Indexing the Bijective BWT

Hideo Bannai (Kyushu University),

Juha Kärkkäinen (Helsinki Institute of Information Technology),

Dominik Köppl (TU Dortmund),

Marcin Piątkowski (Nicolaus Copernicus University)

This presentation received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 690941.

FM Index

ingredients

- BWT
- wavelet tree

FM Index

ingredients

- BWT
- wavelet tree

operation: backward search

- locate pattern
- #occ: number of occurrences
- time independent on #occ

FM Index on bijective BWT

ingredients

- bijective BWT
- wavelet tree

operation: backward search

- locate pattern
- #occ: number of occurrences
- time independent on #occ

FM Index

on bijective BWT

ingredients

- bijective BWT
- wavelet tree

operation: backward search

- locate pattern
- #occ: number of occurrences
- time independent on #occ

reason:
bijective BWT is cool

Lyndon words

- a
- aabab

Lyndon word is smaller than
- any proper suffix
- any rotation

Lyndon words

- a
- aabab

Lyndon word is smaller than
- any proper suffix
- any rotation

not Lyndon words:

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

Lyndon factorization

- input: text T
- output: factorization $T_1 ... T_t$ with
  - T_i is Lyndon word
  - $T_i \geq_{\text{lex}} T_{i+1}$
  - factorization uniquely defined
  - linear time [Duval'88]

(Chen-Fox-Lyndon theorem)

properties [Duval' 88]

- T_t:
  - smallest Lyndon word
  - smallest suffix of T
- T_i primitive
- T_1 longest Lyndon prefix of T[1..]
- T_{i+1} longest Lyndon prefix of $T[|T_1 \cdots T_i|+1..]$

$\prec_\omega$

- $u \prec_\omega w :\Longleftrightarrow uuuu\ldots <_{\text{lex}} wwww\ldots$
- ab < aba but aba $\prec_\omega$ ab

