Computing local periodicities in strings

Maxime Crochemore

Université Paris-Est and **King's College London**

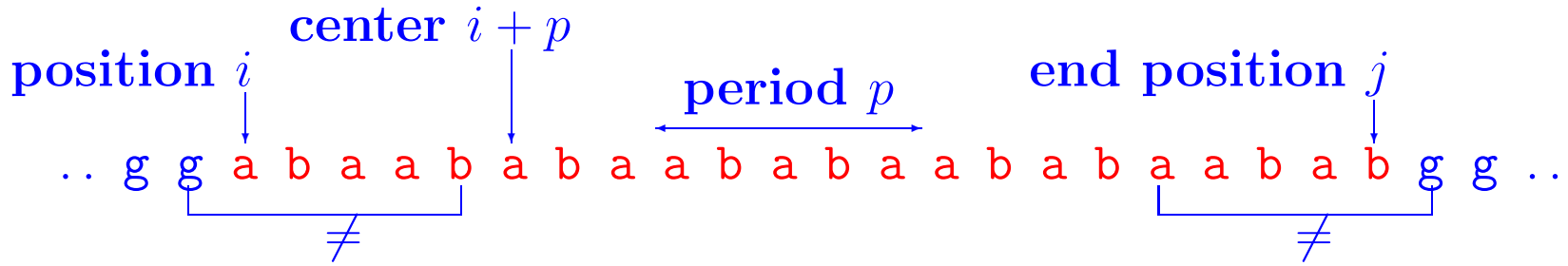
Joint work with

Lucian Ilie

University of Western Ontario

Runs in a string

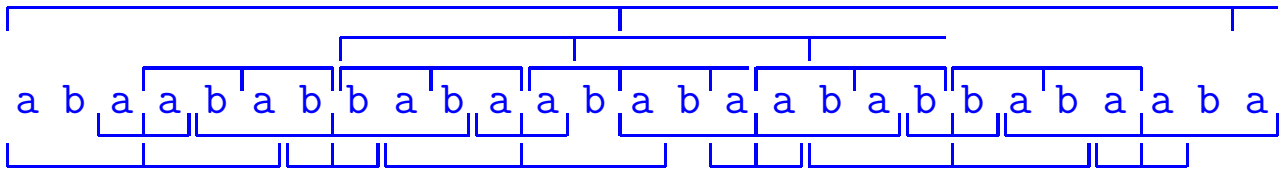
- ⋆ **Run = maximal occurrence of a periodic factor**
- ⋆ **a run of factor** $\mathtt{abaababaababaababaabab} = (\mathtt{abaab})^{22/5}$**:**



- ⋆ **A run is an interval** $[i\,..\,j]$ **of positions on** y **that satisfies:**
  - $p = \mathrm{period}(y[i\,..\,j]) \leq (|y[i\,..\,j]|)/2 = (j-i+1)/2$ **periodicity**
  - $y[i-1] \neq y[i+p-1]$ (if $y[i-1]$ defined) **non left extensib.**
  - $y[j-p+1] \neq y[j+1]$ (if $y[j+1]$ defined) **non right extensib.**

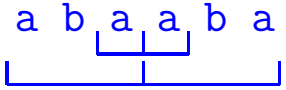
Example with many runs

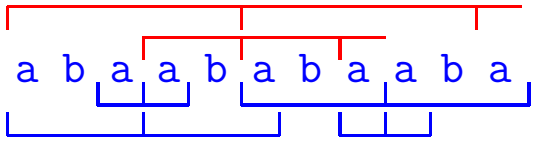
⋆ **number of runs / length** $\longrightarrow \frac{3}{1+\sqrt{5}}$

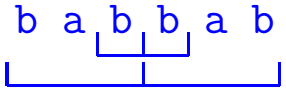


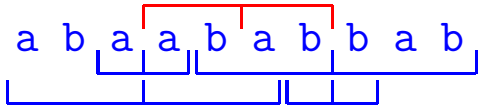
⋆ $f(\mathtt{a}) = \mathtt{abaaba},\ f(\mathtt{b}) = \mathtt{babbab}$

$uw \circ wv = uwv$ (w shortest nonempty overlap between uw and wv)

