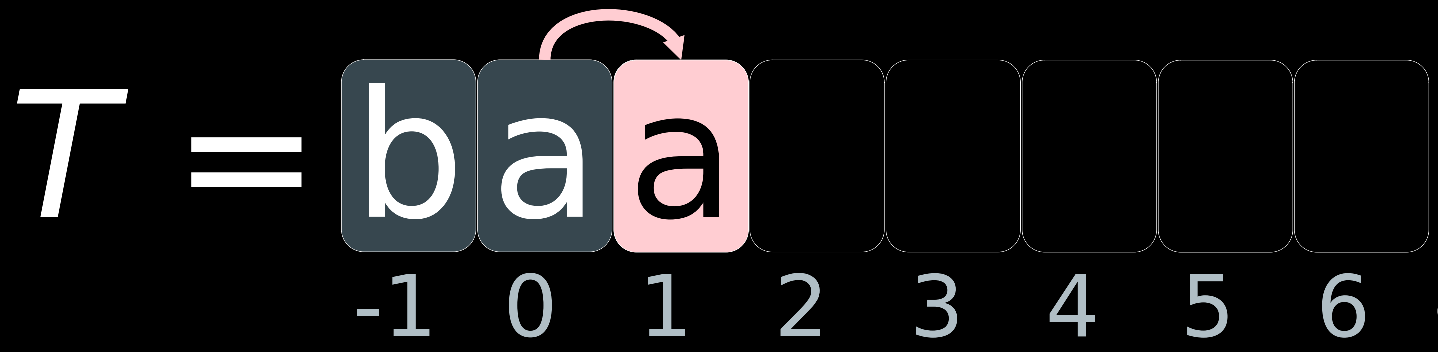
$T =$

b	a						
-1	0	1	2	3	4	5	6

decompression

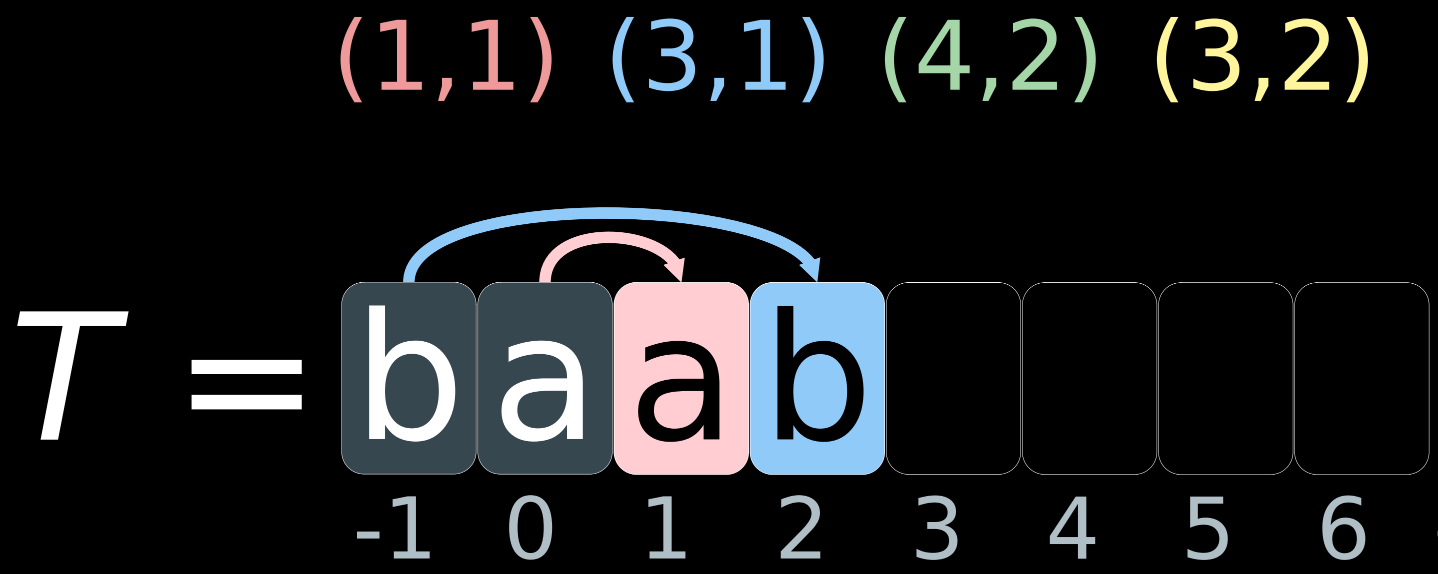
since a reference points always to the already read part, we can decompress the text

(1,1) (3,1) (4,2) (3,2)