abababab⋯
ababaaba⋯

bijective BWT of senescence

s|enes|cen|ce

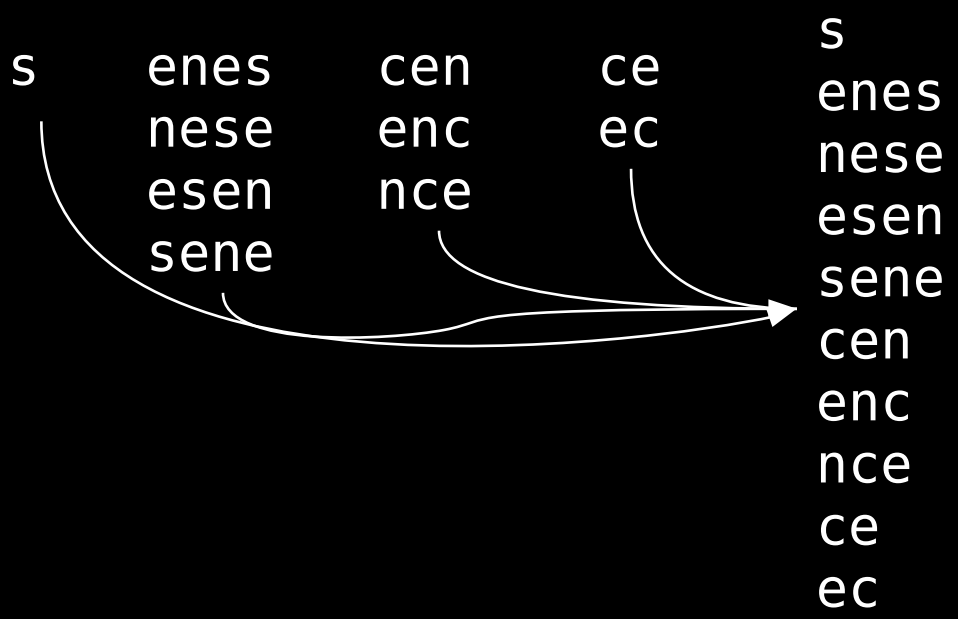
bijective BWT of senescence

s|enes|cen|ce

s	enes	cen	ce
	nese	enc	ec
	esen	nce	
	sene		

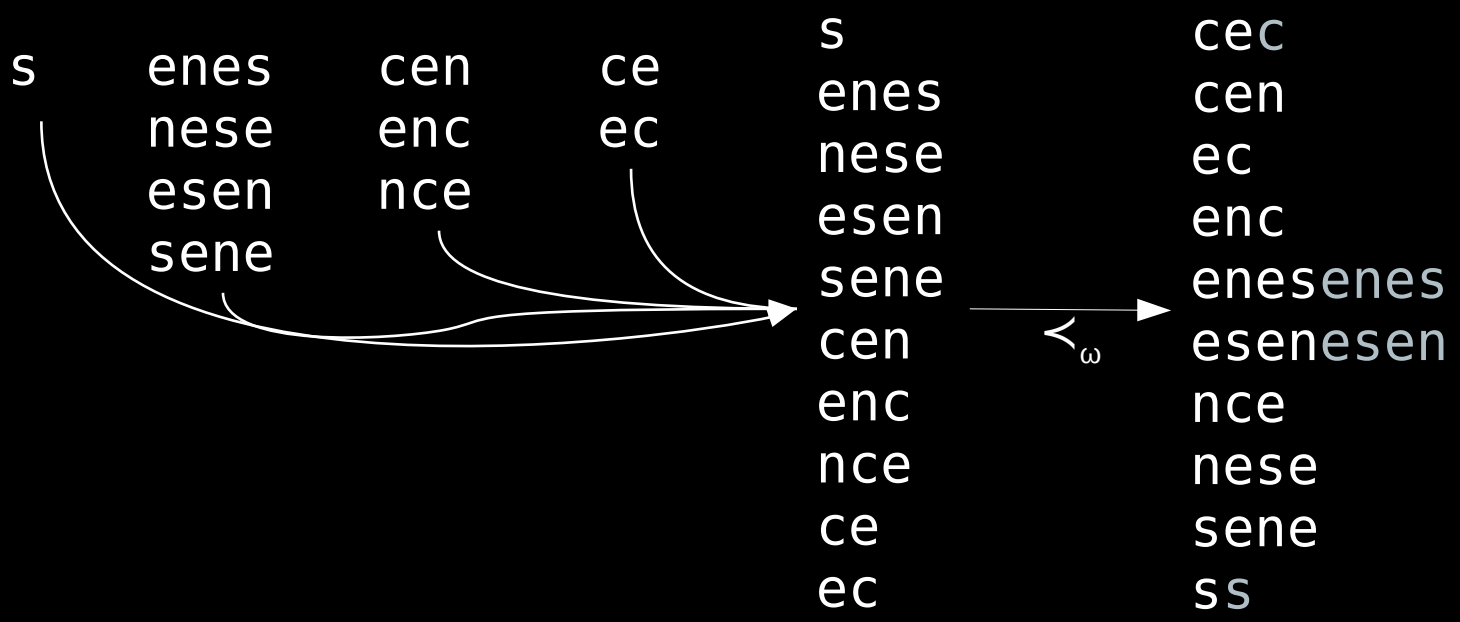
bijective BWT of senescence

s|enes|cen|ce



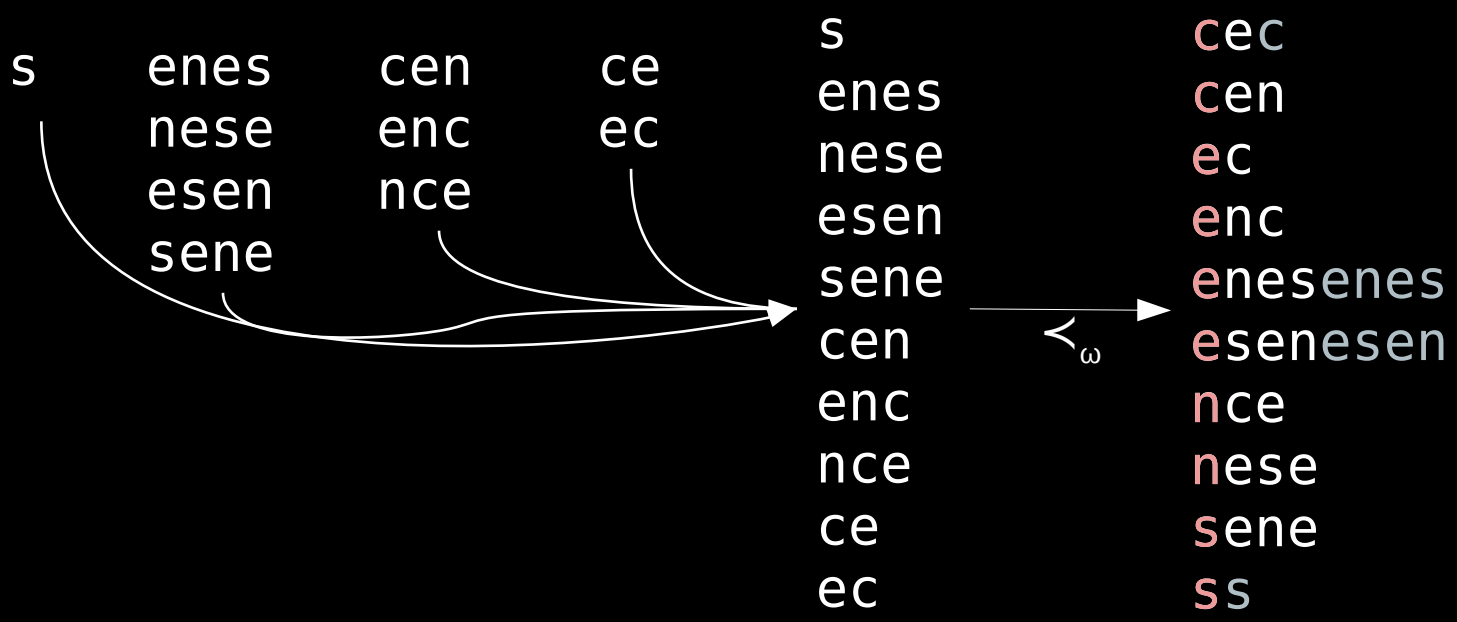
bijective BWT of senescence

s|enes|cen|ce



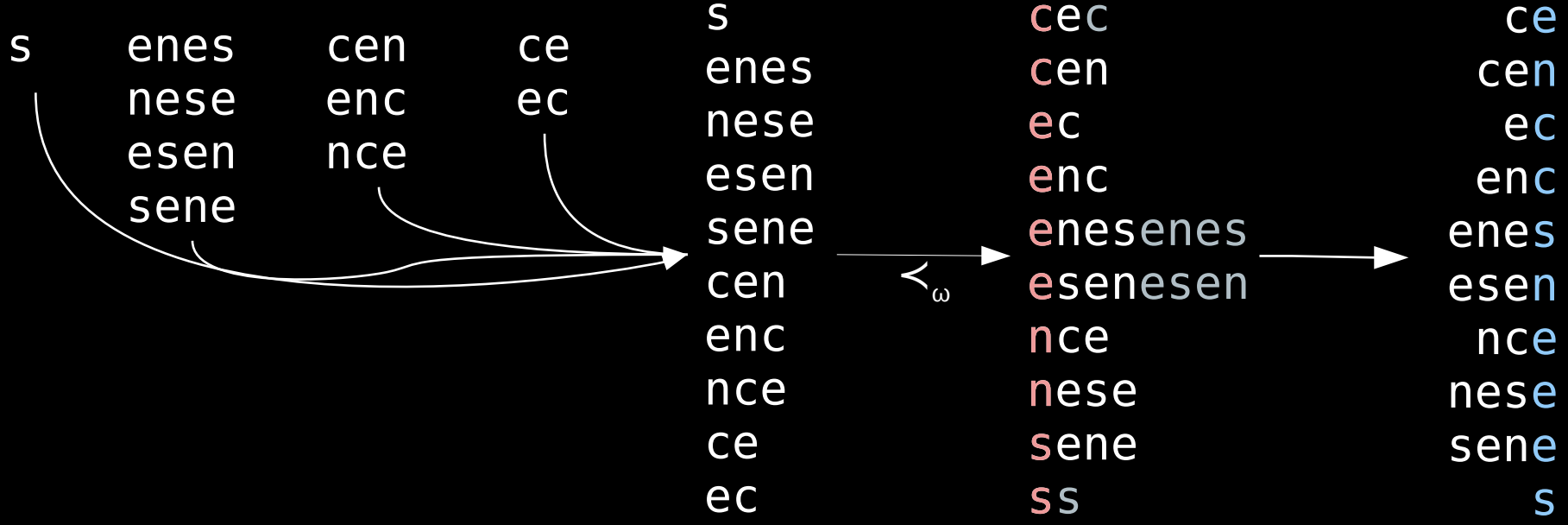
bijective BWT of senescence

s|enes|cen|ce



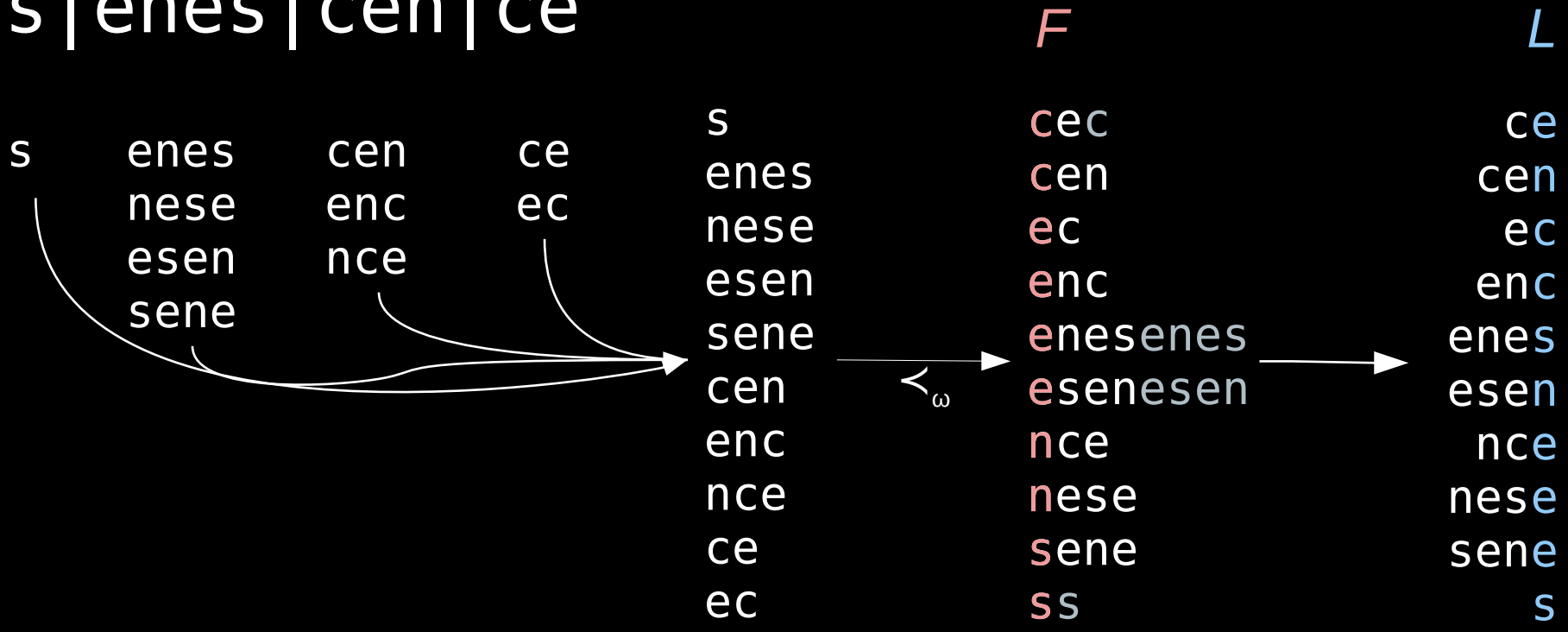
