Repeats in strings

Maxime Crochemore

King's College London & Université Paris-Est

Some types of repetitions

- **String = text = word = sequence of symbols**

- **Repetition = periodic string = power of exponent** ≥ 2

$$\overbrace{\texttt{a b a a b a b a a b a b a a b a b}}^{\text{length} = 17}$$

$$\underbrace{\texttt{a b a a b}}_{\text{period} = 5}$$

$$\text{exponent} = \frac{\text{length}}{\text{period}} = \frac{17}{5} = 3.4$$

Some types of repetitions

⋆ **String = text = word = sequence of symbols**

⋆ **Repetition = periodic string = power of exponent ≥ 2**

$$\texttt{abaab abaab abaab ab} = (\texttt{abaab})^{17/5}$$

$$\texttt{alfalfa} = (\texttt{alf})^{7/3} \qquad \texttt{entente} = (\texttt{ent})^{7/3}$$

Some types of repetitions

- **String = text = word = sequence of symbols**

- **Repetition = periodic string = power of exponent ≥ 2**

$$\texttt{abaab abaab abaab ab} = (\texttt{abaab})^{17/5}$$

$$\texttt{alfalfa} = (\texttt{alf})^{7/3} \qquad \texttt{entente} = (\texttt{ent})^{7/3}$$

- **Repeat:** $1 <$ **exponent** < 2

$$\overbrace{\underbrace{\texttt{a b a a b c c c c c}}_{\text{period} = 10}\ \texttt{a b a a b}}^{\text{length} = 15}$$

$$\text{exponent} = \frac{\text{length}}{\text{period}} = \frac{15}{10} = 1.5$$

Some types of repetitions

- **String = text = word = sequence of symbols**

- **Repetition = periodic string = power of exponent** ≥ 2

$$\texttt{abaab abaab abaab ab} = (\texttt{abaab})^{17/5}$$

$$\texttt{alfalfa} = (\texttt{alf})^{7/3} \qquad \texttt{entente} = (\texttt{ent})^{7/3}$$

- **Repeat:** $1 <$ **exponent** < 2

$$\texttt{abaab ccccc abaab} = (\texttt{abaabccccc})^{15/10}$$

$$\texttt{restore} = (\texttt{resto})^{7/5} \qquad \texttt{all in all} = (\texttt{all in })^{10/7}$$

Some types of repetitions

- ⋆ **String = text = word = sequence of symbols**

- ⋆ **Repetition = periodic string = power of exponent** ≥ 2

$$\texttt{abaab abaab abaab ab} = (\texttt{abaab})^{17/5}$$

$$\texttt{alfalfa} = (\texttt{alf})^{7/3} \qquad \texttt{entente} = (\texttt{ent})^{7/3}$$

- ⋆ **Repeat:** $1 <$ **exponent** < 2

$$\texttt{abaab ccccc abaab} = (\texttt{abaabccccc})^{15/10}$$

$$\texttt{restore} = (\texttt{resto})^{7/5} \qquad \texttt{all in all} = (\texttt{all in })^{10/7}$$

- ⋆ **Palindrome**

`abaab baaba`

Some types of repetitions

⋆ **String = text = word = sequence of symbols**

⋆ **Repetition = periodic string = power of exponent** ≥ 2

$$\texttt{abaab abaab abaab ab} = (\texttt{abaab})^{17/5}$$

$$\texttt{alfalfa} = (\texttt{alf})^{7/3} \qquad \texttt{entente} = (\texttt{ent})^{7/3}$$

⋆ **Repeat:** $1 <$ **exponent** < 2

$$\texttt{abaab ccccc abaab} = (\texttt{abaabccccc})^{15/10}$$

$$\texttt{restore} = (\texttt{resto})^{7/5} \qquad \texttt{all in all} = (\texttt{all in })^{10/7}$$

⋆ **Palindrome**

abaab baaba

CCAGA UUAAGGU UCUGG


www.pxleyes.com

Motivations

- **Pattern matching algorithms**
  String Matching, Time-space optimal String Matching: local and global periods, Indexing
- **Combinatorics on words**
  Avoidability of repetitions, Interaction between periods, Counting repetitions
- **Text Compression**
  Generalised run-length encoding, Dictionary-based compression

Motivations

- **Pattern matching algorithms**
  String Matching, Time-space optimal String Matching: local and global periods, Indexing
- **Combinatorics on words**
  Avoidability of repetitions, Interaction between periods, Counting repetitions
- **Text Compression**
  Generalised run-length encoding, Dictionary-based compression
- **Analysis of biological molecular sequences**
  Intensive study of satellites, Simple Sequence Repeats, or Tandem Repeats in DNA sequences, Molecular structure prediction, Phylogenies
- **Analysis of music**
  Rhythm detection, Chorus location

Huntington's Disease mRNA in EMBL

```
ID   L12392; SV 1; linear; mRNA; STD; HUM; 10348 BP.
...
DE   Homo sapiens Huntington's Disease (HD) mRNA, complete cds.
XX
KW   trinucleotide repeat.
XX
OS   Homo sapiens (human)
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia;
OC   Eutheria; Euarchontoglires; Primates; Haplorrhini; Catarrhini; Hominidae;
OC   Homo.
XX
RN   [1]
RP   1-10348
RX   PUBMED; 8458085.
RA   MacDonald M., Ambrose C.M.;
RT   "A novel gene containing a trinucleotide repeat that is expanded and
RT   unstable on Huntington's disease chromosomes. The Huntington's Disease
RT   Collaborative Research Group [see comments]";
RL   Cell 72(6):971-983(1993).
...
```

Polyglutamine repetition

```
...
FT   CDS             316..9750
...
FT                   /protein_id="AAB38240.1"
FT                   /translation="MATLEKLMKAFESLKSFQQQQQQQQQQQQQQQQQQQQQQQPPPPP
FT                   PPPPPPQLPQPPPQAQPLLPQPQPPPPPPPPPPGPAVAEEPLHRPKKELSATKKDRVNH
...                  ...
FT                   FQSVLEVVAAPGSPYHRLLTCLRNVHKVTTC"
FT   polyA_site      10348
FT                   /gene="HD"
XX
SQ   Sequence 10348 BP; 2408 A; 2807 C; 2744 G; 2389 T; 0 other;
     ttgctgtgtg aggcagaacc tgcgggggca ggggcgggct ggttccctgg ccagccattg        60
     gcagagtccg caggctaggg ctgtcaatca tgctggccgg cgtggccccg cctccgccgg       120
     cgcggccccg cctccgccgg cgcacgtctg ggacgcaagg cgccgtgggg gctgccggga       180
     cgggtccaag atggacggcc gctcaggttc tgcttttacc tgcggcccag agccccattc       240
     attgccccgg tgctgagcgg cgccgcgagt cggcccgagg cctccgggga ctgccgtgcc       300
     gggcgggaga ccgccATGgc gaccctggaa aagctgatga aggccttcga gtccctcaag       360
     tccttcCAGC AGCAGCAGCA GCAGCAGCAG CAGCAGCAGC AGCAGCAGCA GCAGCAGCAG       420
     CAGCAGCAGC AACAGccgcc accgccgccg ccgccgccgc cgcctcctca gcttcctcag       480
     ccgccgccgc aggcacagcc gctgctgcct cagccgcagc cgcccccgcc gccgcccccg       540
...  ...
     atatcagtaa agagattaat tttaacgt                                        10348
//
```

Avoiding repetitions

⋆ **Theorem 1 ([Thue, 1906, 1912])**
There are infinite binary strings with no overlap
(that is, no repetition of exponent > 2)**.**
There are infinite ternary strings with no square.

Avoiding repetitions

- **Theorem 2 ([Thue, 1906, 1912])**
  There are infinite binary strings with no overlap
  (that is, no repetition of exponent > 2)***.***
  There are infinite ternary strings with no square.

- ## Iterated morphisms

  - **no overlap in t:**

$$\begin{cases} t(\mathsf{0}) = \mathsf{01}, \\ t(\mathsf{1}) = \mathsf{10}. \end{cases}$$

  $\mathbf{t} = t^{\infty}(\mathsf{0}) = \mathsf{0110100110010110100101110}\ldots$

  - **no square in f:**

$$\begin{cases} f(\mathsf{a}) = \mathsf{abc}, \\ f(\mathsf{b}) = \mathsf{ac}, \\ f(\mathsf{c}) = \mathsf{b}. \end{cases}$$

  $\mathbf{f} = f^{\infty}(\mathsf{a}) = \mathsf{abcacbabcbacabcacbacabcb}\ldots$

Dejean's framework

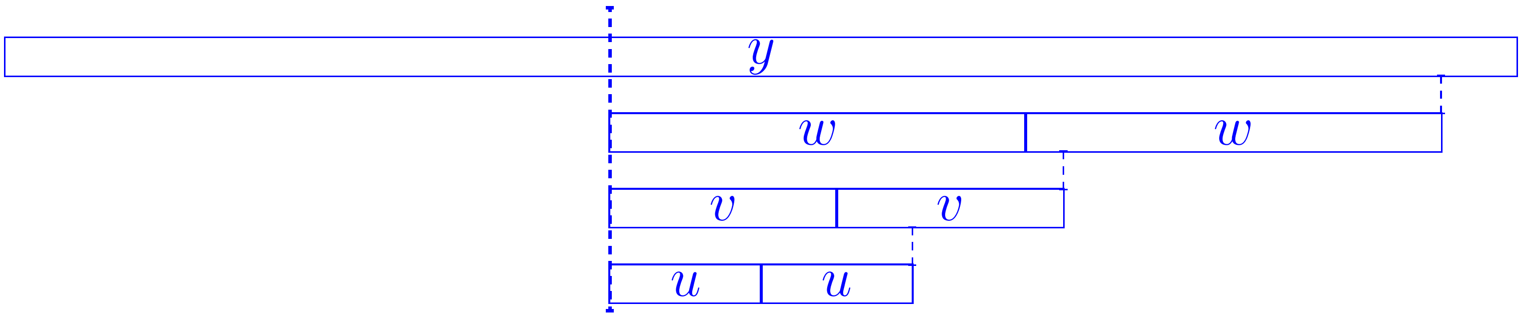
- **Repetitive threshold**
  $\mathrm{RT}(a) =$ **minimal rational** r **for which there exists an infinite word on** a **letters whose maximal exponent of factors is** r
- **Theorem 3**

$$\begin{cases} \mathrm{RT}(2) = 2 \\ \mathrm{RT}(3) = 7/4 \\ \mathrm{RT}(4) = 7/5 \\ \mathrm{RT}(k) = k/(k-1) \end{cases}$$

- **Multi-author proof:**
  [Thue, 1906], [Dejean, 1972], [Pansiot, 1984], [Moulin-Ollagnier, 1992], [Carpi, 2007], [Rao, 2009], [Currie, Rampersad, 2009]

How many squares in a word?