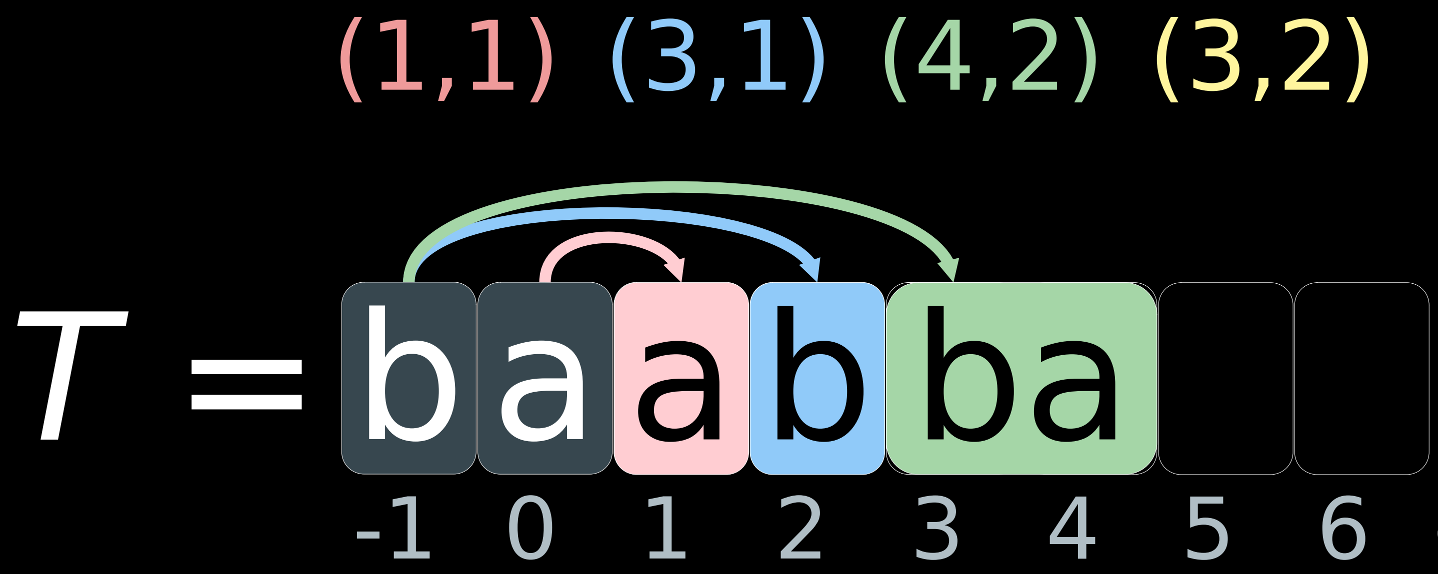
decompression

since a reference points always to the already read part, we can decompress the text



decompression

since a reference points always to the already read part, we can decompress the text



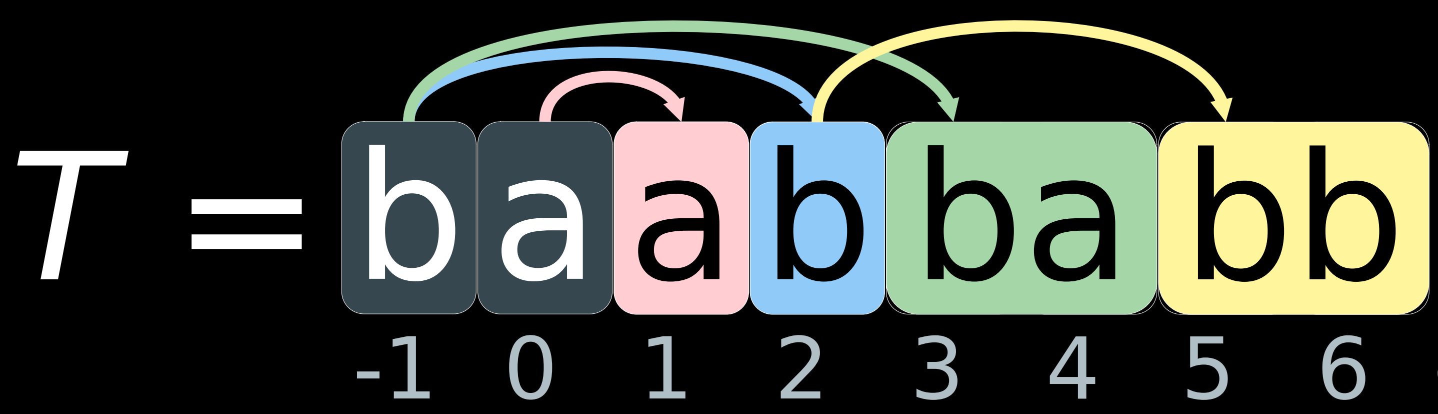
decompression

- since a reference points always to the already read part, we can decompress the text

however in practice:

- the distances do not compress well!

(1,1) (3,1) (4,2) (3,2)



representing distances

new representation:

- pre-processing: compute lengths and starting positions of all factors
- compute the distance based on a list maintaining all prefixes of the read text
- this list is sorted colex(icographically)
- we call the resulting distance holz offset (high order Lempel-Ziv)

colex(icographic) order

= sort according to the lexicographic order of the reversed strings

Example

lexicographic order	colex. order
aaab	abaa
aaba	aaba
abaa	bbba
bbba	aaab

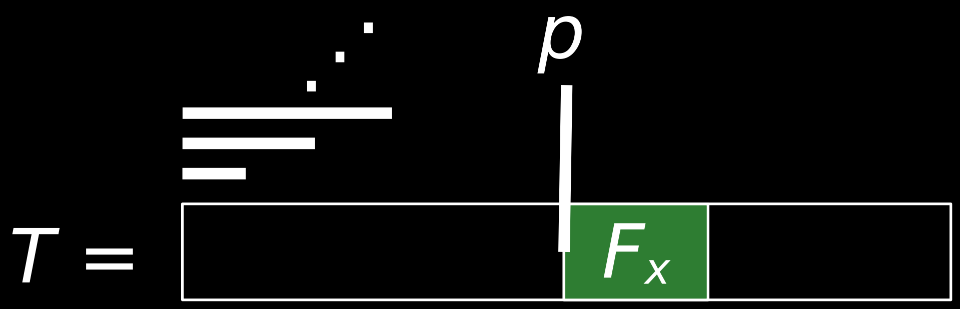
notations

- $T[i..j]$: substring; like $T[0..2]$ = aab
- $T[i..]$: suffix; like $T[3..]$ = babb
- $T[-1..j]$: prefix
- ε : empty string (length 0)
- assume binary alphabet (extension to general ordered alphabets is easy)