bijective BWT of senescence

s|enes|cen|ce



bijective BWT of senescence

s|enes|cen|ce

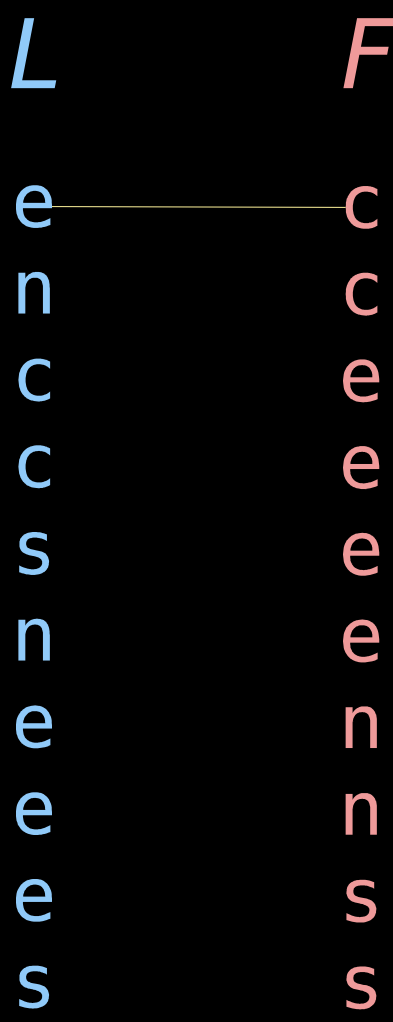


result: enccsneees

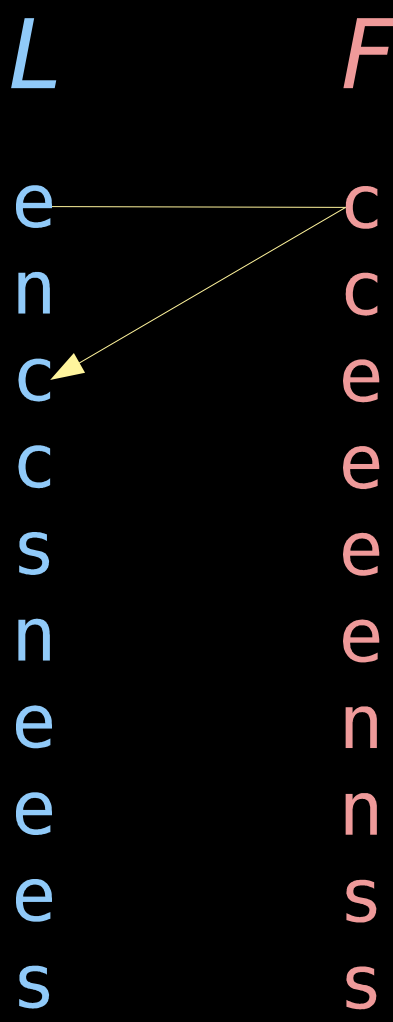
cycles

L	*F*
e	c
n	c
c	e
c	e
s	e
n	e
e	n
e	n
e	s
s	s

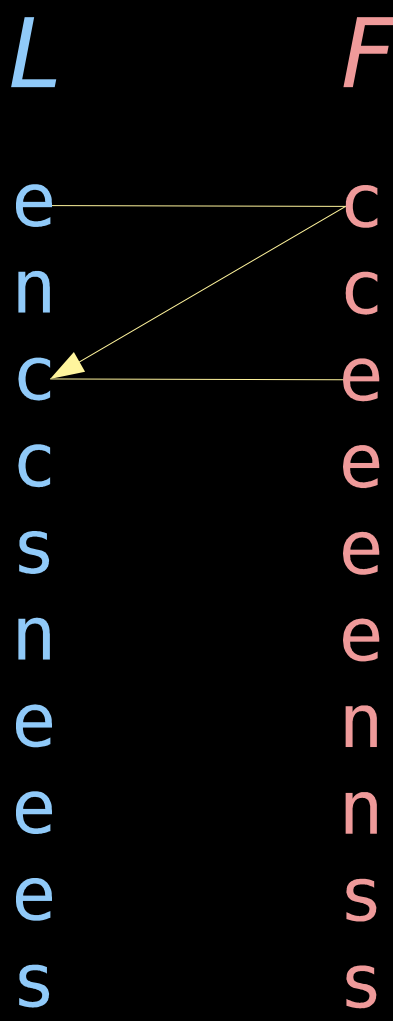
cycles



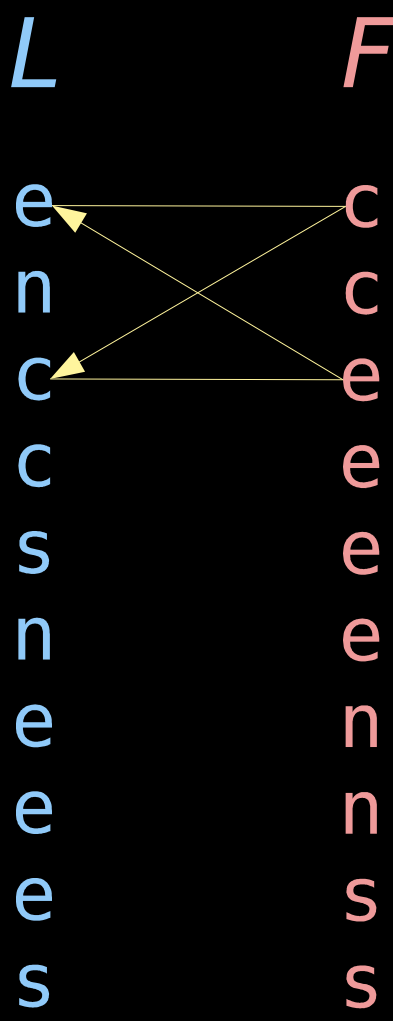
cycles



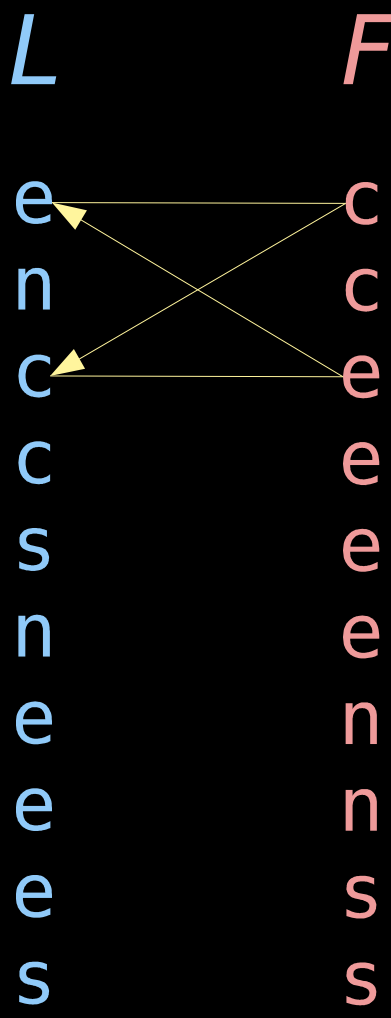
cycles



cycles

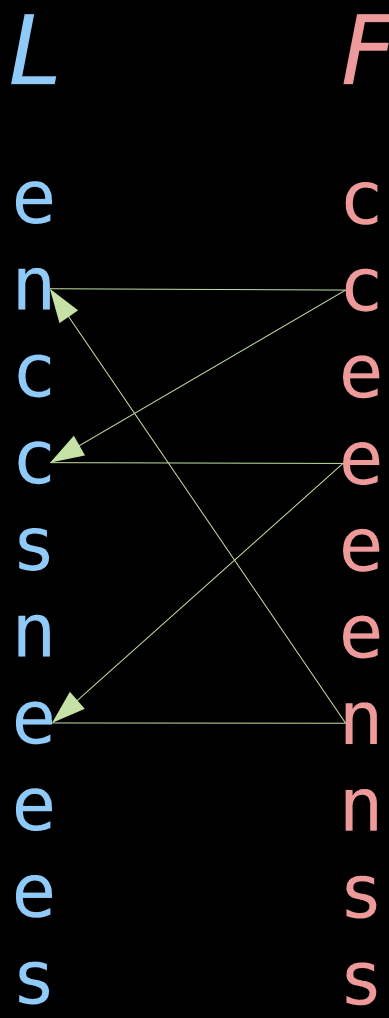


cycles