$f(\mathtt{a})$ 

$f(\mathtt{a}) \circ f(\mathtt{a})$ 

$f(\mathtt{b})$ 

$f(\mathtt{a}) \circ f(\mathtt{b})$ 

⋆ $f^1(\mathtt{a}) = \mathtt{abaaba}$

$f^2(\mathtt{a}) = \mathtt{abaaba} \circ \mathtt{babbab} \circ \mathtt{abaaba} \circ \mathtt{abaaba} \circ \mathtt{babbab} \circ \mathtt{abaaba}$

$= \mathtt{abaababbabaababaabababbabaaba}$

⋆ **[Franek, Simpson, Smyth, 2003]**

Local periodicities in strings

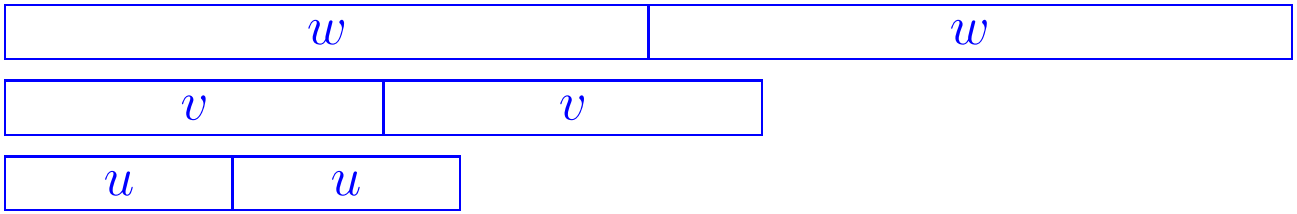
- ⋆ **Combinatorics on words**
  - Avoidability of squares
    [Thue, 1906, 1912]
  - Interaction between periods
    [Fine, Wilf, 1965], [Ilie, 2003], [Zamboni, 2003]
    see [Mignosi, Restivo, 2002] in [Lothaire, 2002]
- ⋆ **Pattern matching algorithms**
  - String Matching: borders and periods, displacement table
    [Knuth, Morris, Pratt, 1977], [Boyer, Moore, 1977]
  - Time-space optimal String Matching: local and global periods
    [Galil, Seiferas, 1981], [Crochemore, Perrin, 1989], [Rytter, 2003]
- ⋆ **Analysis of biological molecular sequences**
  - Intensive study of satellites or Simple Sequence Repeats in DNA involved e.g. in Huntington's and Kennedy's diseases ($(\mathtt{CAG})^e$, $e > 36$)
    [MacDonald, Ambrose, 1993]

Two-way string-matching

Searching for pattern $x = x[0\,..\,m-1]$ in text $y = y[0\,..\,n-1]$

- (u, v) **critical factorization of** x**:**
  $uv = x$ **and** $|u| < \text{loc-per}(u, v) = \text{period}(x)$
- **Searching**
  - **search for an occurrence of** v **by left-to-right scans**
    if mismatch: length of shift = length of scanning, else
  - **search for a preceding occurrence of** u **(right-to-left scan)**
    length of shift = period of x
- **Preprocessing**
  - **compute factorization** (u, v)
  - **compute the (smallest) period of** x
- **Algorithmic complexity**
  - **total linear time; searching in less than** $2n$ **comparisons**
  - **only constant additional memory space required**

Square prefixes



Lemma 1 (Three square prefixes)
If u^2 ***prefix of*** v^2, v^2 ***prefix of*** w^2, ***and*** u ***primitive then*** $|u| + |v| \leq |w|$.