T =	b	a	a	b	b	a	b	b
	-1	0	1	2	3	4	5	6

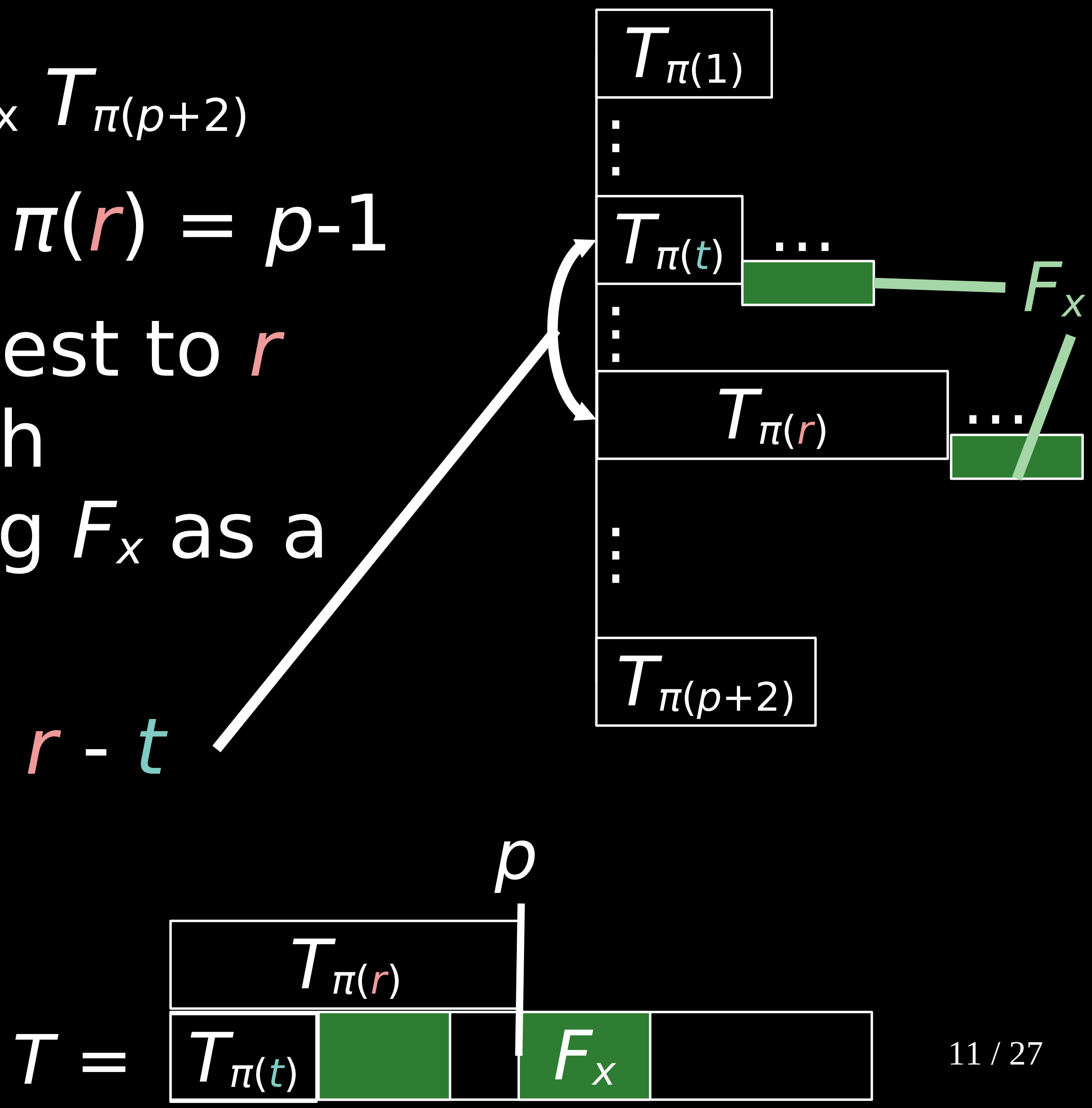
holz: overview

- let $T_p := T[-1..p]$
- $T_{-2} = \varepsilon$, $T_{-1} = \text{b}$, $T_0 = \text{ba}$, $T_1 = \text{baa}$ ...
- suppose we want to compute factor F_x starting at $T[p..]$
- arrange $T_{-2}, \ldots, T_{p-1}$ in colex. order to get $T_{\pi(1)} \prec_{\text{colex}} \ldots \prec_{\text{colex}} T_{\pi(p+2)}$ with π ranking the prefix in colex. order



computing offsets

- $T_{\pi(1)} \prec_{\text{colex}} \ldots \prec_{\text{colex}} T_{\pi(p+2)}$
- let r be given by $\pi(r) = p$-1
- let t be rank closest to r among those with $T[\pi(t)+1..]$ having F_x as a prefix
- F_x's holz offset is r - t



offsets : example

precomputation: sort

- $T_{-2} = \varepsilon$
- $T_{-1} = \text{b}$
- $T_0 = \text{ba}$

in colex. order

$p = 1$

$T =$ b a a b ba bb

-1 0 1 2 3 4 5 6

offsets : example

precomputation: sort

- $T_{-2} = \varepsilon$
- $T_{-1} = b$
- $T_0 = ba$

in colex. order

1	$T_{-2} =$
2	$T_0 =$ ba
3	$T_{-1} =$ b

$p = 1$

$T =$ b a a b ba bb

-1 0 1 2 3 4 5 6

offsets : example

precomputation: sort

- $T_{-2} = \varepsilon$
- $T_{-1} = \text{b}$
- $T_{0} = \text{ba}$

in colex. order

remaining suffix →

1	T_{-2} =	|baabbabb
2	T_{0} =	ba|abbabb
3	T_{-1} =	b|aabbabb

$p = 1$

T = b a a b ba bb

-1 0 1 2 3 4 5 6

offsets : example

- $F_1 = T[1]$ is first factor
- starting position of F_1 is p=1
- F_p=F_1 starts after $T_{p\text{-}1}$=T_0
- rank of $T_{p\text{-}1} = T_0$ is $r = 2$
- rank of T_{-1} is t =3

remaining suffix →

1	T_{-2} =	|baabbabb
(2)	T_0 =	ba|abbabb
(3)	T_{-1} =	b|aabbabb

$r - t = 2 - 3 = -1$

↓ $p = 1$

T = b a a b ba bb

-1 0 1 2 3 4 5 6

offsets : example

- add T_1
- $p = 2$
- $r = 2$
- $t = 1$
- $r - t = 1$

remaining suffix →

①	T_{-2} =	| baabbabb
②	T_1 = baa	| bbabb
3	T_0 = ba	| abbabb
4	T_{-1} = b	| aabbabb