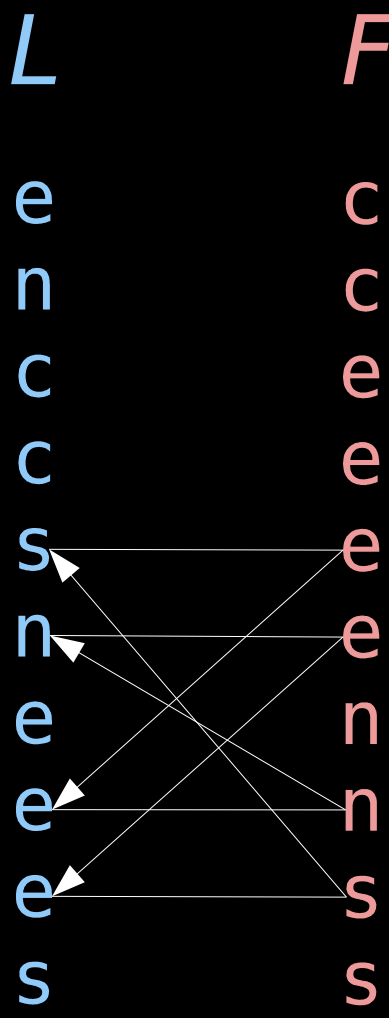
ce

cycles



ce

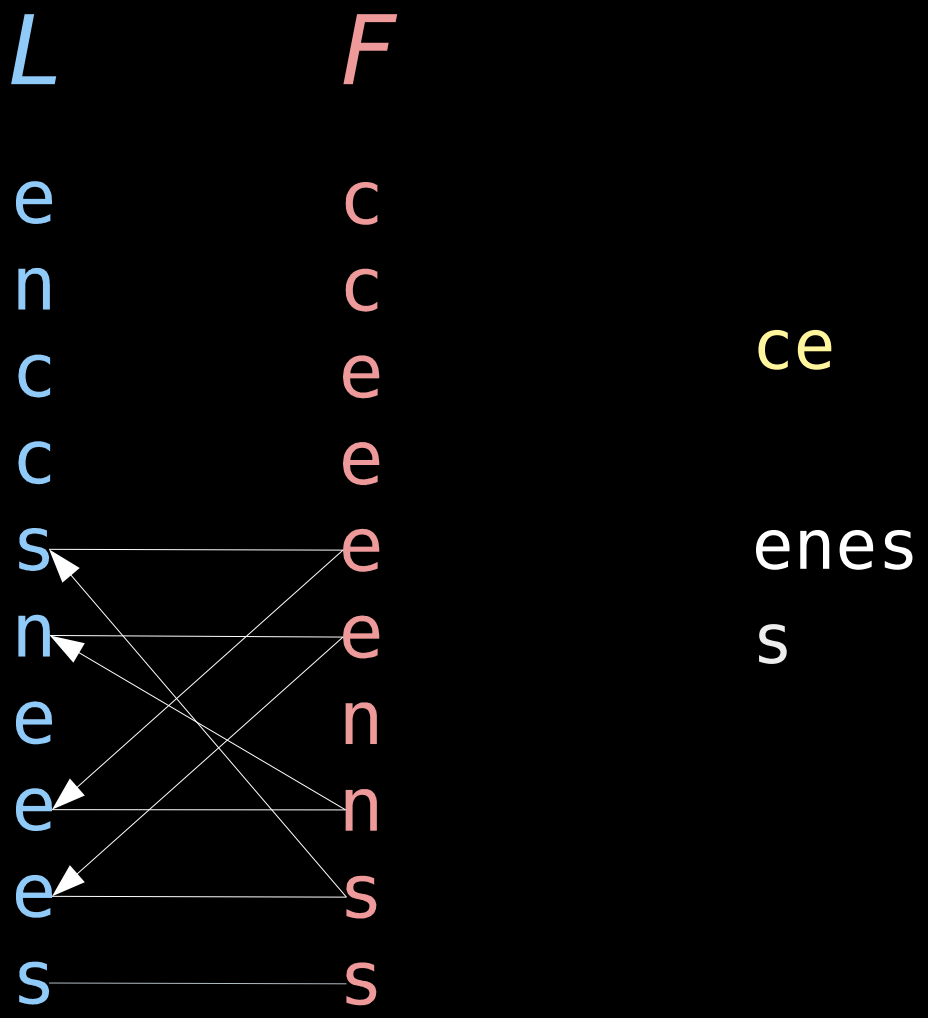
cycles



ce

enes

cycles

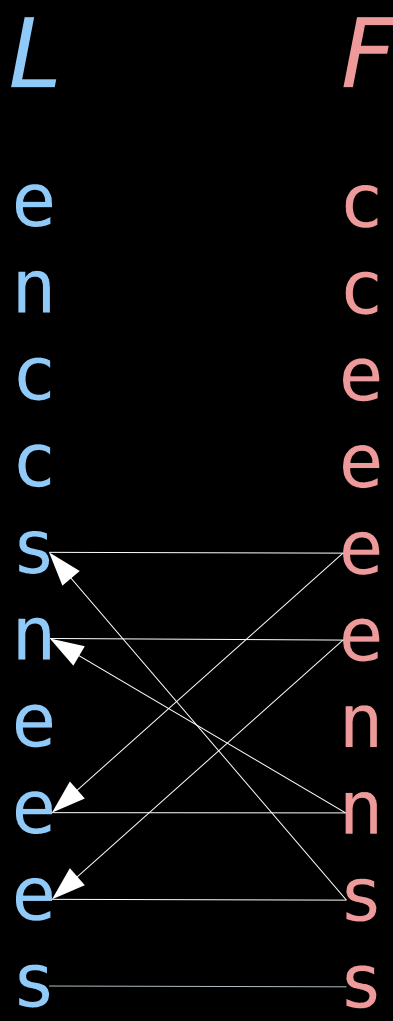


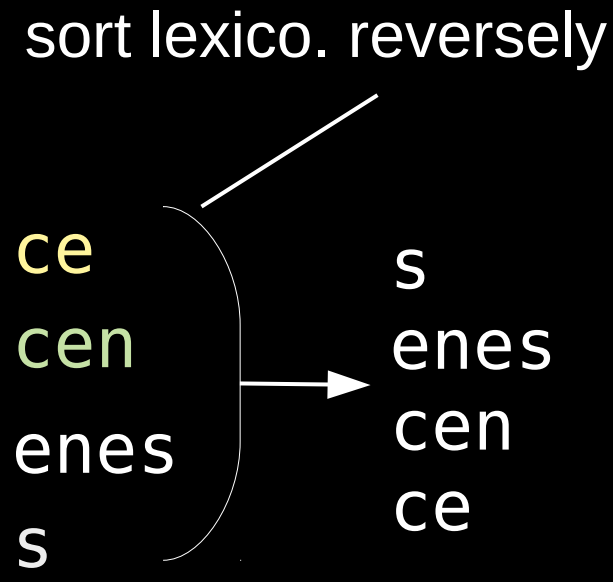
ce

enes

s

cycles


L
F
e
n
c
c
s
n
e
e
e
s
c
c
e
e
e
e
n
n
s
s


sort lexico. reversely
ce
cen
enes
s
s
enes
cen
ce

backward search 'cen'

s|enes|cen|ce

F	*L*
c	ce
c	cen
e	ec
e	enc
e	enes
e	esen
n	nce
n	nese
s	sene
s	s

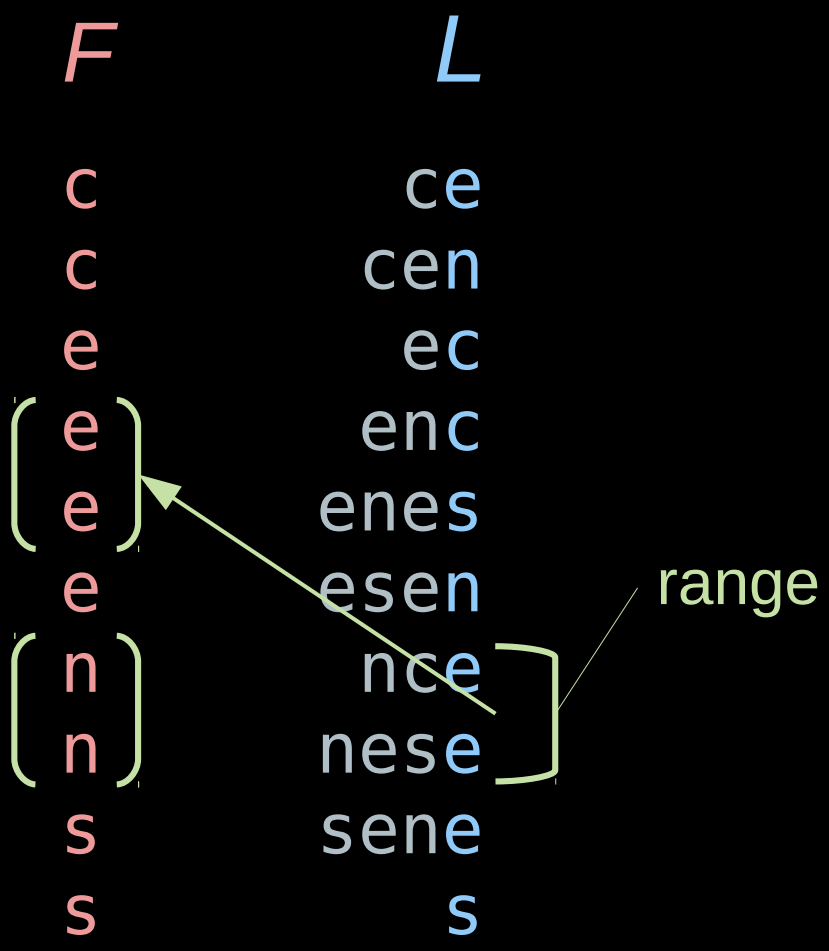
backward search 'cen'