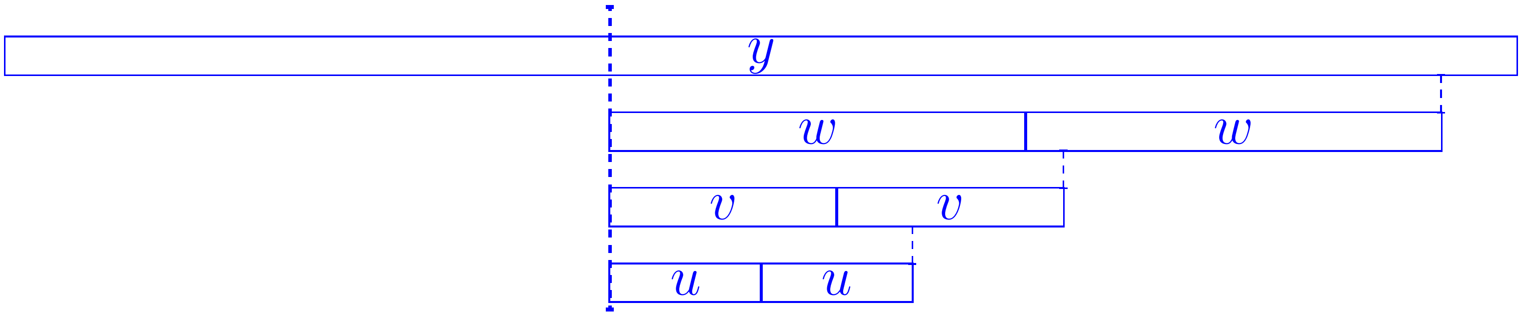
⋆ **Proposition 1 ([Fraenkel, Simpson, 1998])**
No more than $2n$ ***primitively-rooted squares.***



largest position of u^2, v^2, and w^2 in y? impossible!

How many squares in a word?

- **Proposition 2 ([Fraenkel, Simpson, 1998])**
  No more than $2n$ ***primitively-rooted squares.***



largest position of u^2, v^2, and w^2 in y? impossible!

- **Direct proofs [Hickerson, 2004], [Ilie, 2005]**
- **Best bounds:** $2n - \Theta(\log n)$ **[Ilie, 2005],** $\frac{95}{48}n$ **[Lam, 2013],** $\frac{11}{6}n$ **[Deza, Franek, Thierry, 2014]**
- **Computation in time** $O(n \log a)$ **[Gusfield, Stoye, 1999]**
- **Proposition 3 ([C., 1981], [Gusfield, Stoye, 1999])**
  Maximal number of occurrences of primitively-rooted squares : $cn \log n$. ***Attained by Fibonacci words.***

How few squares in a word?

- **Proposition 4 ([Fraenkel, Simpson, 1995])**
  There is an infinite binary word containing only 3 squares, 2 cubes, and no other repetition of exponent ≥ 2.
- **Several other proofs: [Rampersad, Shallit, Wang, 2005], [Harju, Nowotka, 2006], [Badkobeh, C., 2010]**

How few squares in a word?

- **Proposition 5 ([Fraenkel, Simpson, 1995])**
  There is an infinite binary word containing only 3 squares, 2 cubes, and no other repetition of exponent ≥ 2.
- **Several other proofs: [Rampersad, Shallit, Wang, 2005], [Harju, Nowotka, 2006], [Badkobeh, C., 2010]**
- **Morphism h_0:**

$$\begin{cases} h_0(\mathtt{a}) = \mathtt{01001110001101}, \\ h_0(\mathtt{b}) = \mathtt{0011}, \\ h_0(\mathtt{c}) = \mathtt{000111}. \end{cases}$$

  $\mathbf{h_0} = h_0(f^{\infty}(\mathtt{a}))$ **contains:**

  - **the** 3 **squares** $\mathtt{00}$**,** $\mathtt{11}$**,** $\mathtt{1010}$
  - **the** 2 **cubes** $\mathtt{000}$ **and** $\mathtt{111}$
  - **no other repetition of exponent** ≥ 2

How few squares in a repetition-constrained word?

- ⋆ **Theorem 4 ([Karhumäki, Shallit, 2004], [Shallit, 2008])**
  There is an infinite binary word avoiding $7/3^+$-powers with finitely many squares. $7/3$ is the smallest such exponent.

How few squares in a repetition-constrained word?

- **Theorem 5 ([Karhumäki, Shallit, 2004], [Shallit, 2008])**
  There is an infinite binary word avoiding $7/3^+$-powers with finitely many squares. $7/3$ is the smallest such exponent.
- **Theorem 6 ([Badkobeh, C., 2010])**
  . . . with 12 squares, the fewest possible.

$$\begin{cases} g(\mathtt{a}) = \mathtt{abac}, \\ g(\mathtt{b}) = \mathtt{babd}, \\ g(\mathtt{c}) = \mathtt{eabdf}, \\ g(\mathtt{d}) = \mathtt{fbace}, \\ g(\mathtt{e}) = \mathtt{bace}, \\ g(\mathtt{f}) = \mathtt{abdf}. \end{cases} \qquad \begin{cases} h(\mathtt{a}) = \mathtt{10011}, \\ h(\mathtt{b}) = \mathtt{01100}, \\ h(\mathtt{c}) = \mathtt{01001}, \\ h(\mathtt{d}) = \mathtt{10110}, \\ h(\mathtt{e}) = \mathtt{0110}, \\ h(\mathtt{f}) = \mathtt{1001}. \end{cases}$$

- **$\mathbf{h} = h(g^\infty(\mathtt{a}))$ contains:**
  - 12 **squares,** 2 $7/3$**-powers (**`0110110` **and** `1001001`**)**
  - **no other repetition of exponent** ≥ 2

Finite-Repetition Threshold

⋆ **Finite-Repetition Thresholds for the binary alphabet [Badkobeh, 2010]**

Maximal exponent e	Allowed number of e-powers	Minimum number of squares
7/3	2	12
	1	14
5/2	2	8
	1	11
3	2	3
	1	4

Finite-Repetition Threshold

- **Finite-Repetition Thresholds for the binary alphabet [Badkobeh, 2010]**

Maximal exponent e	Allowed number of e-powers	Minimum number of squares
7/3	**2**	**12**
	1	**14**
5/2	**2**	**8**
	1	**11**
3	**2**	**3**
	1	**4**

- **…for three letters:** $\mathrm{FRt}(3) = \mathrm{RT}(3) = 7/4$ **with 2** $7/4$**-powers**
- **…for four letters:** $\mathrm{FRt}(4) = \mathrm{RT}(4) = 7/5$ **with 2** $7/5$**-powers**
- **…for five letters:** $\mathrm{FRt}(5) = \mathrm{RT}(5) = 5/4$ **with 60** $5/4$**-powers**
- **and** $\mathrm{FRt}(k) = \mathrm{RT}(k)$**,** $k \geq 6$ **[Badkobeh, C., Rao, 2013]**

Runs

⋆ **Repetition = periodic string = power: exponent ≥ 2**

$$\texttt{abaab abaab abaab ab} = (\texttt{abaab})^{17/5}$$

⋆ **Run = maximal periodicity = maximal occurrence of a repetition**

position i center $i+p$ period p

.. c $\underbrace{\texttt{f a b a a b}}_{\neq}$ a b a a b a b a a b a b $\underbrace{\texttt{a a b a b g}}_{\neq}$ k ..

Runs

⋆ **Repetition = periodic string = power: exponent ≥ 2**

$$\texttt{abaab abaab abaab ab} = (\texttt{abaab})^{17/5}$$

⋆ **Run = maximal periodicity = maximal occurrence of a repetition**

$$..\texttt{cfabaababaababaababaababgk}..$$

Runs

- **Repetition = periodic string = power: exponent** ≥ 2

$$\texttt{abaab abaab abaab ab} = (\texttt{abaab})^{17/5}$$

- **Run = maximal periodicity = maximal occurrence of a repetition**

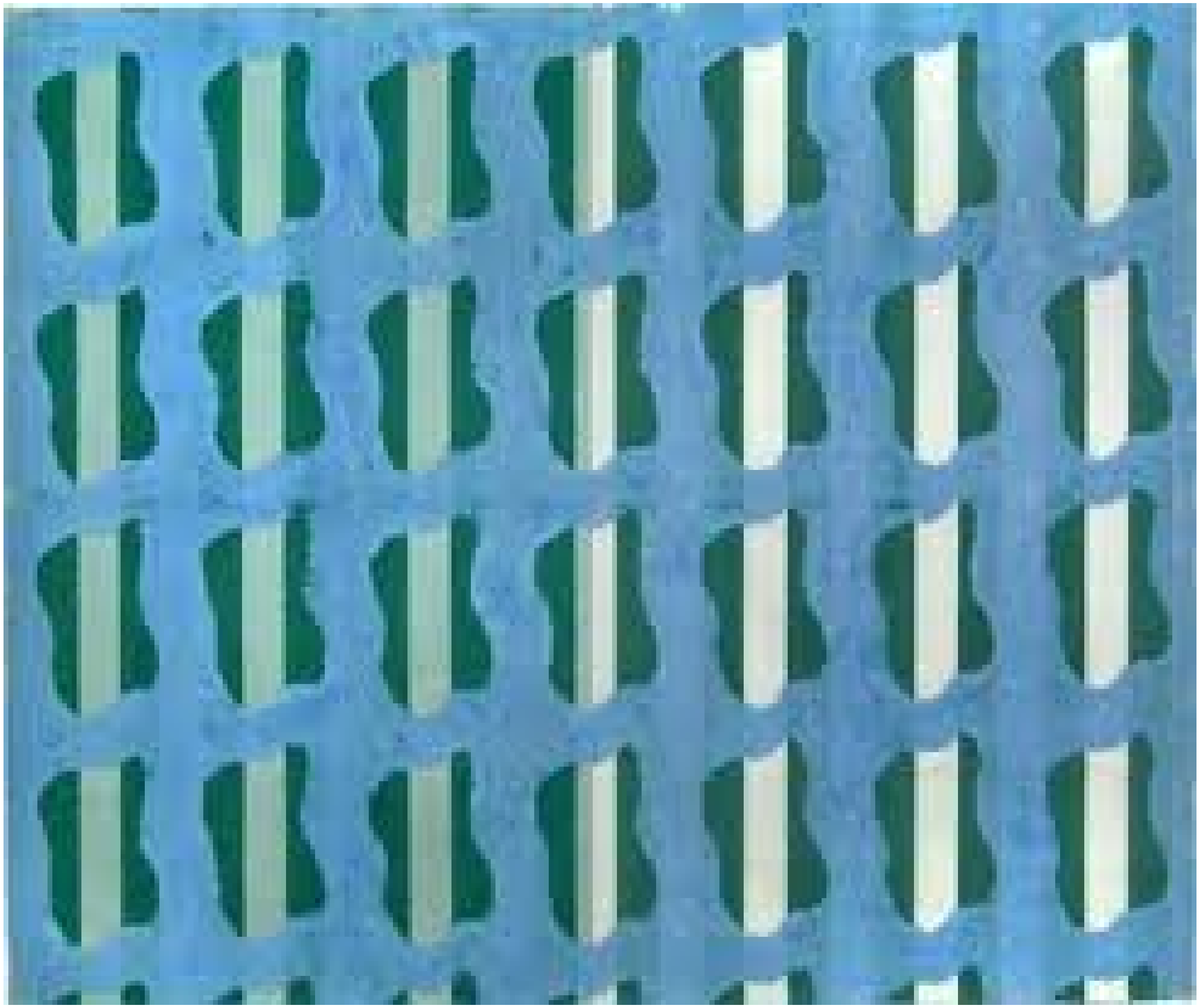
$$..\texttt{cfabaababaababaababaababgk}..$$