T = b a a b ba bb

-1 0 1 2 3 4 5 6

offsets : example

- add T_2
- $p = 3$
- $r = 5$
- $t = 1$
- $r - t = 4$

remaining suffix →

1	T_{-2} =		| baabbabb
2	T_1 =	baa	| bbabb
3	T_0 =	ba	| abbabb
4	T_{-1} =	b	| aabbabb
5	T_2 =	baab	| babb

T = b a a b ba bb

-1 0 1 2 3 4 5 6

offsets : example

- add T_3 and T_4
- $p = 5$
- $r = 4$
- $t = 2$
- $r - t = 2$

1	T_{-2} =		|	baabbabb
2	T_1 =	baa	|	bbabb
3	T_0 =	ba	|	abbabb
4	T_4 =	baabba	|	bb
5	T_{-1} =	b	|	aabbabb
6	T_2 =	baab	|	babb
7	T_3 =	baabb	|	abb

T = b a a b ba bb

-1	0	1	2	3	4	5	6
b	a	a	b	b	a	b	b

experiments

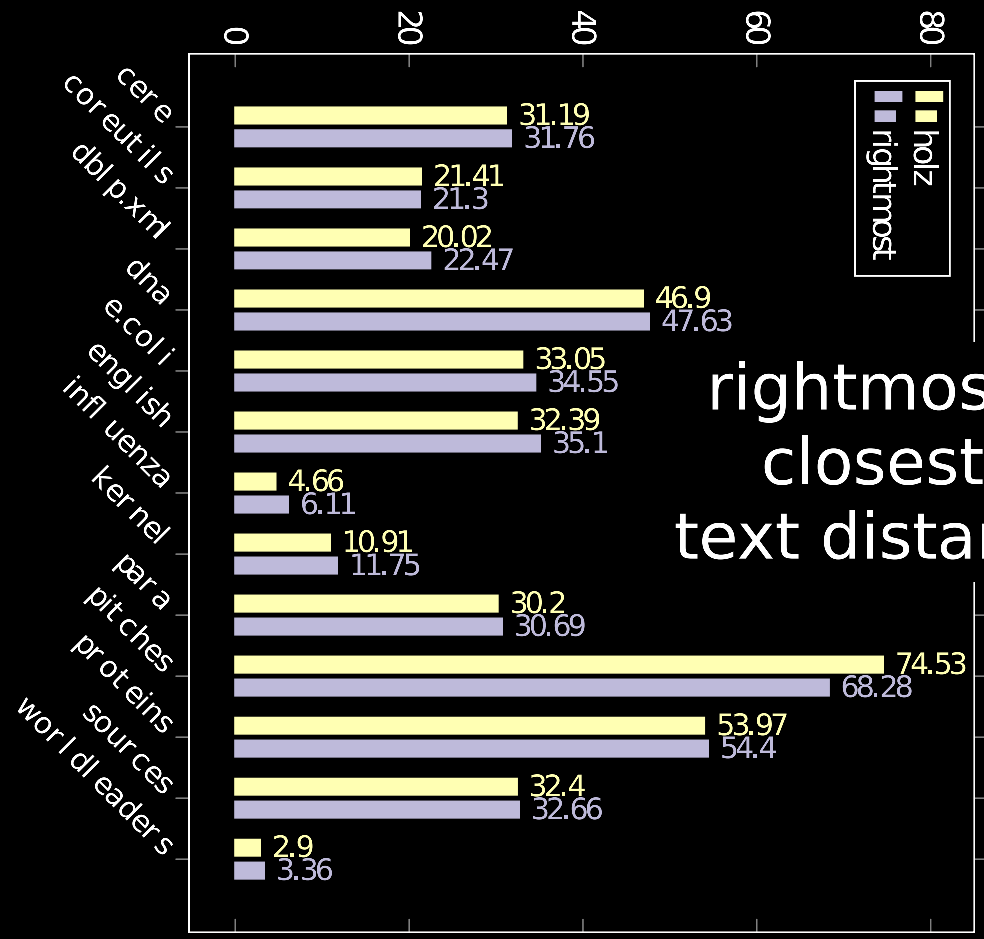
- datasets from Pizza & Chili corpus
- take 20 MB prefix of each dataset,
- compute LZ factorization,
- encode pairs with Elias γ code,
- compare compression ratios

experiments

dataset	σ	z [K]	H_0	H_2	H_4
cere	5	8492	2.20	1.79	1.78
coreutils	235	3010	5.45	2.84	1.31
dblp.xml	96	3042	5.22	1.94	0.89
dna	14	12706	1.98	1.92	1.91
e.coli	11	8834	1.99	1.96	1.94
english	143	5478	4.53	2.89	1.94
inf uenza	15	876	1.97	1.93	1.91
kernel	160	1667	5.38	2.87	1.47
para	5	8254	2.17	1.83	1.82
pitches	129	10407	5.62	4.28	2.18
proteins	25	8499	4.20	4.07	2.97
sources	111	4878	5.52	2.98	1.60
worldleaders	89	408	4.09	1.74	0.73