s|enes|cen|ce

F	*L*
c	ce
c	cen
e	ec
e	enc
e	enes
e	esen
n	nce
n	nese
s	sene
s	s

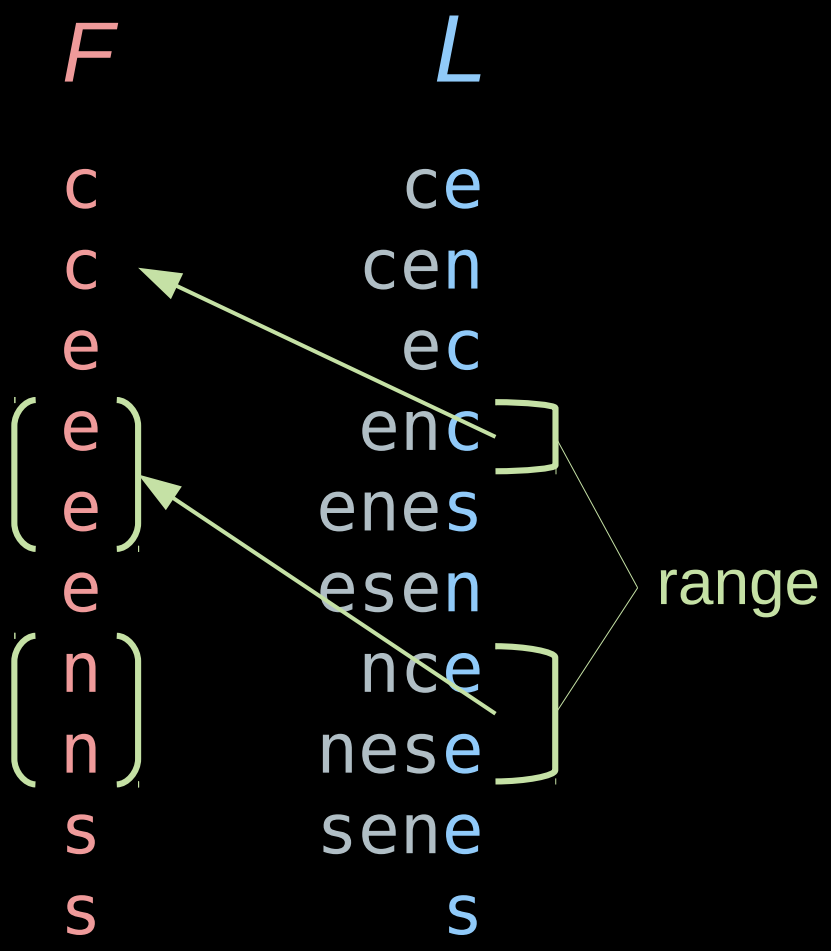
backward search 'cen'

s|enes|cen|ce

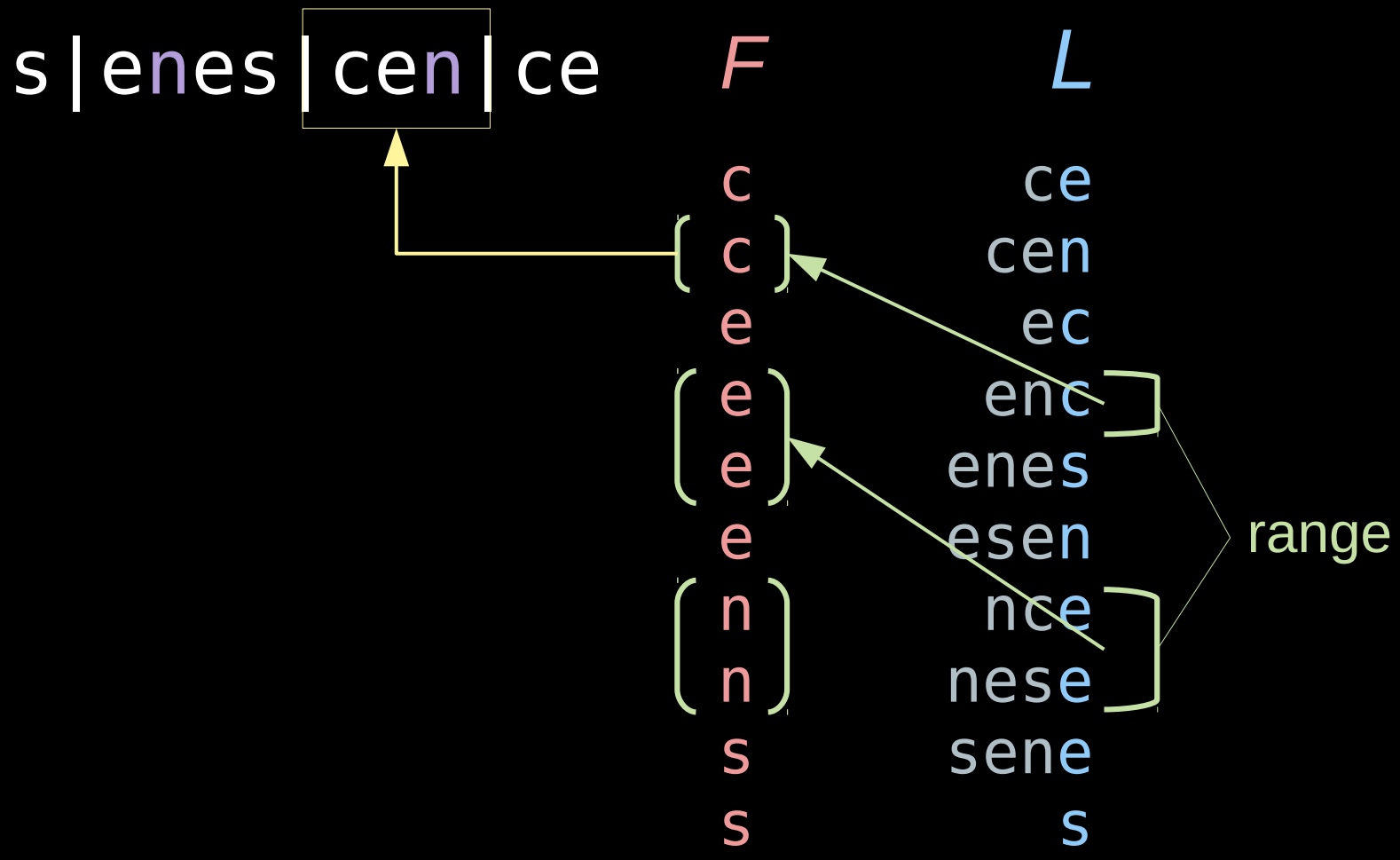


backward search 'cen'

s|enes|cen|ce



backward search 'cen'



backward search 'ss'

s|enes|cen|ce

F	*L*
c	ce
c	cen
e	ec
e	enc
e	enes
e	esen
n	nce
n	nese
s	sene
s	s

backward search 'ss'

s|enes|cen|ce

F	*L*
c	ce
c	cen
e	ec
e	enc
e	enes
e	esen
n	nce
n	nese
s	sene
s	s

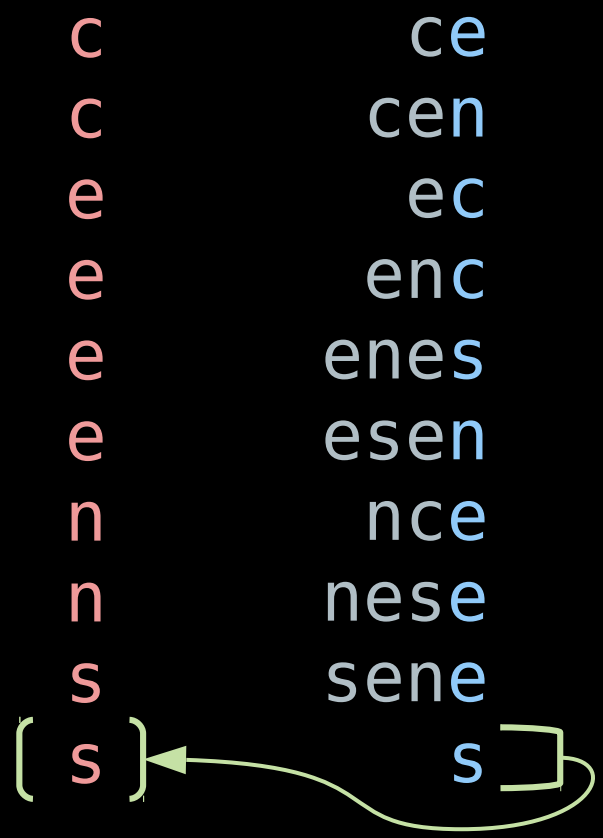
backward search 'ss'