- 10 **runs in** `a a b b a a b b a a a b b a a a b b`

repetitions $= \{\mathsf{aa}, \mathsf{aaa}, \mathsf{bb}, (\mathsf{aabb})^{5/2}, (\mathsf{aabba})^{14/5}\}$

- **Notion introduced by [Iliopoulos, Moore, Smyth, 1997]**


Campbell's
SOUP
GOLDEN
MUSHROOM
HOT DOG BEAN
NEW ENGLAND
CLAM CHOWDER
VEGETARIAN
VEGETABLE
CHEDDAR
CHEESE
OYSTER STEW



How many runs in a string?

- **Useful for any algorithm dealing with repetitions in string**
- **Word of length** 18 **with** 10 **runs**

  `a a b b a a b b a a a b b a a a b b`

- **Theorem 7 ([Kolpakov, Kucherov, 1999])**
  There is no more than a linear number of runs in a string.

How many runs in a string?

- Useful for any algorithm dealing with repetitions in string
- Word of length 18 with 10 runs

`a a b b a a b b a a a b b a a a b b`

- **Theorem 8** ([Kolpakov, Kucherov, 1999])
  There is no more than a linear number of runs in a string.

  Conjecture 1 (Kolpakov, Kucherov, 1999)
  A string contains less runs than its length

- In binary strings:

n	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
runs(n)	2	3	4	5	5	6	7	8	8	10	10	11	12	13	14

n	20	21	22	23	24	25	26	27	28	29	30	31
runs(n)	15	15	16	17	18	19	20	21	22	23	24	25

Known bounds on runs

- **Upper bounds**
  - $5n$ [Rytter, 2006]
  - $3.44n$ [Rytter, 2007][Puglisi, Simpson, Smyth, 2007]
  - $1.6n$ [C., Ilie, 2007]

  with computer verification

  - $1.29n$ for binary strings [Giraud, 2009]
  - $1.029n$ [C., Ilie, Tinta, 2008]

Known bounds on runs

- **Upper bounds**
  - $5n$ [Rytter, 2006]
  - $3.44n$ [Rytter, 2007][Puglisi, Simpson, Smyth, 2007]
  - $1.6n$ [C., Ilie, 2007]

  with computer verification

  - $1.29n$ for binary strings [Giraud, 2009]
  - $1.029n$ [C., Ilie, Tinta, 2008]
- **Lower bounds**
  - $\frac{3}{1+\sqrt{5}}n \approx 0.927n$ [Franek, Simpson, Smyth, 2003]
  - $0.94457564n$
    [Kusano, Matsubara, Ishino, Bannai, Shinohara, 2008]
  - $0.944575712n$ [Simpson, 2009]

How many cubic runs in a string?

`b a b a a a b a a a b a a a b b a a b b b b a`

- ⋆ **Maximal periodicities of exponent** ≥ 3
- ⋆ **Upper bound:** $0.5n$**. Lower bound:** $0.406n$

How many cubic runs in a string?

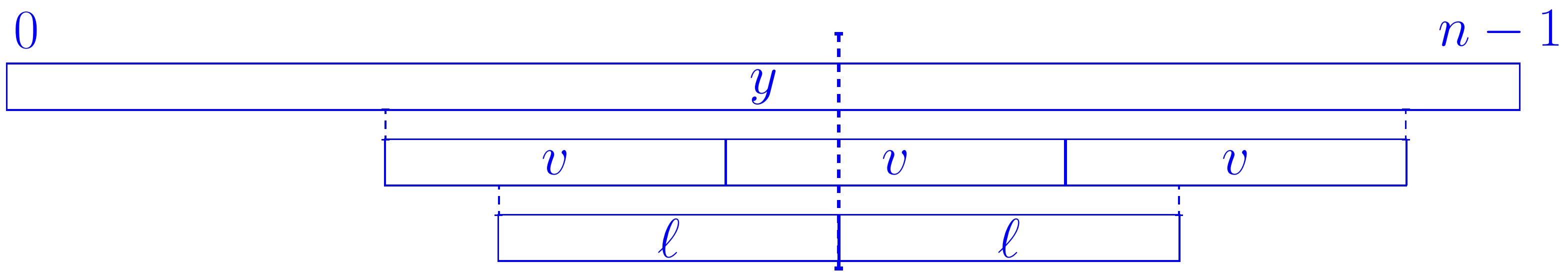
b a b a a a b a a a b a a a b b a a b b b b a

* Maximal periodicities of exponent ≥ 3
* **Upper bound:** $0.5n$. **Lower bound:** $0.406n$
* # occurrences of primitively-rooted cubes can be $\Omega(n \log n)$
* **No obvious relation with the number of (distinct) cubes:**

b a a a c a a a d a a a e a a a f .. $\begin{cases} 1 & \textbf{cube} \\ n/4 & \textbf{runs} \end{cases}$

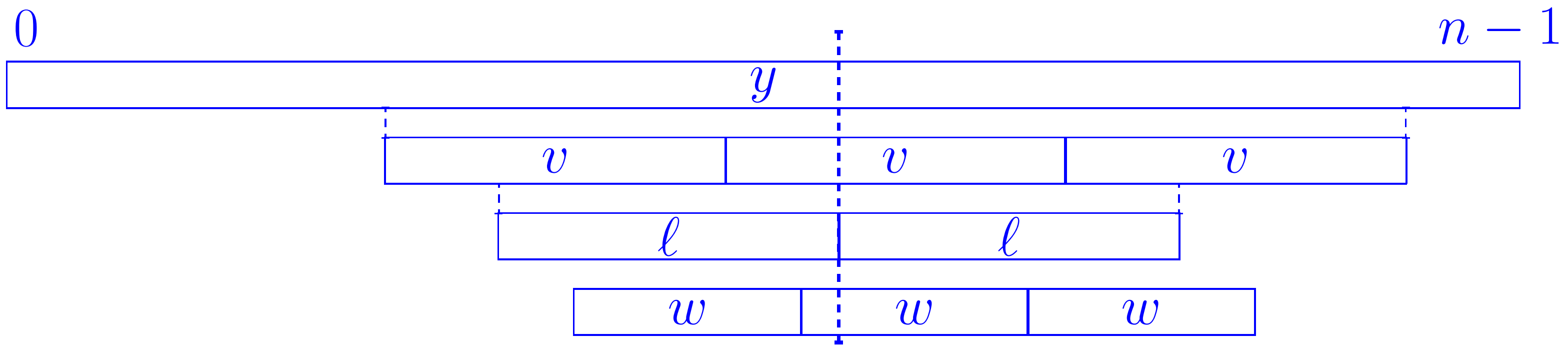
a b b a a b b a a b b a a b b a $\begin{cases} n/4 & \textbf{cubes} \\ 1 & \textbf{run} \end{cases}$

Upper bound on cubic runs



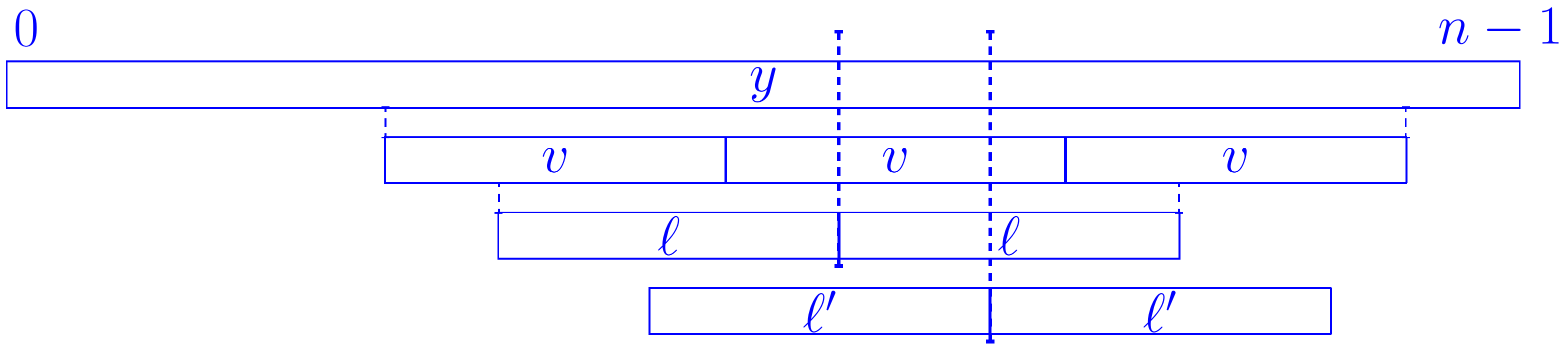
ℓ L-root: Lyndon root of run, smallest rotation of v