- z : #factors
- [K] : 10^3 (kilo)
- σ: alphabet size
- H_k : k-th order empirical entropy

compression ratio
(lower = better)



rightmost:
closest
text distance

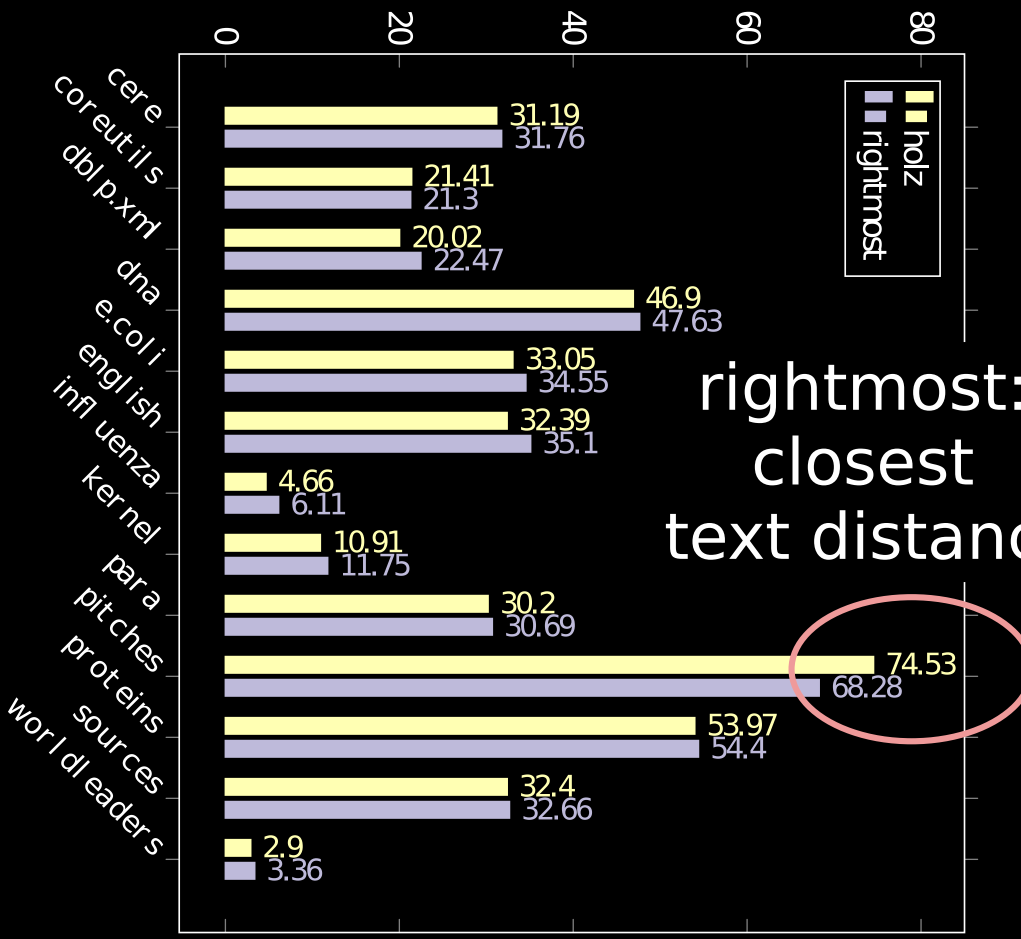
(Elias γ encoded)

experiments

dataset	σ	z [K]	H_0	H_2	H_4
cere	5	8492	2.20	1.79	1.78
coreutils	235	3010	5.45	2.84	1.31
dblp.xml	96	3042	5.22	1.94	0.89
dna	14	12706	1.98	1.92	1.91
e.coli	11	8834	1.99	1.96	1.94
english	143	5478	4.53	2.89	1.94
inf uenza	15	876	1.97	1.93	1.91
kernel	160	1667	5.38	2.87	1.47
para	5	8254	2.17	1.83	1.82
pitches	129	10407	5.62	4.28	2.18
proteins	25	8499	4.20	4.07	2.97
sources	111	4878	5.52	2.98	1.60
worldleaders	89	408	4.09	1.74	0.73

- z : #factors
- [K] : 10^3 (kilo)
- σ: alphabet size
- H_k : k-th order empirical entropy

compression ratio (lower = better)



rightmost: closest text distance

(Elias γ encoded)

holz is only worse when H_k is high!

about compression ratio

why are the holz offsets smaller than the distances most of the time?

answer sketch :

- contexts before the references are similar to the contexts before the factors ⇒ offsets are small
- similar observation for the Burrows-Wheeler transform (BWT) obtaining compression close to k-th order entropy via so-called *compression boosting* [Ferragina,Manzini ‘04]

algorithmic aspects

problem:

how to maintain the colex. order of the prefixes?

idea : use dynamic BWT

- index processed text in reverse order (BWT maintains suffixes in lex. order)

⇒ reversed BWT maintains prefixes in colex. order

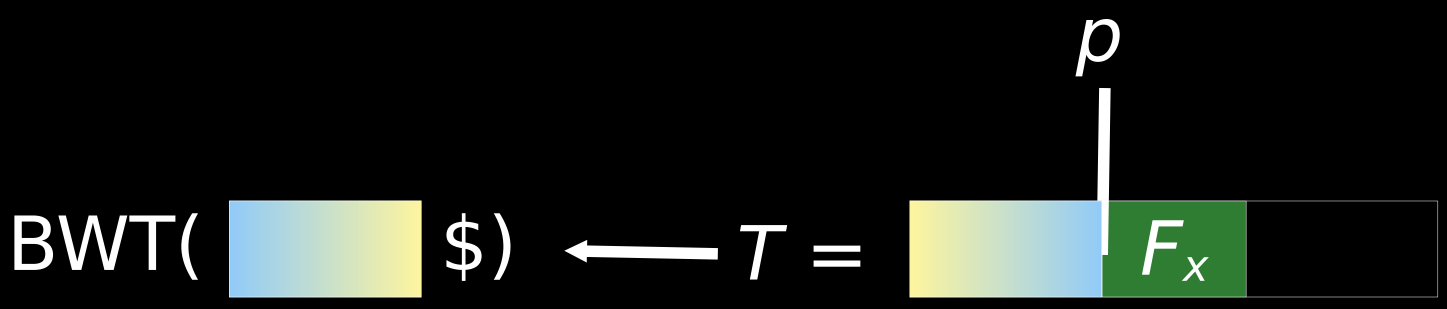
- $n\ \mathrm{H}_k + o(n \lg \sigma)$ space
- $O(n \lg n / \lg \lg n)$ time

H_k : k-th order empirical entropy

[Policriti, Prezza ’18] + [Munro, Nekrich ‘15]

offsets via BWT

- $T[-1..n]$ = baabbabb
- $T^R\$$ = bbabbaab\$ (reverse T and append artificial character \$)
- pre-compute BWT(ab\$)
- invariant:
  have BWT($T^R[n\text{-}p+1..n+2]\$$) computed
  when computing factor F_x starting at $T[p..]$