s|enes|cen|ce

F	*L*
c	ce
c	cen
e	ec
e	enc
e	enes
e	esen
n	nce
n	nese
s	sene
s	s

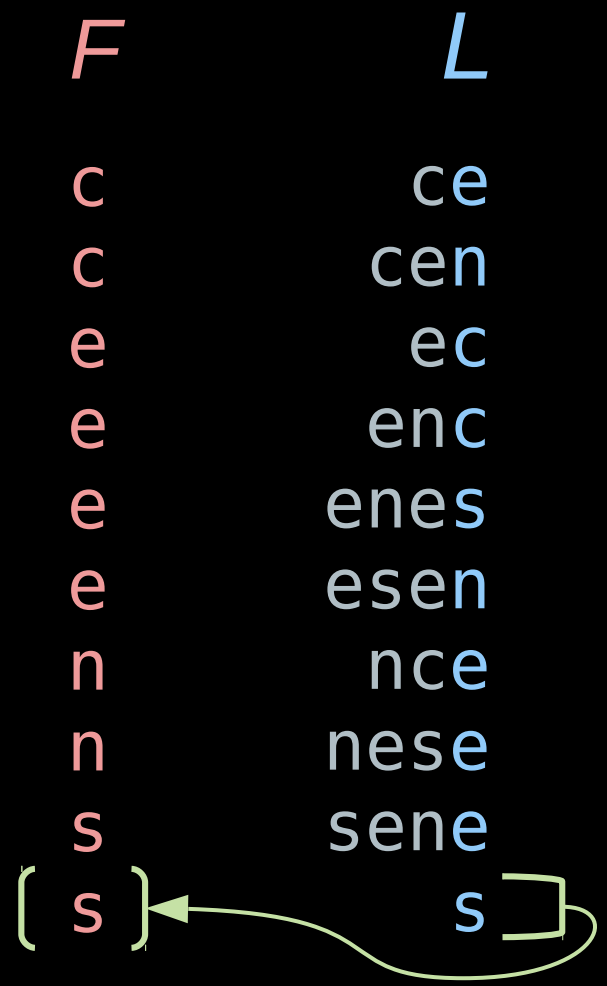
backward search 'ss'

s|enes|cen|ce



backward search 'ss'

s|enes|cen|ce



- cen is Lyndon word
- ss is **not**

pattern is Lyndon word

⇒ occurrences inside factors

⇒ found within cycles

backward search ≅ FM-index

pattern P is not a Lyndon word

- Lyndon factorization: $P = P_1 \cdots P_m$
- P_j substring of T_i or equal to T_i
- search P_m
- take care when starting with $P_{m\text{-}1}$!