Upper bound on cubic runs



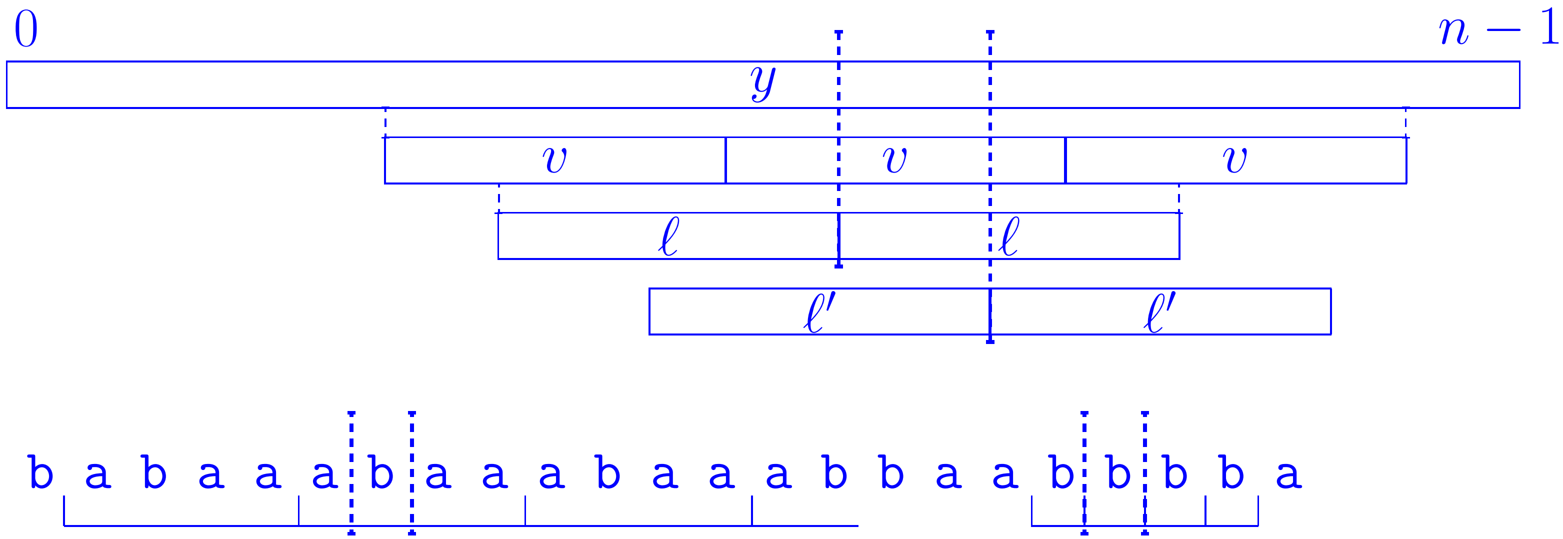
impossible

Upper bound on cubic runs

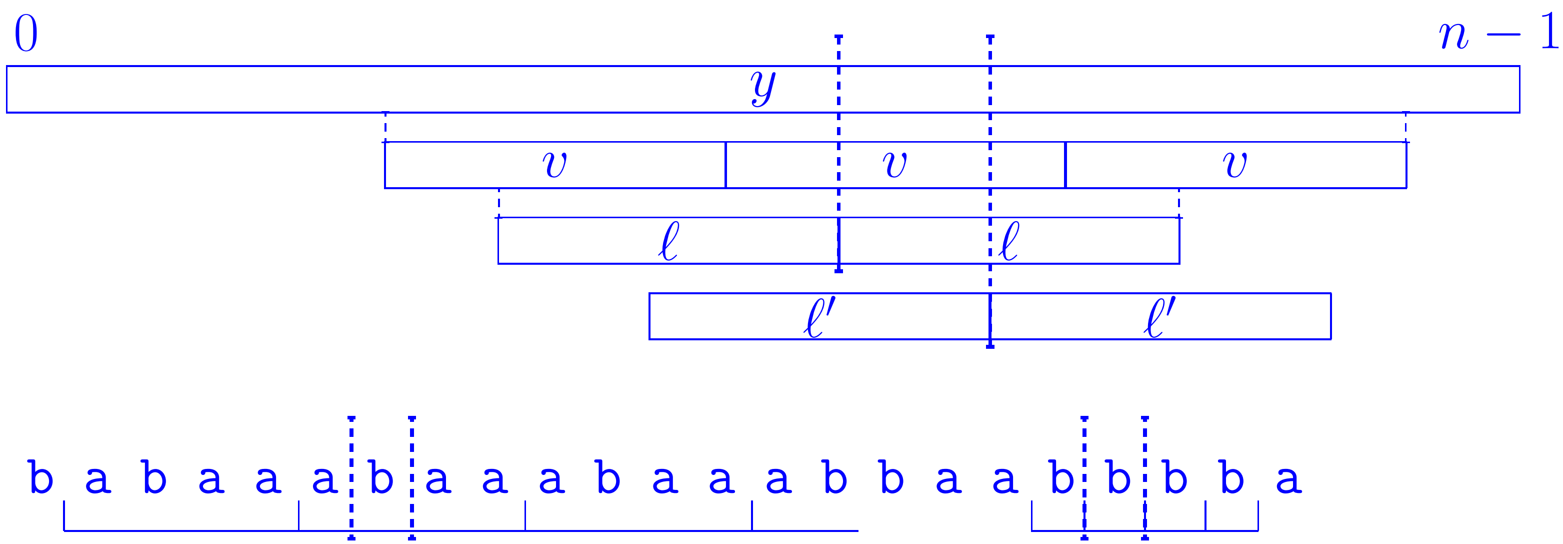


ℓ' largest rotation of v

Upper bound on cubic runs



Upper bound on cubic runs



- ⋆ **the two (inter-)positions are associated with only one run**
- ⋆ **thus: no more than** $(n-1)/2$ **runs with exponent** ≥ 3

Latest news on bounds on runs

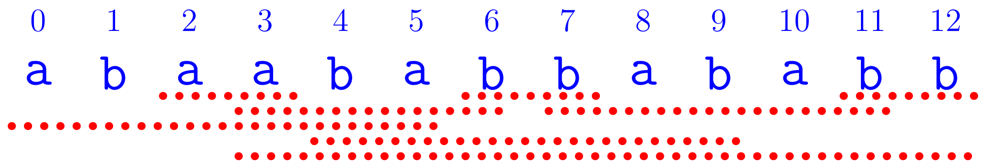
- **Upper bounds**
  - $\mathbf{5}n$ [Rytter, 2006]
  - $\mathbf{3.44}n$ [Rytter, 2007][Puglisi, Simpson, Smyth, 2007]
  - $\mathbf{1.6}n$ [C., Ilie, 2007]
  - $\mathbf{1.5}n$
    [Bannai, I, Inenaga, Nakashima, Takeda, Tsuruta, 2014]

  with computer verification

  - $\mathbf{1.29}n$ for binary strings [Giraud, 2008]
  - $\mathbf{1.029}n$ [C., Ilie, Tinta, 2008]

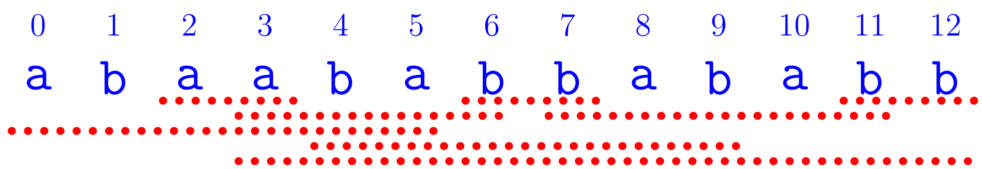
Lyndon roots

⋆ 8 **Runs in** abaababbababb

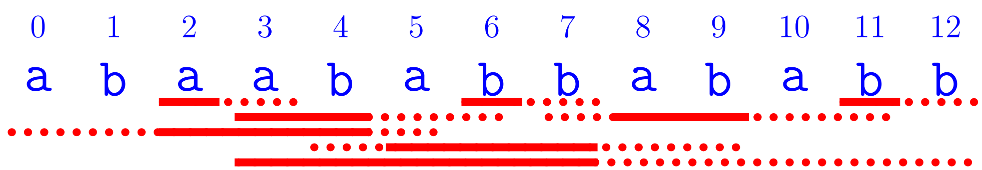


Lyndon roots

⋆ 8 **Runs in** abaababbababb



⋆ **Their L-roots** (a $<$ b)



Lyndon roots

⋆ 8 **Runs in** abaababbababb

0	1	2	3	4	5	6	7	8	9	10	11	12
a	b	a	a	b	a	b	b	a	b	a	b	b

⋆ **Their L-roots** (a $<$ b)

0	1	2	3	4	5	6	7	8	9	10	11	12
a	b	a	a	b	a	b	b	a	b	a	b	b

⋆ **L-roots in Lyndon word** #abaababbababb (# $<$ a $<$ b)

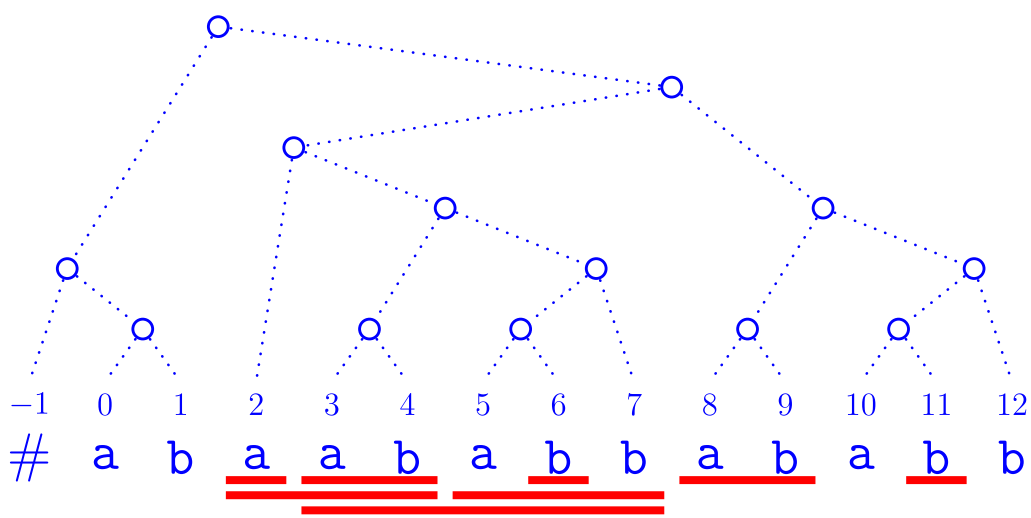
-1	0	1	2	3	4	5	6	7	8	9	10	11	12
#	a	b	a	a	b	a	b	b	a	b	a	b	b

Lyndon tree

⋆ **Standard factorisation:**
any Lyndon word x is a letter or uniquely factorises into uv where u, v are Lyndon words and $u < v$