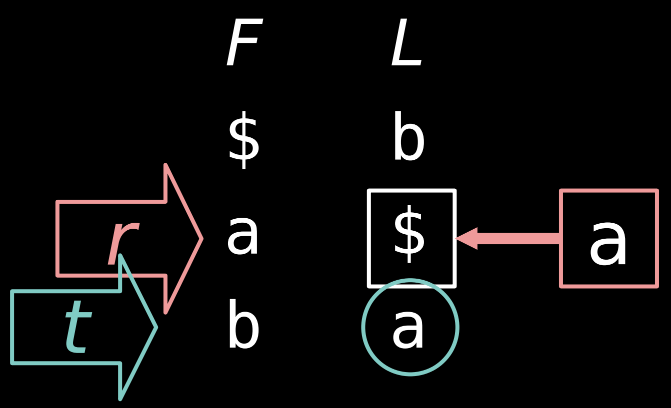
offset of F_1

BWT(ab$)

F	L
$	b
a	$
b	a

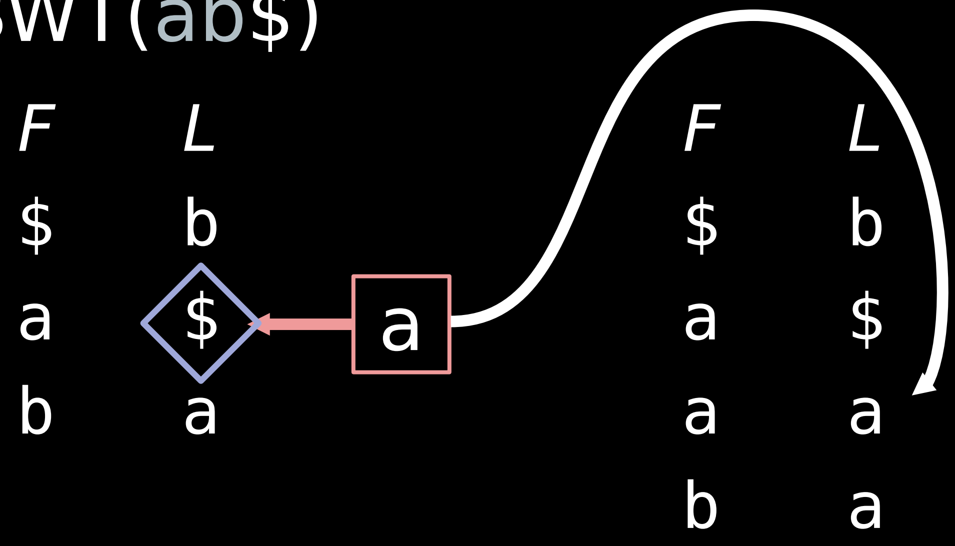
r : place of $

t : reference



BWT: prepend a character

BWT(ab$) BWT(aab$)



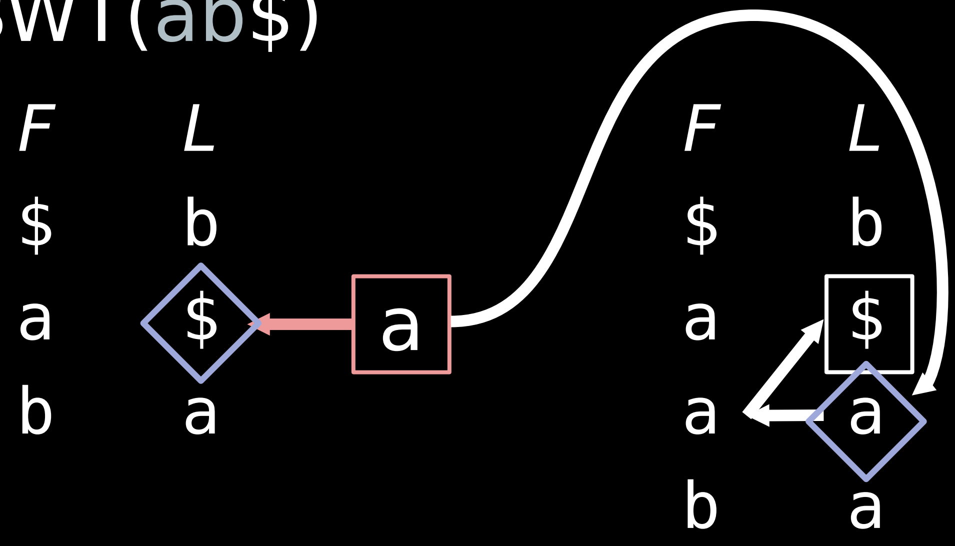
[Crochemore+ ‘15]:

Given a character *a* we want to prepend

1) replace $L[i]$ = $ with *a*
2) if $L[i]$ is now the *j-th* *a* in $L[1..i]$,
   insert $ at $L[k]$, where $F[k]$ is the *j*-th *a* of *F*

BWT: prepend a character

BWT(ab$) BWT(aab$)