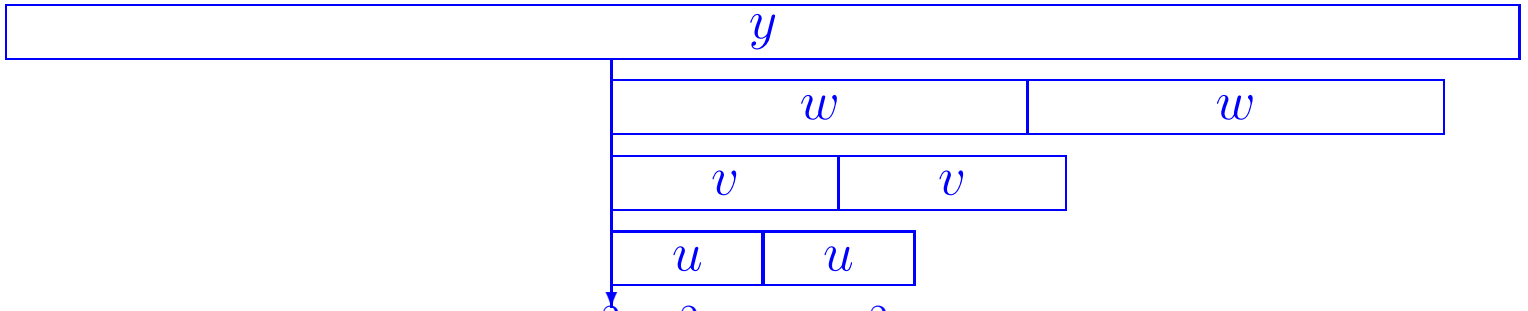
[Crochemore, Rytter, 1995]

```
10    a a b a a b a a a b a a b a a b a a a b
 7    a a b a a b a a a b a a b a
 3    a a b a a b
```

Proposition 1
No more than $\log_\Phi |y|$ ***prefixes of*** y ***are primitively-rooted squares.***

How many squares in a word?

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



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

- ⋆ **Direct proofs [Hickerson, 2004], [Ilie, 2005]**
- ⋆ **Best bound:** $2n - \Theta(\log n)$ **[Ilie, 2005]**
- ⋆ **Computation in time** $O(n \log a)$ **[Gusfield, Stoye, 1999]**

Proposition 3 ([Crochemore, 1981], [Gusfield, Stoye, 1999])
Maximum number of occurrences of primitively-rooted squares : $cn \log n$***. Attained by Fibonacci words.***

Smallest Local Period

- i **position on** y
- **Local square at** i **=**
  shortest nonempty square centered in i
- **Word of (global) period** 5**:**

	0	1	2	3	4	5	6	7
y		a	b	a	b	a	a	b
loc-per	**1**	**2**	**2**	**2**	**5**	**1**	**3**	**1**

- **Note:** $\text{loc-per}[i] \leq \text{period}(y)$
  $\text{loc-per}[i] = \text{period}(y)$ **iff** i **defines a critical factorization of** y
- **Computation of all local periods in time** $O(n \log a)$
  [Duval, Kolpakov, Kucherov, Lecroq, Lefebvre, 2003]

All powers

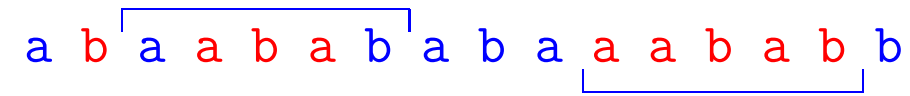
- **Finding all powers efficiently**
  - **Problem: too many occurrences**
  - **Solution: select some, encode them in a compact form**
- **All primitively-rooted right-maximal integer:** $O(n \log n)$ **time [Crochemore, 1981]**
- **All primitively-rooted right-maximal:** $O(n \log n)$ **time [Apostolico, Preparata, 1983]**
- **All primitively-rooted maximal:** $O(n \log n)$ **time [Main, Lorentz, 1985]**
- **All leftmost maximal:** $O(n \log a)$ **time [Main, 1989]**
- **All runs in Fibonacci strings:** $O(n)$ **time [Iliopoulos, Moore, Smyth, 1997]**

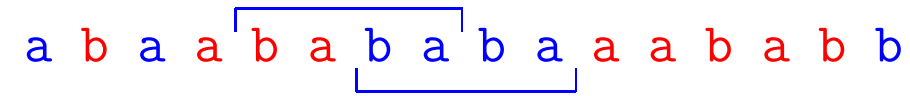
Computing runs

- All powers are encoded in runs
- Computation in $O(n \log a)$ time
  [Kolpakov, Kucherov, 1998]
- .. based on
  - modified Main's algorithm
  - f-factorization (similar to Lempel-Ziv factorization)
  - linear upper bound on the number of runs
- Explicit bounds:
  $5n$ [Rytter, 2006]
  $1.6n$ [Crochemore, Ilie, 2006]

Factorization

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





- **Computation with the suffix tree of** y**:** $O(n \log a)$ **time**
- **.. however possible in linear time:**
  - with linear time suffix tree construction on integer alphabet **[Farach, 1997]**
  - if branching is avoided (branching time is $O(\log a)$)
- **.. but simple linear-time with the suffix array of** y