Lyndon tree

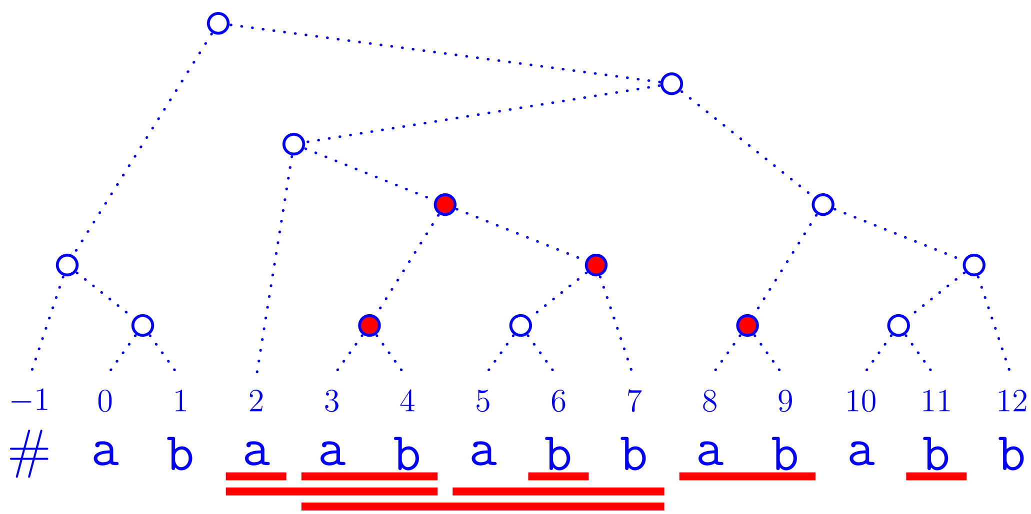
- **Standard factorisation:**
  any Lyndon word x is a letter or uniquely factorises into uv where u, v are Lyndon words and $u < v$
- **Leads to the Lyndon tree of a Lyndon word**



- **Nodes are associated with Lyndon factors**

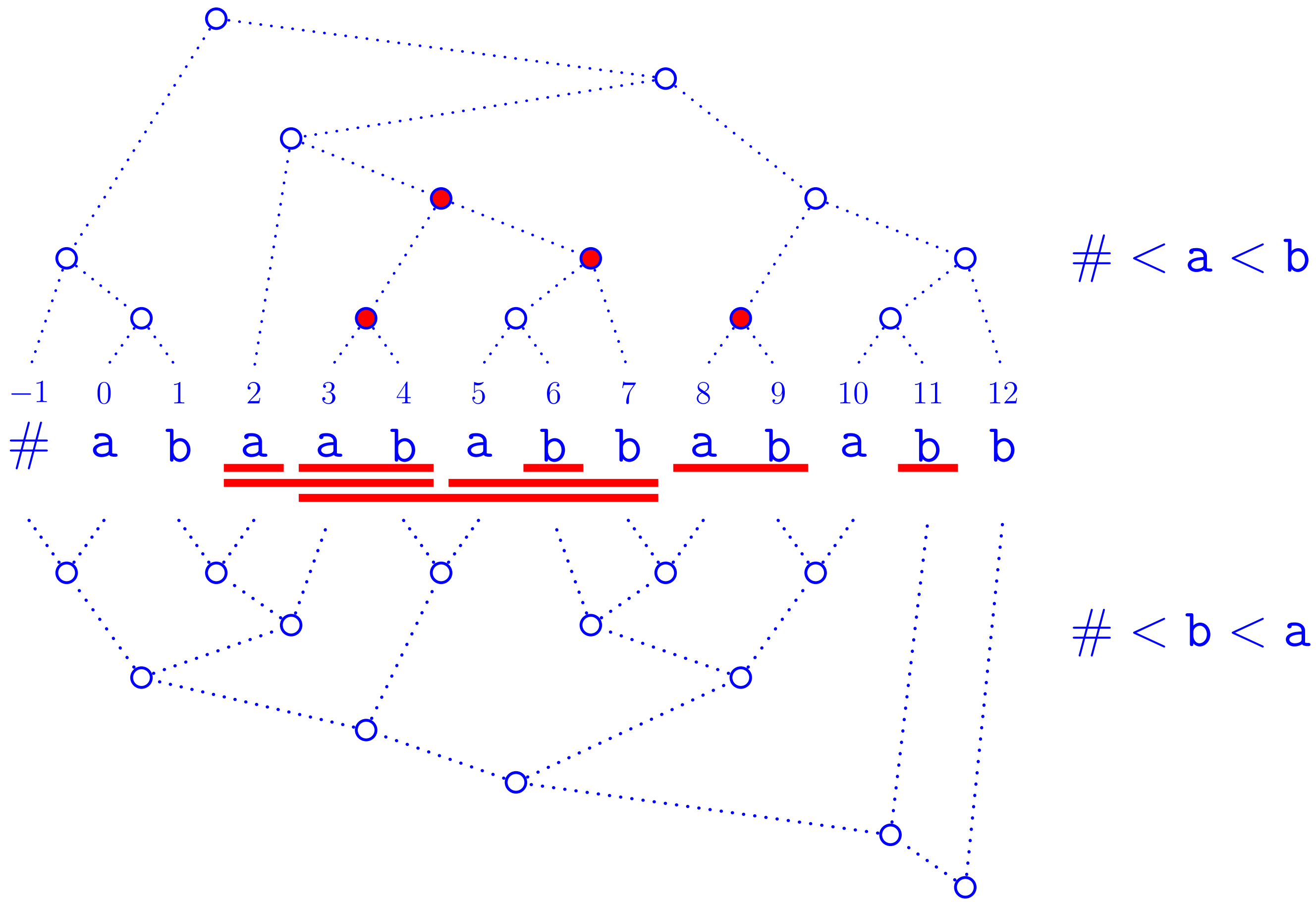
Lyndon tree

- ⋆ **Nodes are associated with Lyndon factors**
- ⋆ **. . . but some L-roots do not correspond to any internal node**



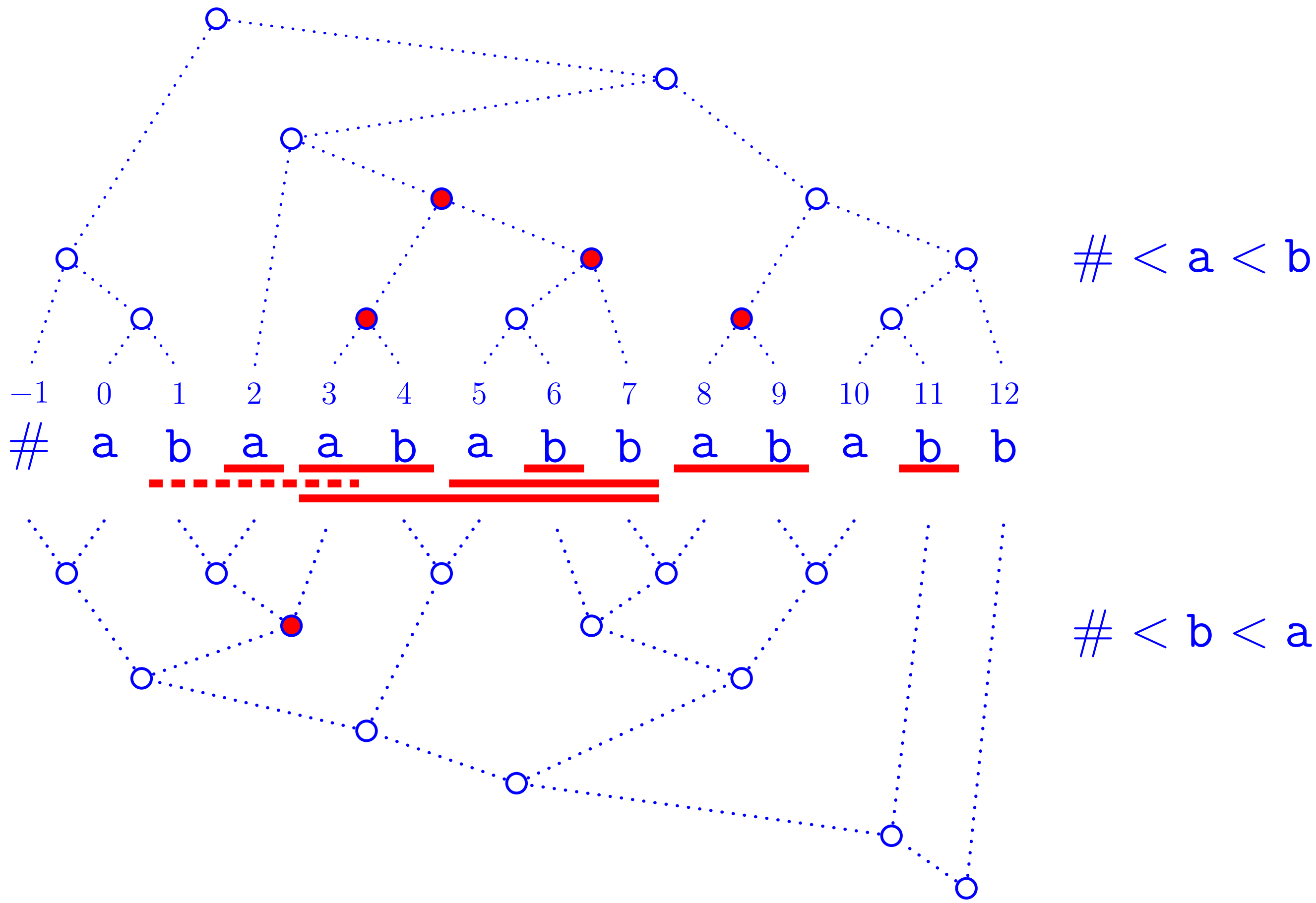
Lyndon tree

- Nodes are associated with Lyndon factors
- . . . but some L-roots do not correspond to any internal node



Lyndon tree

- Nodes are associated with Lyndon factors
- . . . but some L-roots do not correspond to any internal node