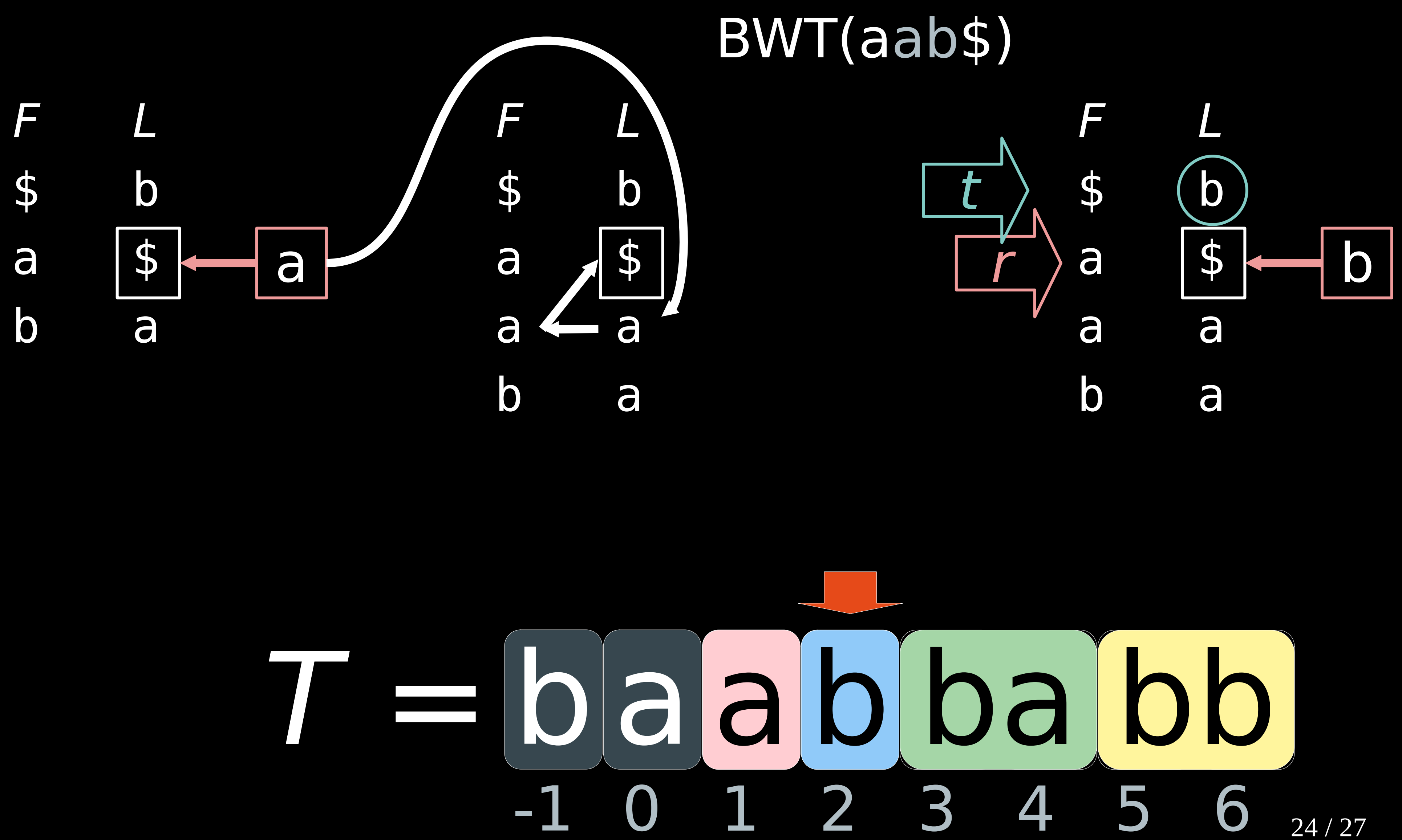


[Crochemore+ '15]:

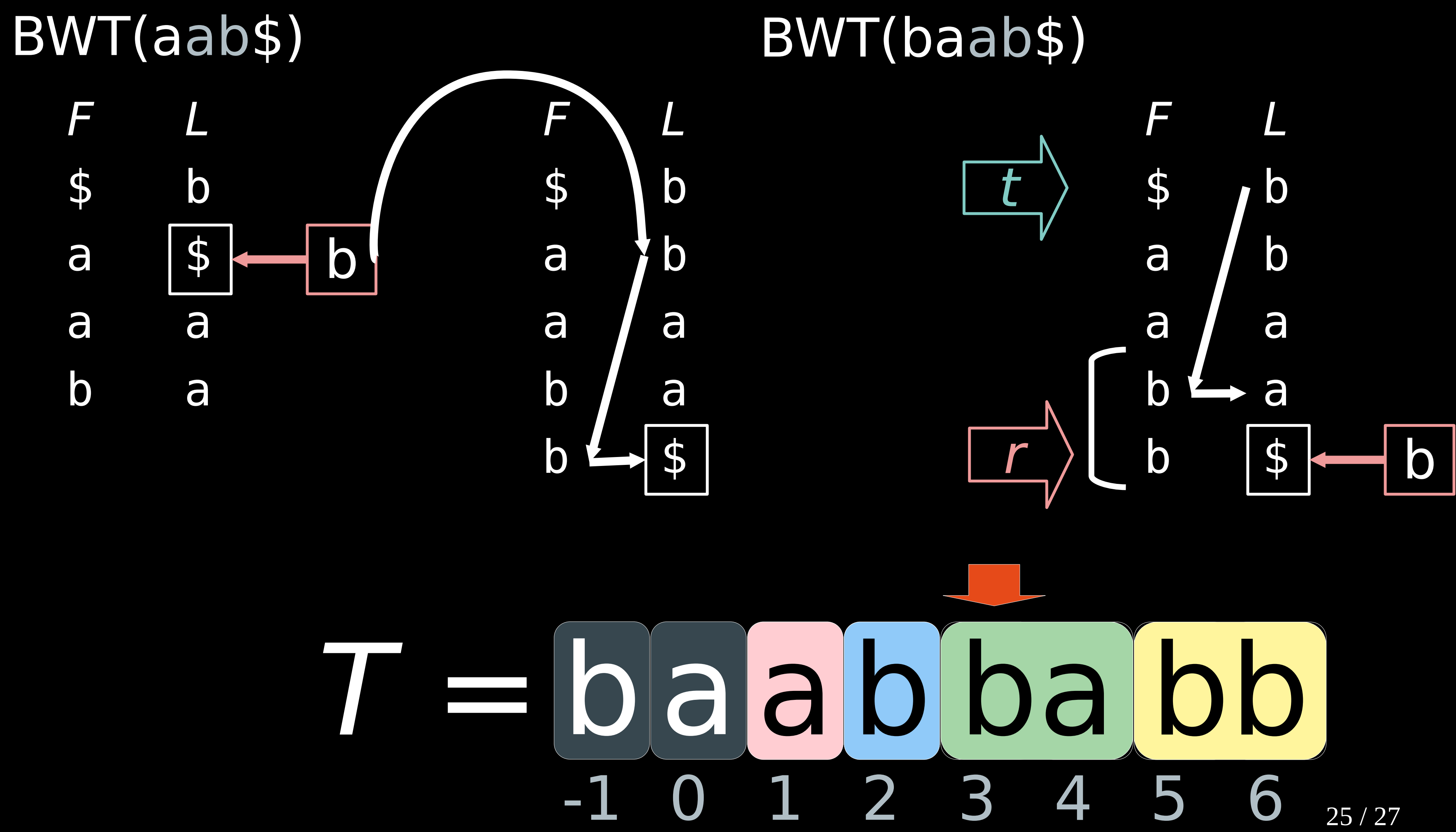
Given a character *a* we want to prepend