Suffix Array

- ⋆ **POS: list of positions of sorted suffixes**
- ⋆ **LCP: maximal lengths of common prefixes btwn consec suffixes**

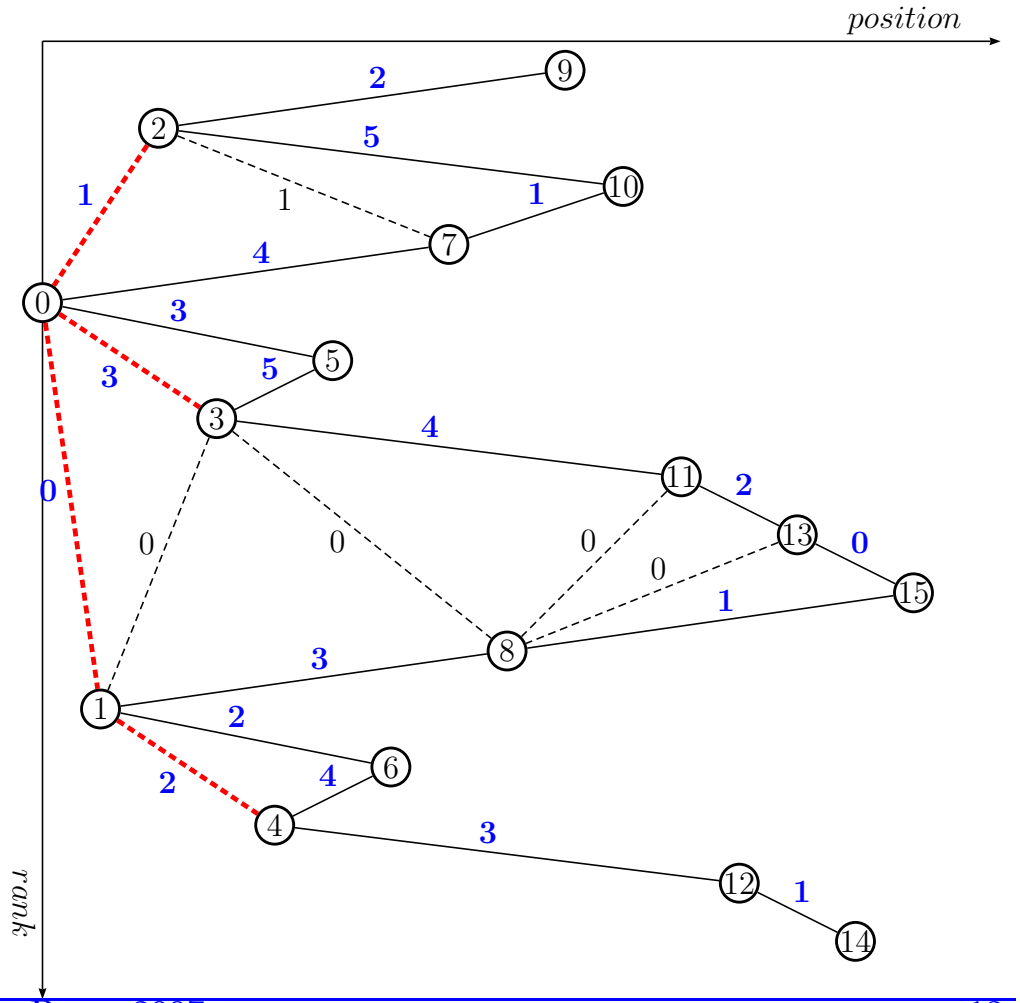
i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$y[i]$	a	b	a	a	b	a	b	a	b	a	a	a	b	a	b	b

rank	POS	LCP	suffix
0	**9**		aaababb
		2	
1	**2**		aababababaaababb
		5	
2	**10**		aababb
		1	
3	**7**		abaaababb
		4	
4	**0**		abaababababaaababb
		3	
5	**5**		ababaaababb
		5	
6	**3**		abababaaababb
		4	
7	**11**		ababb
		2	
8	**13**		abb
		0	
9	**15**		b
		1	
10	**8**		baaababb
		3	
11	**1**		baababababaaababb
		2	
12	**6**		babaaababb
		4	
13	**4**		bababaaababb
		3	
14	**12**		babb
		1	
15	**14**		bb