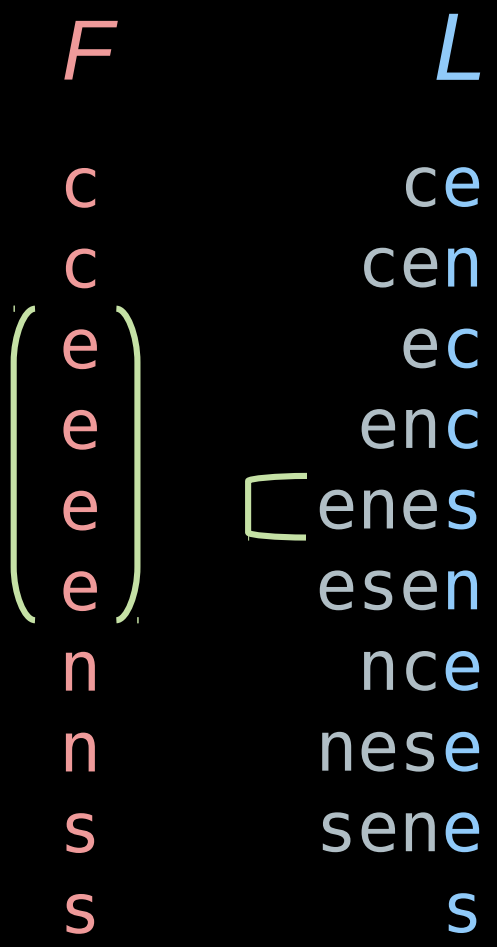
- backward search *P* = se

F	*L*
c	ce
c	cen
e	ec
e	enc
e	enes
e	esen
n	nce
n	nese
s	sene
s	s

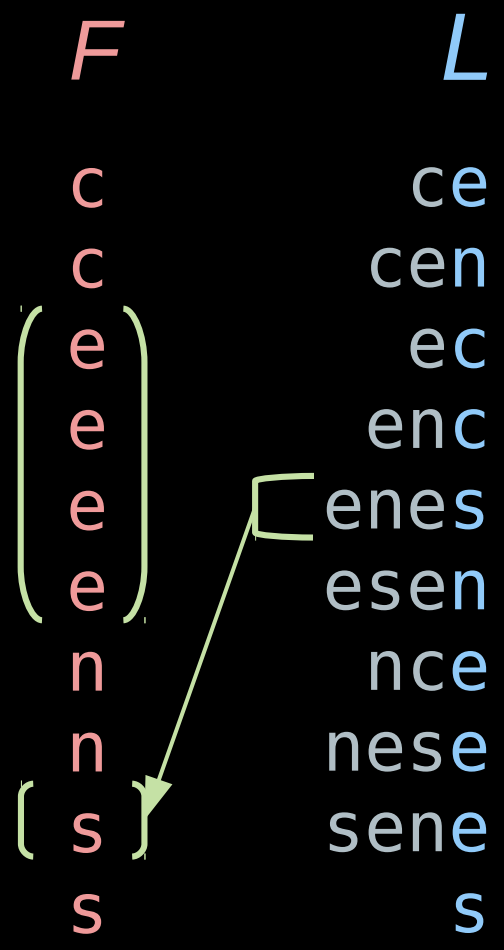
- backward search P = se
- P_2 = e

F	*L*
c	ce
c	cen
e	ec
e	enc
e	enes
e	esen
n	nce
n	nese
s	sene
s	s

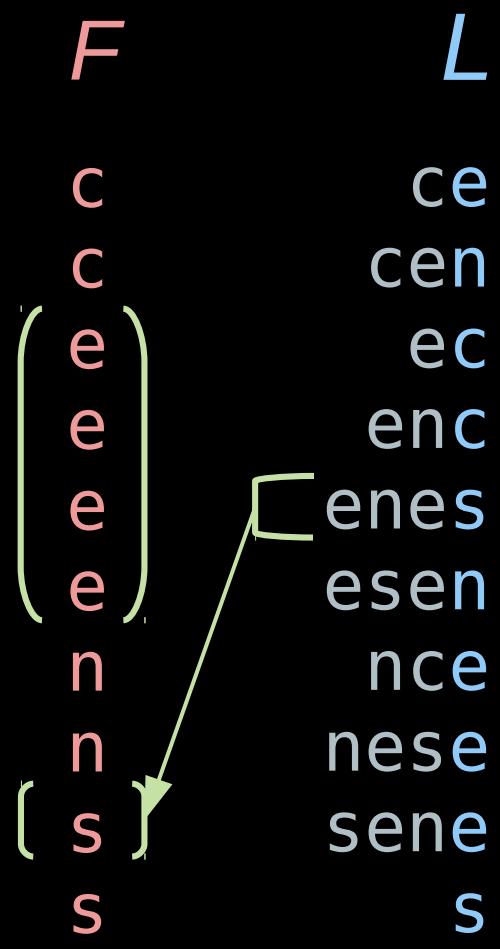
- backward search P = se
- P_2 = e



- backward search P = se
- P_2 = e
- P_1 = s



- backward search P = se
- P_2 = e
- P_1 = s

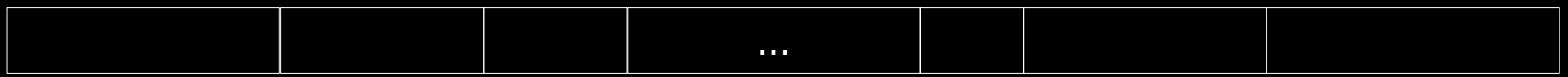


found

s|enes|cen|ce

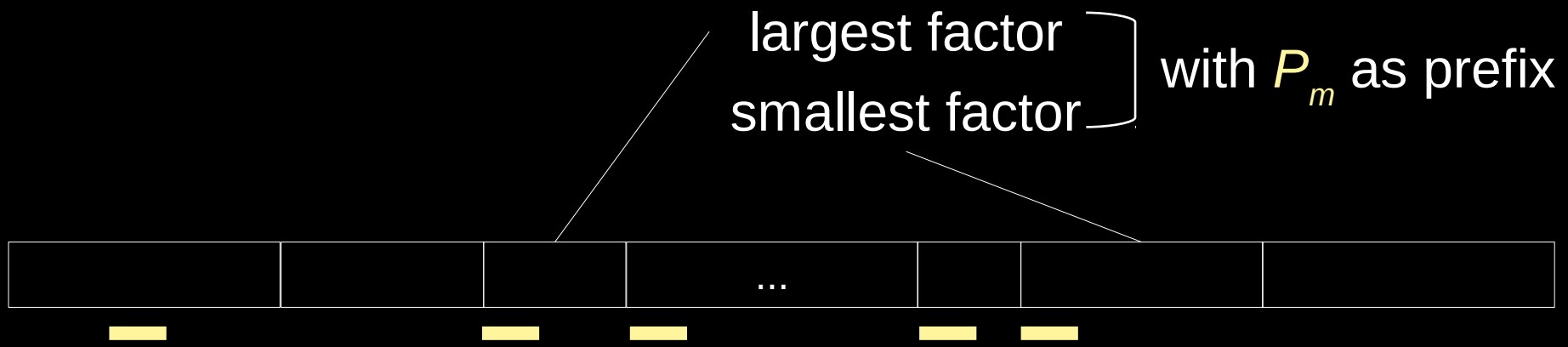
not counted

false occurrence

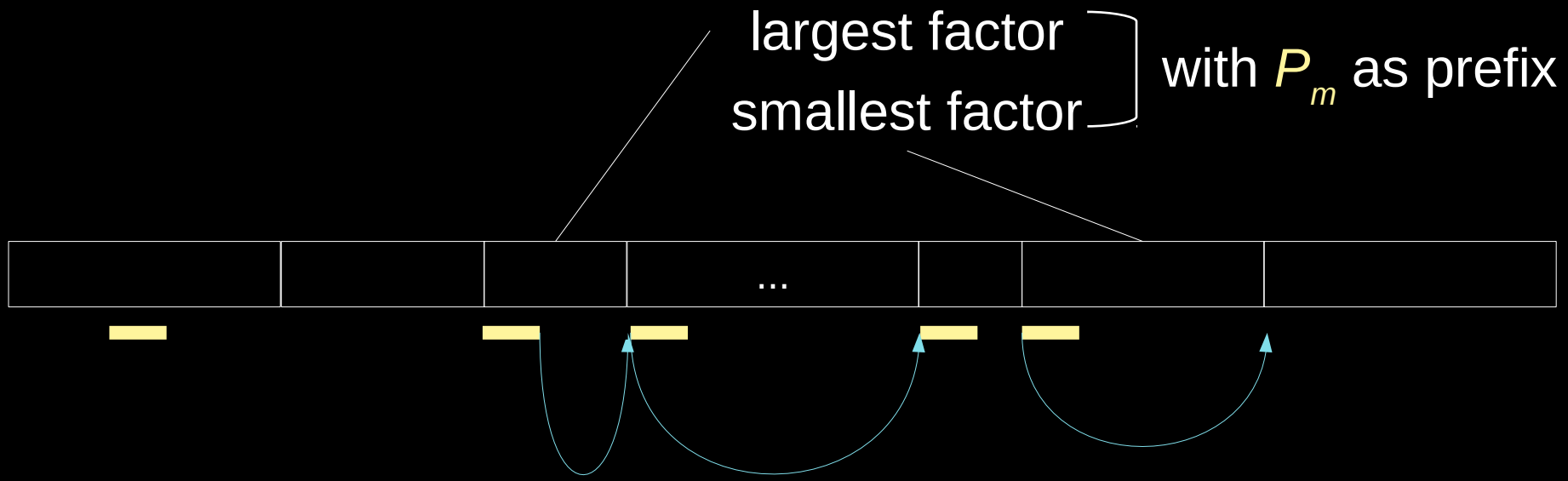


- backward search P_m

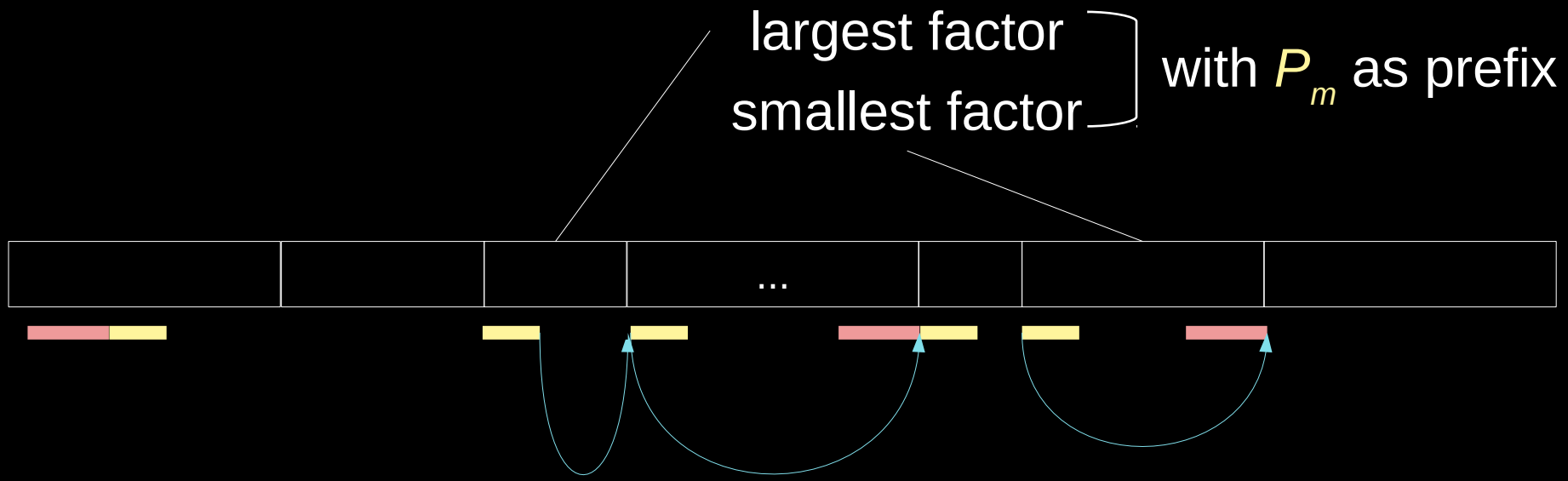
- backward search P_m



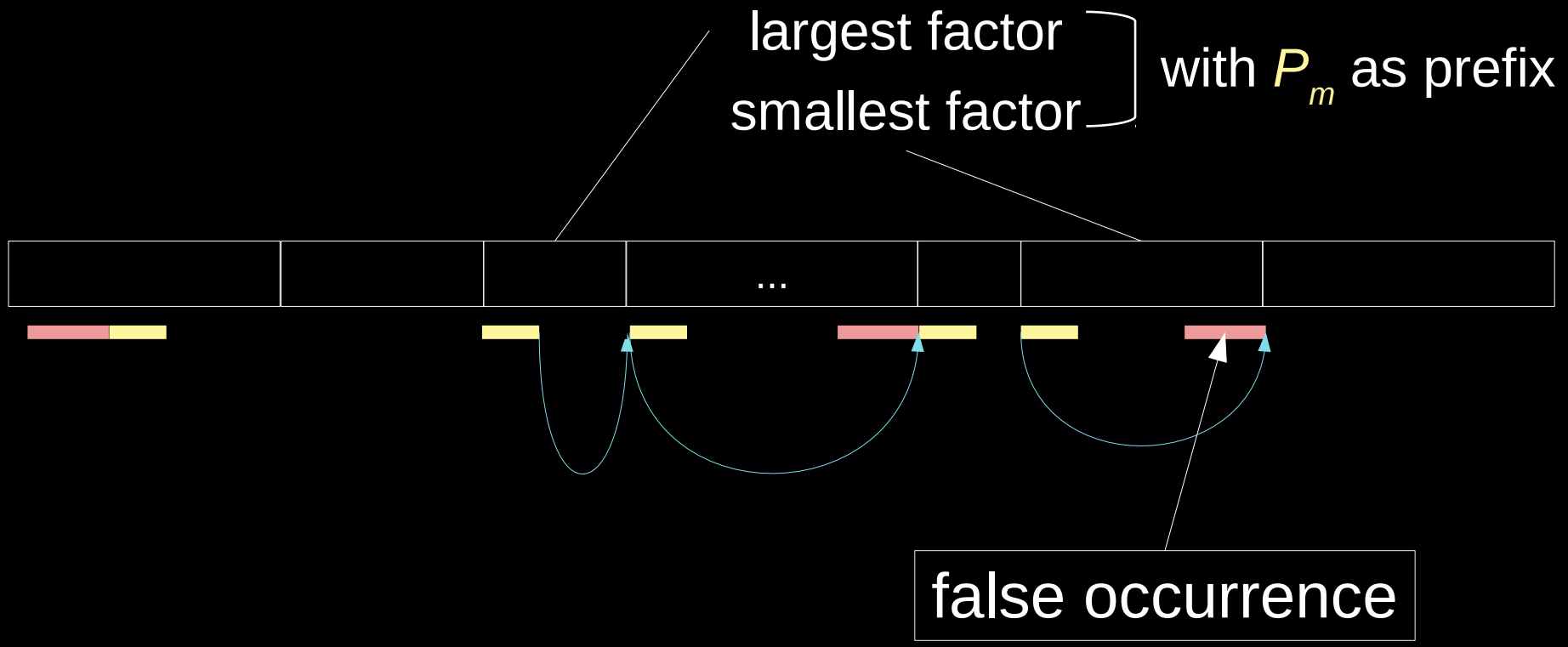
- backward search P_m
- continue search $P_{m-1} P_m$



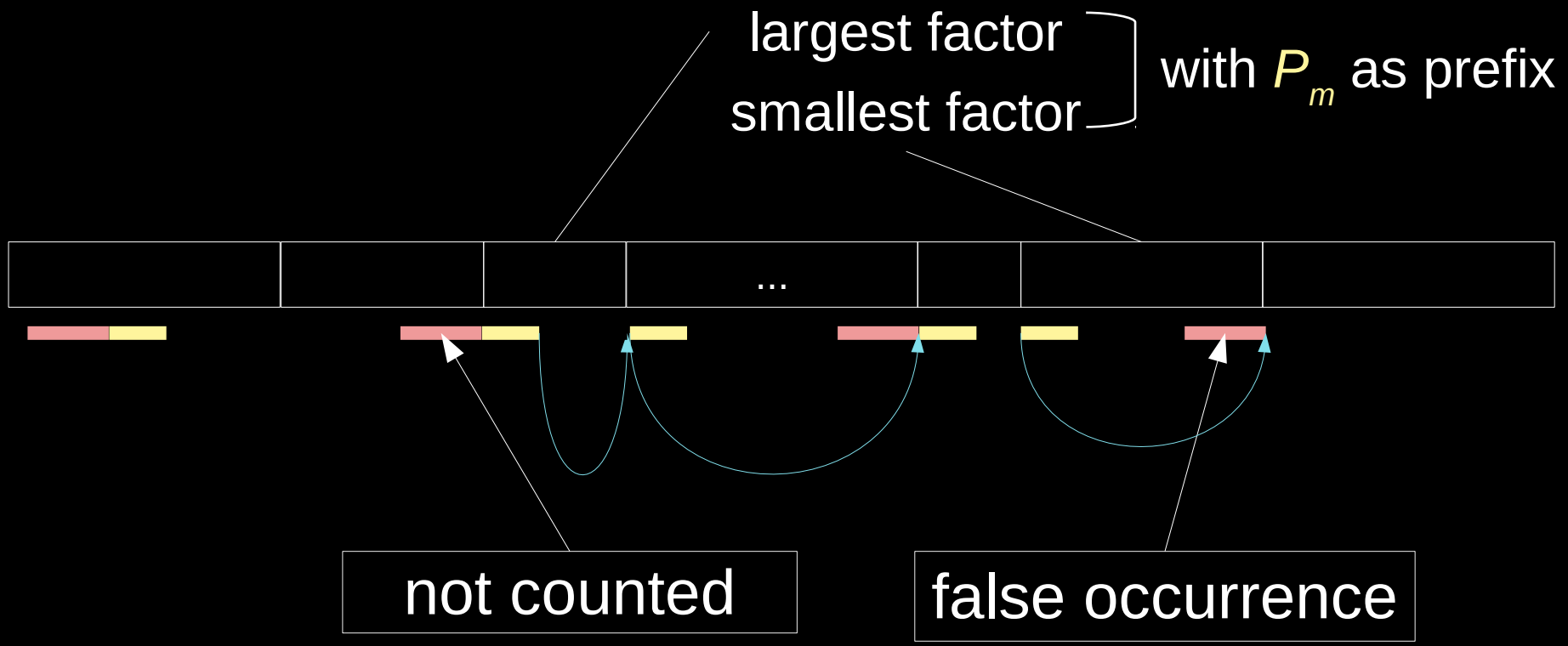
- backward search P_m
- continue search $P_{m-1} P_m$



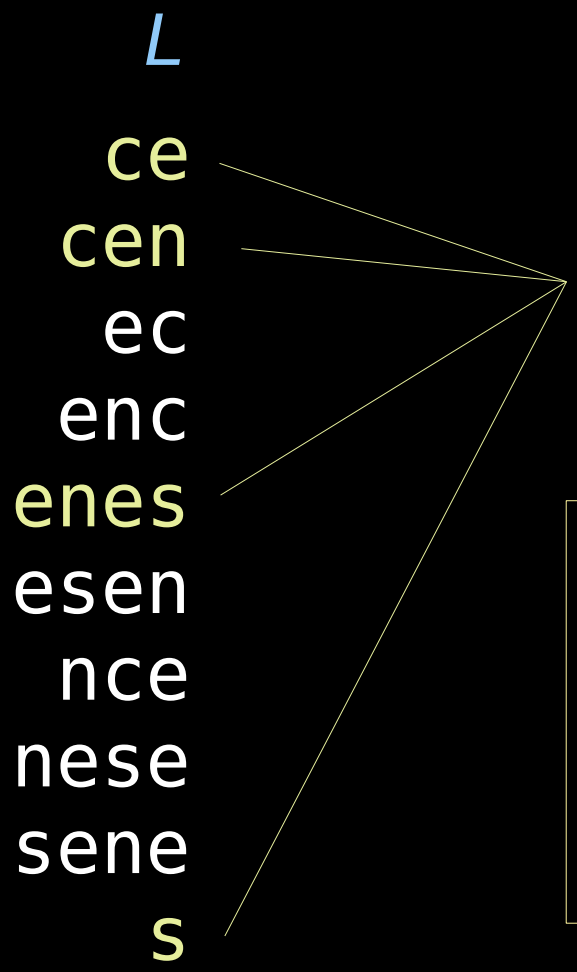
- backward search P_m
- continue search $P_{m-1} P_m$



- backward search P_m
- continue search $P_{m-1} P_m$



location of factors T_i



Lyndon factors $T_t \ldots T_1$

u, v Lyndon words:
$u <_{lex} v \iff u \prec_{\omega} v$

- after finding range of P_m :
  - for border $P_{m-1}P_m$ maintain
    - pointer to not-counted occurrence
    - pointer to false occurrence
- in total backward search on
  - range
  - at most $2m$ individual values

conclusion

- FM index with bijective BWT
- uses properties of Lyndon factorization on
  - text
  - pattern $P = P_1 \cdots P_m$
- currently O(m) times slower than FM index
- extended BWT does not seem to be a good candidate (no Lyndon word properties)

conclusion

- FM index with bijective BWT
- uses properties of Lyndon factorization on
  - text
  - pattern $P = P_1 \cdots P_m$
- currently O(m) times slower than FM index
- extended BWT does not seem to be a good candidate (no Lyndon word properties)

Thank you for your attention. Any questions are welcome!