1) replace $L[i]$ = $ with *a*
2) if $L[i]$ is now the *j-th* *a* in $L[1..i]$,
   insert $ at $L[k]$, where $F[k]$ is the *j*-th *a* of *F*

offset of F_2

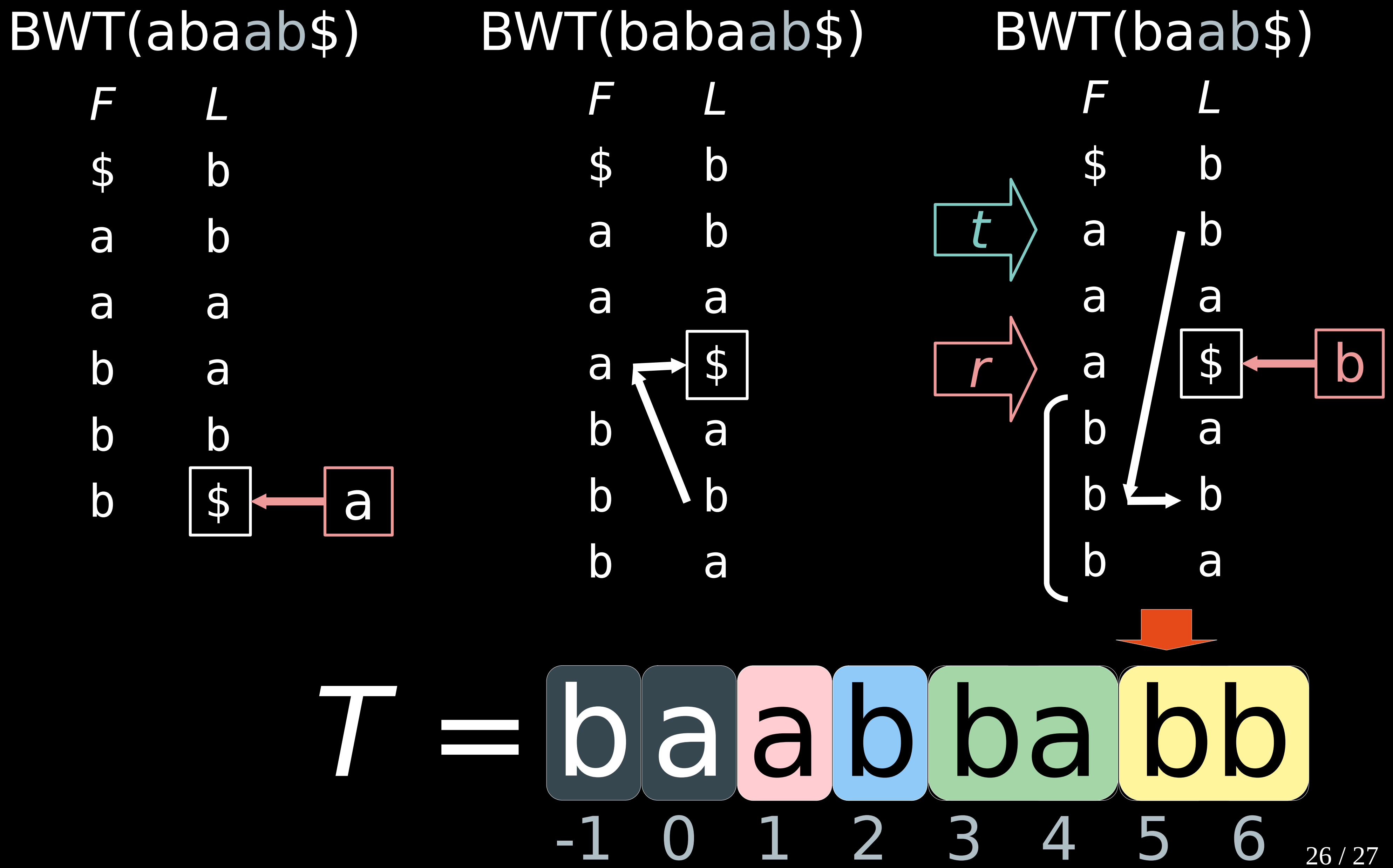


offset of F_3



BWT(aab$)
F L
$ b
a $
a a
b a
b
F L
$ b
a b
a a
b a
b $
BWT(baab$)
t
r
F L
$ b
a b
a a
b a
b $
b
T = b a a b b a b b
-1 0 1 2 3 4 5 6

offset of F_4


BWT(abaab$)
F L
$ b
a b
a a
b a
b b
b $
a
BWT(babaab$)
F L
$ b
a b
a a
a $
b a
b b
b a
BWT(baab$)
t
r
F L
$ b
a b
a a
a $
b a
b b
b a
b
T = baabbabb
-1 0 1 2 3 4 5 6

summary

- LZ compressors usually represent factors by pairs of lengths and distances
- distances compress badly
- exchange distances with holz offsets:
  = distance within the list of prefixes of the read text maintained in colex. order
- for low-entropy texts, holz offsets provide empirically better compression ratios

future work

- dynamic BWT is practical bottleneck wrt. time