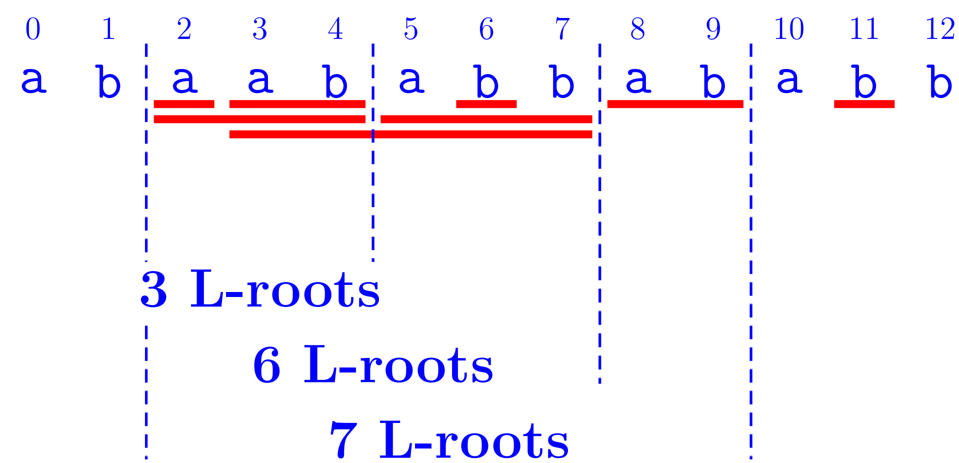
Use of Lyndon trees

- **Lyndon trees show: no more than $2.5n$ runs**
  (n internal nodes in each tree and no more than $0.5n$ runs of period 1)
- **Theorem 9 ([Bannai et al., 2014])**
  No more than $1.5\,n$ runs in a string of length n.
- **On integer alphabet:**
  - **Lyndon tree constr. in linear time**
    (Lyndon tree = Cartesian tree of ISA)
  - **Constant time to check if an internal node corresponds to an L-root** (with LCE and RMQ)
- **Lyndon trees provide a new algorithm for locating runs**

New conjecture

Conjecture 2 ***Each string interval contains no more Lyndon roots than its length.***

- **L-roots in** abaababbababb



- **Properties of full words? Their lengths?**
  factors accepting as many L-roots as their length

Computing repetitions in strings

- ⋆ **Computing runs**
  $O(n \log n)$ **optimal time in the** $\{=, \neq\}$**-comparison model**
  [C., Kociumaka, Rytter, Toopsuwan, Tyczyński, Waleń, 2012]
  $O(n \log a)$ **time [Kolpakov, Kucherov, 1999]**
  $O(n)$ **on int. alph. [C., Ilie, 2008], [Bannai et al., 2014]**
- ⋆ **Computing local periods**
  $O(n \log n)$ **optimal time in the** $\{=, \neq\}$**-comparison model**
  $O(n \log a)$ **time**
  [Duval, Kolpakov, Kucherov, Lecroq, Lefebvre, 2004]
- ⋆ **Computing maximal-exponent factors**
  $O(n \log a)$ **time [Badkobeh, C., Toopsuwan, 2012]**

Local periods

- $|w|$ **is a local period of** uv **at position** $|u|$ **if** $w \neq \varepsilon$ **and:**
  - **of** u **and** w **one is a suffix of the other**
  - **of** v **and** w **one is a prefix of the other**

  $\mathrm{LP}(|u|) =$ **smallest local period**

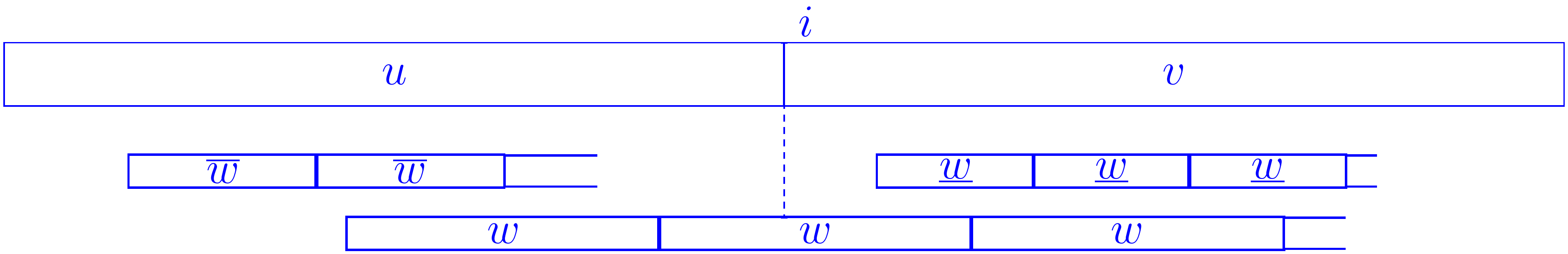
```
a b a b|b a          a|b a b b a           a b a|b b a
      b|b          b a|b a             b b a b a|b b a b a
```

- **Local periods of** `ababba`

position i	0	1	2	3	4	5	6
$y[i]$	a	b	a	b	b	a	
$\mathrm{LP}[i]$	1	2	2	5	1	3	1

Divide and conquer

⋆ **String** $y = uv$



⋆ $\mathrm{LP}[i]$ = **local period at position** i**:**

– **initialised with the (global) period of** y **. . .**

– **. . . and at the ends of** y

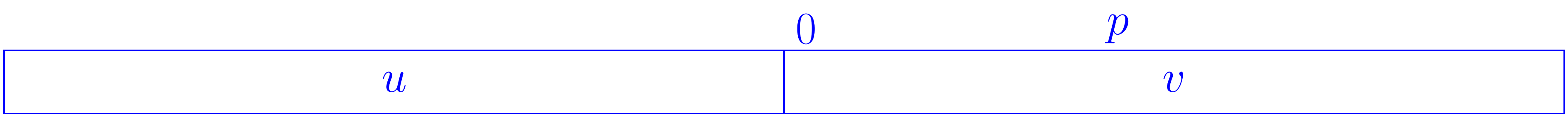
– **updated each time** i **is in the middle of a run**

⋆ **Attention: avoid non-primitive roots and several detections of the same run**

⋆ $O(n \log n)$ **occurrences of primitively-rooted squares** $\Longrightarrow O(n \log n)$ **time**

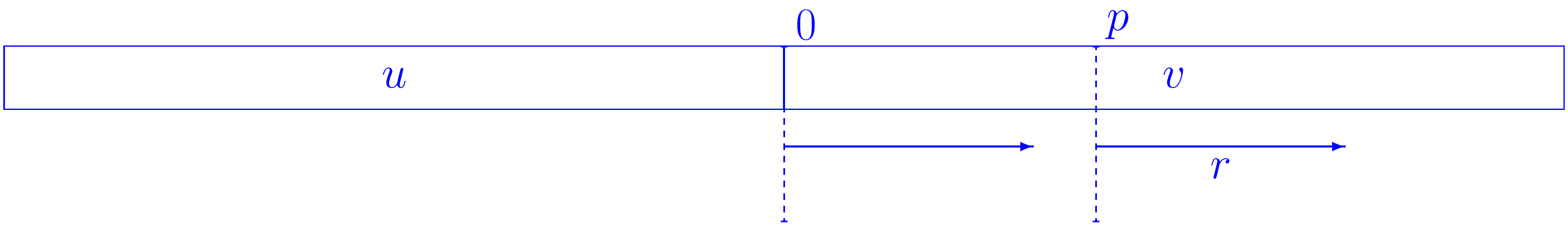
Run in a product

- Computing runs having a full period in v,
  for each period length p



Run in a product

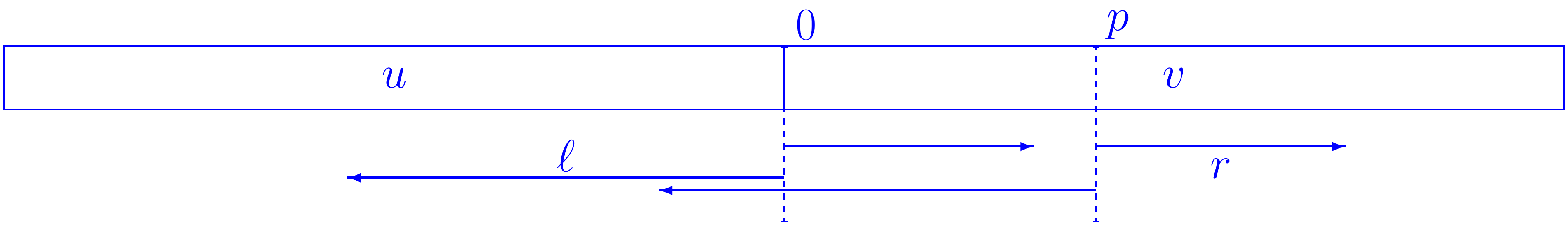
- **Computing runs having a full period in v, for each period length p**



- **Maximal length r of common prefixes between v and $v[p\,.\,.\,|v|-1]$:** $\mathrm{Prefixes}_v[p]$

Run in a product

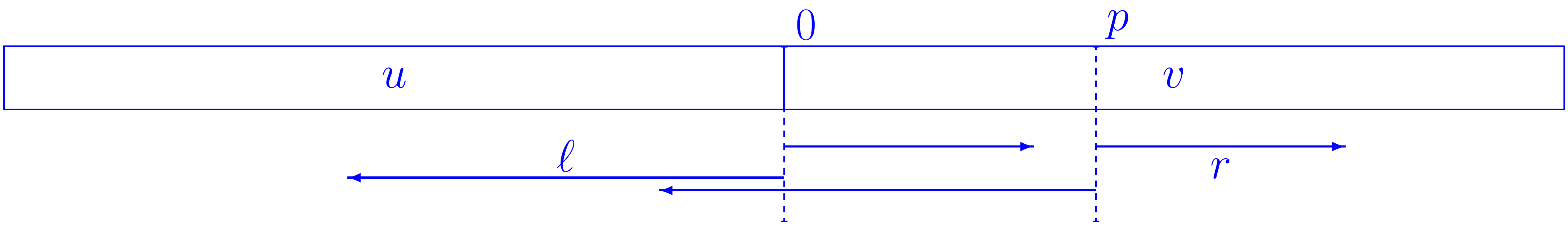
- ⋆ **Computing runs having a full period in v, for each period length p**



- ⋆ **Maximal length r of common prefixes between v and $v[p\,..\,|v|-1]$:** $\mathrm{Prefixes}_v[p]$
- ⋆ **Maximal length ℓ of common suffixes between u and $uv[0\,..\,p-1]$: deduced from** $\mathrm{Prefixes}_{\tilde{u}\#\tilde{v}\tilde{u}}$