LCP and LPF

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$y[i]$	a	b	a	a	b	a	b	a	b	a	a	a	b	a	b	b

rank	POS	LCP	LPF
0	**9**	**2**	**2**
1	**2**	**5**	**1**
2	**10**	**1**	**5**
3	**7**	**4**	**4**
4	**0**	**3**	**0**
5	**5**	**5**	**5**
6	**3**	**4**	**3**
7	**11**	**2**	**4**
8	**13**	**0**	**2**
9	**15**	**1**	**1**
10	**8**	**3**	**3**
11	**1**	**2**	**0**
12	**6**	**4**	**4**
13	**4**	**3**	**2**
14	**12**	**1**	**3**
15	**14**		**1**

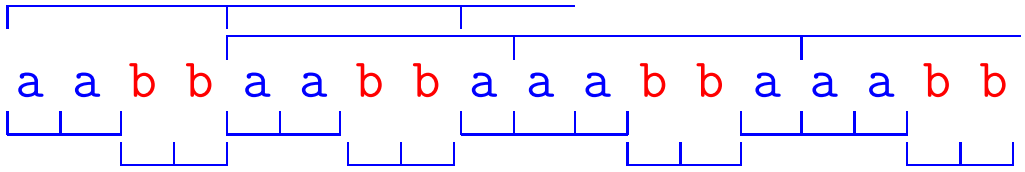


Computing all runs

- ⋆ **Hypothesis on integer alphabet: Alph$(y) \subseteq [0\,..\,|y|^c]$**
  (sorting letters can be done in linear time)
- ⋆ **Linear-time suffix sorting by [Kärkkäinen, Sanders, 2003] [Kim, Sim, Park, Park, 2003], [Ko, Aluru, 2003]**
- ⋆ **Linear-time computation of LCPs by [Kasai, Lee, Arimura, Arikawa, Park, 2001]**
- ⋆ **.. no more $O(\log a)$ branching time, although a can be n^c**
- ⋆ **Conclusion on integer alphabet:**
  - **Linear-time computation of Longest Previous Factors**
    (no more than $n-2$ additional edges)
  - **Linear-time Lempel-Ziv factorization**
  - **Linear-time computation of all runs**
    (linear number of them)

Number of runs

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



- ⋆ **Conjectures**
  - $\leq n$ **[Kolpakov, Kucherov, 1998]**
  - $\leq \frac{3}{1+\sqrt{5}}n = 0.927..n$ **[Franek, Simpson, Smyth, 2003]**
  - **..** $g^k(\mathtt{a})$
- ⋆ **Known bounds**
  - $5n$ **[Rytter, 2006]**
  - $3.48n$ **[Puglisi, Simpson, Smyth, 2006]**
  - $3.44n$ **[Rytter, 2006]**
  - $1.6n$ **[Crochemore, Ilie, 2006]**

Proof ideas

- ⋆ **Short-period runs (microruns):**
  no more than n runs of period ≤ 9 (hand-checked)
- ⋆ **Long-period runs:**
  - **count the δ-runs: runs of period between 2δ and 3δ for**

$$\delta = \frac{10}{2},\ \frac{10}{2}\frac{3}{2},\ \frac{10}{2}\left(\frac{3}{2}\right)^2,\ \ldots$$

  - **either two centers of δ-runs are at distance $\geq \delta$**
  - **.. or they are close but push away other centers of δ-runs**
  - **on the average: 1 center per interval of length δ**
    $\Longrightarrow$ no more than n/δ δ-runs
- ⋆ **Total number of runs:**

$$< \ n + \left(\frac{2}{10}\sum_{i=0}^{\infty}\left(\frac{2}{3}\right)^i\right)n \ = \ 1.6n$$

Some open questions

- ⋆ **Algorithmics**
  - **Any simple run detection algorithm derived from the proof?**
  - **Any simple algorithm to compute LCP, LPF, and the like?**
  - **How many additional dotted edges to get constant-time LCP queries?**
- ⋆ **Combinatorics**
  - **Conjecture: runs$(n) < n$ does an extension of the short-period-run argument prove it?**
  - **Is runs(n) attained by cube-free word?**
  - **What is the optimal bound on the number of runs? $\frac{3}{1+\sqrt{5}}n$?**