Run in a product

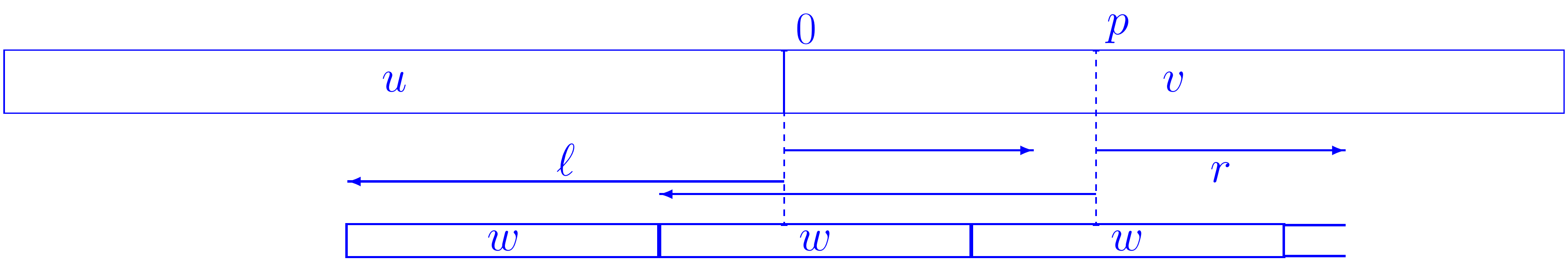
- ⋆ **Computing runs having a full period in v, for each period length p**



- ⋆ **Maximal length r of common prefixes between v and $v[p\,.\,.\,|v|-1]$:** $\mathrm{Prefixes}_v[p]$
- ⋆ **Maximal length ℓ of common suffixes between u and $uv[0\,.\,.\,p-1]$: deduced from** $\mathrm{Prefixes}_{\tilde{u}\#\tilde{v}\tilde{u}}$
- ⋆ **Linear-time precomputation of the two** Prefixes **tables**

Run in a product

- **Computing runs having a full period in v, for each period length p**



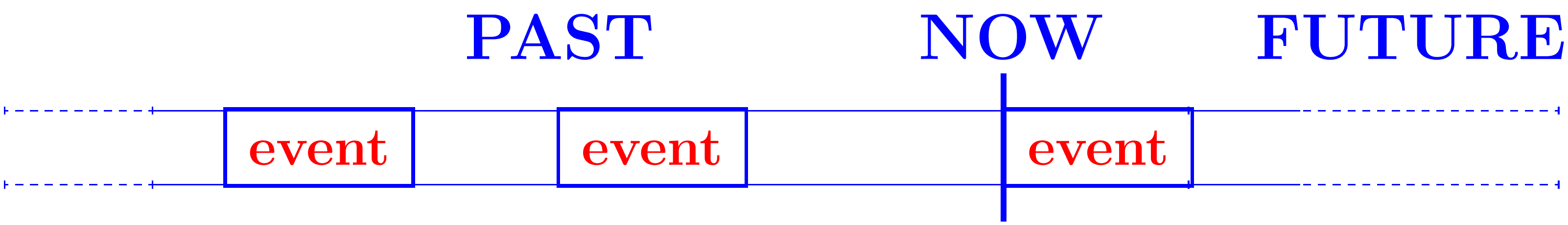
- **Maximal length r of common prefixes between v and $v[p\,..\,|v|-1]$:** $\text{Prefixes}_v[p]$
- **Maximal length ℓ of common suffixes between u and $uv[0\,..\,p-1]$: deduced from** $\text{Prefixes}_{\tilde{u}\#\tilde{v}\tilde{u}}$
- **Linear-time precomputation of the two** Prefixes **tables**
- **Run of period p if $\ell + r \geq p$**
  Constant time for each p and total linear time

Computing runs

- In $O(n \log n)$ time with previous technique
- Optimal in the $\{=, \neq\}$-model
  optimality is a consequence of [Main, Lorentz, 1979]
- In $O(n \log a)$ time based on
  - modified Main's algorithm
  - f-factorisation (kind of Ziv-Lempel factorisation)
  - linear upper bound on the number of runs

  [Kolpakov, Kucherov, 1998]
- f-factorisation is the bottleneck
- Linear-time solution on integer alphabet
  [C., Ilie, 2007]

Remembering the Past



“***Who so neglects learning in his youth, loses the past and is dead for the future.***”

Euripides (484 BC - 406 BC)

f-factorisation

- **Phrase = longest factor occurring before (LPF)**
- **Example of** $y =$ `abaababababaaababb`

a b a a b a b a b a a a b a b b

a b a a b a b a b a a a b a b b

f-factorisation

- **Phrase = longest factor occurring before (LPF)**
- **Example of** $y =$ `abaababababaaababb`

`a b a a b a b a b a a a b a b b`

`a b a a b a b a b a a a b a b b`

- **LZ77 [Ziv, Lempel, 1977]**
  phrases are carefully encoded as

$$(\textit{\textbf{distance to previous position}}, \textit{\textbf{length}})$$

- **Very efficient: many variants implemented in compress, gzip, PKzip, rzm, lzturbo, etc.**
- **Computation in time** $O(n \log a)$ **(**$a =$ alphabet size**)**

Storing the Past: LPF table

⋆ Longest Previous Factor table

position i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$y[i]$	a	b	a	a	b	a	b	a	b	b	a	b	b	b
LPF$[i]$	0	0	1	3	2	4	3	2	1	4	3	2	2	1

7

a b a a b a b a b b a b b b

5

a b a a b a b a b b a b b b

Storing the Past: LPF table

- ⋆ **Longest Previous Factor table**

position i	0	1	2	3	4	5	6	7	8	9	10	11	12	13
$y[i]$	a	b	a	a	b	a	b	a	b	b	a	b	b	b
$\text{LPF}[i]$	0	0	1	3	2	4	3	2	1	4	3	2	2	1

7

a b a a b a b a b b a b b b

5

a b a a b a b a b b a b b b

- ⋆ **Useful for optimising compression, computing repetitions, etc.**
- ⋆ **Same notion in [McCreight, 1976]**
and [Franek, Holub, Smyth, Xiao, 2003]
- ⋆ **Linear-time computation with a Suffix Array**

LPF from Suffix Array

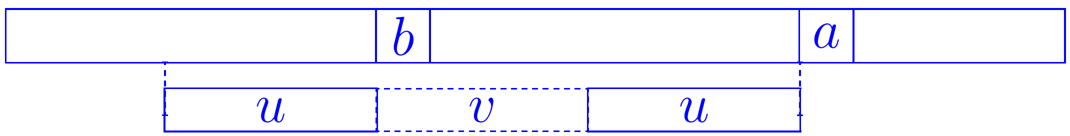
- ⋆ **Integer alphabet:** sorting letters can be done in linear time
- ⋆ **Suffix Array construction: suffix sorting + LCP**
  - **Linear-time suffix sorting by**
    [Kärkkäinen, Sanders, 2003], [Ko, Aluru, 2003]
    [Kim, Sim, Park, Park, 2003], [Nong, Zhang, Chan, 2009]
  - **Linear-time computation of LCP table by**
    [Kasai, Lee, Arimura, Arikawa, Park, 2001]

LPF from Suffix Array

- **Integer alphabet:** sorting letters can be done in linear time
- **Suffix Array construction: suffix sorting + LCP**
  - **Linear-time suffix sorting by** **[Kärkkäinen, Sanders, 2003], [Ko, Aluru, 2003] [Kim, Sim, Park, Park, 2003], [Nong, Zhang, Chan, 2009]**
  - **Linear-time computation of LCP table by** **[Kasai, Lee, Arimura, Arikawa, Park, 2001]**
- **Computation of LPF table**
  - **total linear time + constant space**
  - **Possible fast implementation with permuted-LCP** **[Kärkkäinen, Manzini, Puglisi, 2009]**
  - **several variants (LPnF, LPrF)** **[C., Ilie, 2007], [C., Tischler, 2009], [Chairungsee, C., 2009], [C., Iliopoulos, Kubica, Rytter, Waleń, 2012], [C., Ilie, Iliopoulos, Kubica, Rytter, Waleń, 2013]**

Maximal-Exponent Factors

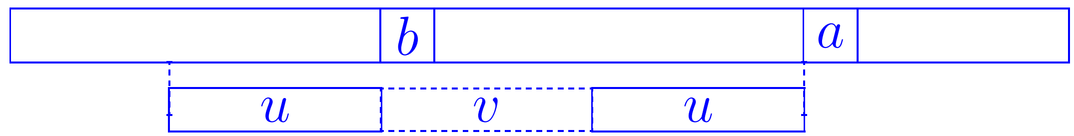
- Overlap-free string y of length n on a fixed alphabet
  Maximal exponent of all factors of y?
- MEF: maximal-exponent factor occurring in y



- Related to Maximal Pairs
  [Gusfield, 1997], [Brodal et al., 1999],
  to Return words [Vuillon, 2001],
  and to Closed words
  [Fici, 2011], [Badkobeh, Fici, Lipták, 2013]

Maximal-Exponent Factors

- Overlap-free string y of length n on a fixed alphabet
  Maximal exponent of all factors of y?
- MEF: maximal-exponent factor occurring in y

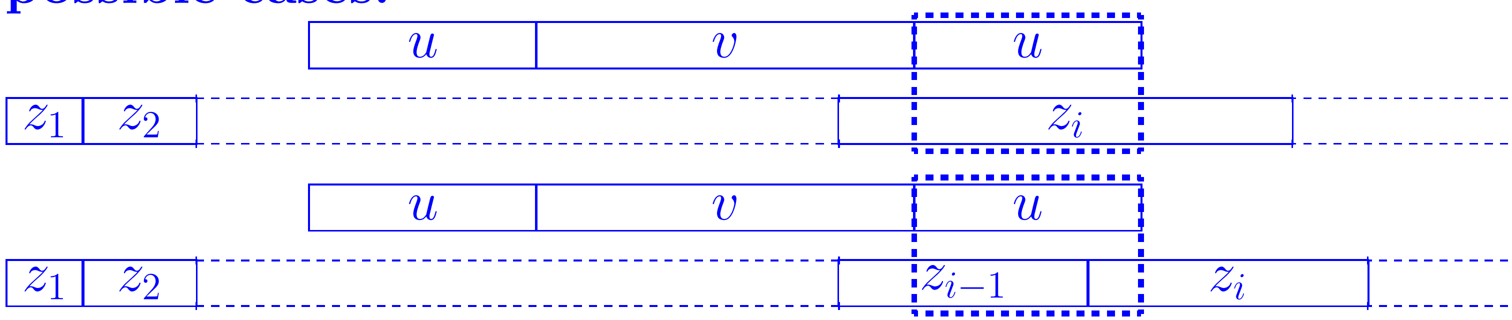


- Related to Maximal Pairs
  [Gusfield, 1997], [Brodal et al., 1999],
  to Return words [Vuillon, 2001],
  and to Closed words
  [Fici, 2011], [Badkobeh, Fici, Lipták, 2013]
- Locating MEF occurrences in an overlap-free string?

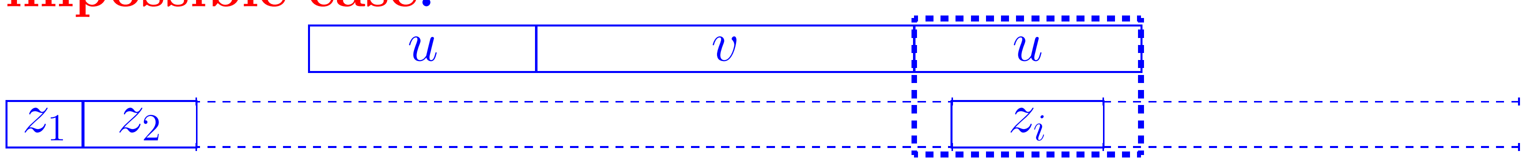
Theorem 10 ([Badkobeh, C., Toopsuwan, 2012]) *All the occurrences of maximal-exponent factors in an overlap-free string over a fixed alphabet can be listed in linear time.*

Maximal exponent of factors of a word

- ⋆ y **overlap-free** $\Longrightarrow$ **maximal exponent** ≤ 2
- ⋆ **MEF: factor of the form** uvu
- ⋆ **Naive computation in** $O(n^4)$
- ⋆ **Use of the f-factorisation of** y**:** $z_1 z_2 \ldots z_\ell$
  - **possible cases:**

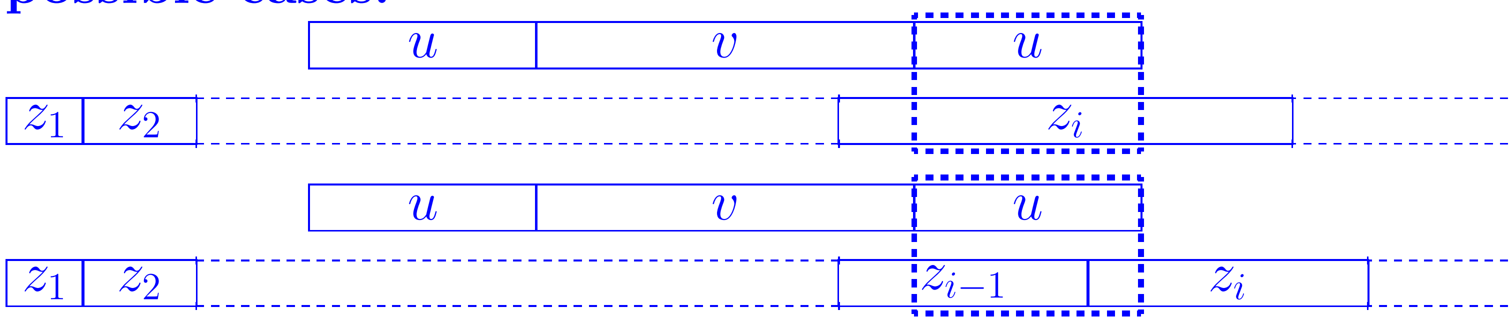


  - **impossible case:**



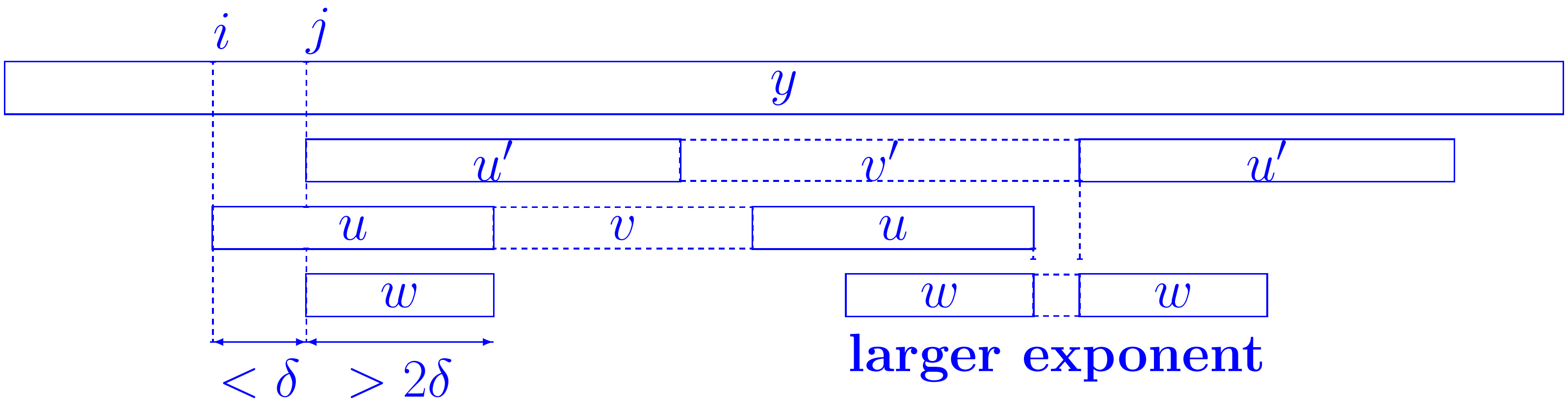
Maximal exponent of factors of a word

- ⋆ y **overlap-free** $\Longrightarrow$ **maximal exponent** ≤ 2
- ⋆ **MEF: factor of the form** uvu
- ⋆ **Naive computation in** $O(n^4)$
- ⋆ **Use of the f-factorisation of** y**:** $z_1 z_2 \dots z_\ell$
  - **possible cases:**



  - **RT**(a) $\Longrightarrow$ **search for left occurrence of** u **in a bounded context. Essential use of the Suffix Automaton of** $z_{\widetilde{i-1}} z_i$
  - **overall time:** $O(n \log a)$
    [Badkobeh, C., Toopsuwan, 2012]

Counting MEF occurrences



- δ-MEF: MEF whose border length satisfies $3\delta \le b < 4\delta$.
- then no more than one δ-MEF occ. in each δ interval
- with $\Delta = \{1/3, 2/3, 1, 3/4, (3/4)^2, \ldots\}$
  $\#$MEF-occ $\le \sum_{\delta\in\Delta} \frac{n}{\delta} = n\left(3 + \frac{3}{2} + 1 + \frac{3}{4} + \left(\frac{3}{4}\right)^2 + \ldots\right) < 8.5\,n$
- Consequence: linear computation of all MEF occurrences

Theorem 11 ***Less than*** $2.25\,n$ ***occurrences of MEFs in a string of length*** n***. There can be*** $2n/3 - \epsilon$ ***occurrences.***

Approximate runs

⋆ **Approximation:**
k = **smallest number of changes to get a consensus period**

.. x y c a g c t g c a g $\overset{\text{period}}{\overleftrightarrow{\texttt{c a g}}}$ a a g a a x y ..

Approximate runs

⋆ **Approximation:**
k = **smallest number of changes to get a consensus period**

period

`.. x y c a g c t g c a g c a g a a g a a x y ..`

`.. x y c a g c a g c a g c a g c a g c a x y ..`

Approximate runs

- **Approximation:**
  k = smallest number of changes to get a consensus period

```
                                        period
                                      <------->
.. x y   c a g   c t g   c a g   c a g   a a g   a a   x y ..
.. x y   c a g   c a g   c a g   c a g   c a g   c a   x y ..
```

- **Several other notions based on mismatches between potential periods and leading to different algorithms**
  [Sim, Iliopoulos, Park, Smyth, 1999], [Landau, Schmidt, Sokol, 2001], [Kolpakov, Kucherov, 2003]

Approximate runs

- **Approximation:**
  k = smallest number of changes to get a consensus period

$$\overbrace{\phantom{\texttt{c a g}}}^{\text{period}}$$

`.. x y  c a g  c t g  c a g  c a g  a a g  a a  x y ..`

`.. x y  c a g  c a g  c a g  c a g  c a g  c a  x y ..`

- **Several other notions based on mismatches between potential periods and leading to different algorithms [Sim, Iliopoulos, Park, Smyth, 1999], [Landau, Schmidt, Sokol, 2001], [Kolpakov, Kucherov, 2003]**
- **k-MAR: maximal occ. of approx. runs with $\leq k$ changes**
- **Solutions in [Amit, C., Landau, 2013]:**
  using Parikh vectors: $O(n^2)$ time
  kangaroo jumps with Suffix Tree LCA queries + tuning: $O(nk^2 \log \frac{n}{k} \log k)$ time

Conclusion and open questions

- **Computing runs and local periods:**
  $O(n \log n)$ **optimal time in the** $\{=, \neq\}$**-comparison model**
  linear-time on an integer alphabet
- **Computing MEF occurrences, gapped palindromes:**
  linear-time on a fixed alphabet

Conclusion and open questions

- **Computing runs and local periods:**
  $O(n \log n)$ **optimal time in the** $\{=, \neq\}$**-comparison model**
  linear-time on an integer alphabet
- **Computing MEF occurrences, gapped palindromes:**
  linear-time on a fixed alphabet

- **Q: conjectures: number of runs, of squares. Less than** n**?**
  Q: maximal number of MEF occurrences? Less than n**?**
- **Q: computing MEF occurrences on integer alphabet?**
- **Q: faster** k**-MAR computation?**
- **Q: is** 2 **the actual threshold exponent?**
  Q: any other threshold?
  Note: no more than $\frac{1}{\epsilon} n \ln n$ **maximal periodicities of exponent more than** $1 + \epsilon$ **[Kolpakov, Kucherov, Ochem, 2010]**

Collaborators

- **On presented works**
  - **Mika Amit,** University of Haifa
  - **Golnaz Badkobeh,** University of Sheffield
  - **Supaporn Chairungsee,** Walailak University
  - **Lucian Ilie,** University of Western Ontario
  - **Costas Iliopoulos,** King's College London
  - **Tomasz Kociumaka,** Warsaw University
  - **Marcin Kubica,** Warsaw University
  - **Gad Landau,** University of Haifa
  - **Jakub Radoszewski,** Warsaw University
  - **Michaël Rao,** ENS de Lyon
  - **Wojciech Rytter,** Warsaw University
  - **German Tischler,** Sanger Institute
  - **Chalita Toopsuwan,** King's College London
  - **Wojciech Tyczyński,** Warsaw University
  - **Tomasz Waleń,** Warsaw University