Computing Abelian periods in words

G. Fici, T. Lecroq, **A. Lefebvre** and É. Prieur-Gaston
gabriele.fici@unice.fr
{thierry.lecroq,arnaud.lefebvre,elise.prieur}@univ-rouen.fr

I3S, CNRS/Université de Nice-Sophia Antipolis, France
University of Rouen, LITIS EA 4108, 76821 Mont-Saint-Aignan Cedex, France

Prague Stringology Conference 2011
31st August 2011

Outline

Outline

Parikh vector

Definition

$\mathcal{P}(w)$: Parikh vector of a word w, that enumerates the cardinality of each letter of the alphabet in w

Definition

$\mathcal{P}_w(i,k)$: Parikh vector of the factor of w starting at position i and of length k

Example with $w =$ abaababa

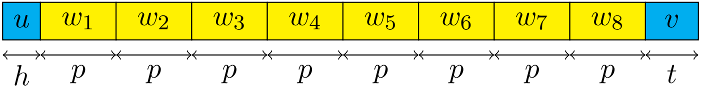
$\mathcal{P}(\textbf{abaababa}) = (5, 3)$
$\mathcal{P}_w(2,3) = (2,1)$ ($w =$ **abaababa**)

Abelian periods

Definition

A word w has an Abelian period (h, p) if

- $w = uw_1 \cdots w_k v$
- $|u| = h$, $|w_i| = p$ for all i
- $\mathcal{P}(u) \subset \mathcal{P}(w_1) = \cdots = \mathcal{P}(w_k) \supset \mathcal{P}(v)$



u (or h) is called the head and v (or t) is called the tail

weak repetitions or *Abelian powers* when $u, v = \varepsilon$ and $k \geq 2$

Abelian periods

Example

$(2, 3)$ is an Abelian period of **ab** $\cdot$ **aab** $\cdot$ **aba** with $\mathcal{P} = (2, 1)$

$(1, 2)$, $(0, 3)$, $(2, 3)$, $(1, 4)$, $(2, 4)$, $(3, 4)$, $(0, 5)$, $(1, 5)$, $(2, 5)$, $(3, 5)$, $(0, 6)$, $(1, 6)$, $(2, 6)$, $(0, 7)$, $(1, 7)$ and $(0, 8)$ are all the Abelian periods of **abaababa**

Periods

Classical: well studied

Abelian: properties but only one algorithm

Maximal number of Abelian periods

Lemma

The maximal number of Abelian periods of a word of length n is $O(n^2)$.

Proof

a^n has

- n periods with head length 0
- $n-2$ periods with head length 1
- $\vdots$
- 0 or 1 periods with head length $\lfloor n/2 \rfloor$ depending on the parity of n

Outline

Brute-force algorithm in $O(n^3 \times \sigma)$

```
BruteForce(w, n)
for all possible head h do
    for all possible p > h such that p + h ⩽ n do
        if (h, p) is an Abelian period of w then
            Output(h, p)
```

$select$ array

The $select$ array is defined as follows: $select[a, i]$ is equal to the i-th position of letter a in w. For example: $w =$agcgatacagatctagcgact

	1	2	3	4	5	6	7
'a'	1						
'c'							
'g'							
't'							

select array

The *select* array is defined as follows: $select[a, i]$ is equal to the i-th position of letter a in w. For example: $w =$agcgatacagatctagcgact

	1	2	3	4	5	6	7
'a'	1						
'c'							
'g'	2						
't'							

select array

The *select* array is defined as follows: $select[a, i]$ is equal to the i-th position of letter a in w. For example: $w =$agcgatacagatctagcgact

	1	2	3	4	5	6	7
'a'	1						
'c'	3						
'g'	2	4					
't'							

select array

The *select* array is defined as follows: $select[a, i]$ is equal to the i-th position of letter a in w. For example: $w =$agcgatacagatctagcgact

	1	2	3	4	5	6	7
'a'	1	5	7	9	11	15	19
'c'	3	8	13	17	20		
'g'	2	4	10	16	18		
't'	6	12	14	21			

select function: $O(n^2) \to O(n+\sigma)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
w	a	g	c	g	a	t	a	c	a	g	a	t	c	t	a	g	c	g	a	c	t

	'a'	'c'	'g'	't'
ind	1	2	3	4

	1	2	3	4	5
C	1	8	13	18	22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4	10	16	18	6	12	14	21

$$\forall a \in \Sigma, i \leqslant |w|_a, select_a(w,i) = S[C[ind[a]] + i - 1]$$

The second 'c' in w appears at index

$select_{\text{c}}(w,2) = S[C[ind[\text{c}]] + 2 - 1] = 8$

$select$ function: $O(n^2) \rightarrow O(n+\sigma)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
w	a	g	c	g	a	t	a	c	a	g	a	t	c	t	a	g	c	g	a	c	t

	'a'	'c'	'g'	't'
ind	1	2	3	4

	1	2	3	4	5
C	1	8	13	18	22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4	10	16	18	6	12	14	21

$$\forall a \in \Sigma, i \leqslant |w|_a, select_a(w,i) = S[C[ind[a]] + i - 1]$$

The second 'c' in w appears at index

$select_{\mathtt{c}}(w,2) = S[C[ind[\mathtt{c}]] + 2 - 1] = 8$

$select$ function: $O(n^2) \rightarrow O(n+\sigma)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
w	a	g	c	g	a	t	a	c	a	g	a	t	c	t	a	g	c	g	a	c	t

	'a'	'c'	'g'	't'
ind	1	2	3	4

	1	2	3	4	5
C	1	8	13	18	22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4	10	16	18	6	12	14	21

$$\forall a \in \Sigma, i \leqslant |w|_a, select_a(w,i) = S[C[ind[a]] + i - 1]$$

The second 'c' in w appears at index

$select_{\texttt{c}}(w,2) = S[C[ind[\texttt{c}]] + 2 - 1] = 8$

select function: $O(n^2) \rightarrow O(n+\sigma)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
w	a	g	c	g	a	t	a	c	a	g	a	t	c	t	a	g	c	g	a	c	t

	'a'	'c'	'g'	't'
ind	1	2	3	4

	1	2	3	4	5
C	1	8	13	18	22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4	10	16	18	6	12	14	21

$$\forall a \in \Sigma, i \leqslant |w|_a, select_a(w,i) = S[C[ind[a]] + i - 1]$$

The second 'c' in w appears at index

$select_{\mathsf{c}}(w,2) = S[C[ind[\mathsf{c}]] + 2 - 1] = 8$

$select$ function: $O(n^2) \to O(n+\sigma)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
w	a	g	c	g	a	t	a	c	a	g	a	t	c	t	a	g	c	g	a	c	t

	'a'	'c'	'g'	't'
ind	1	2	3	4

	1	2	3	4	5
C	1	8	13	18	22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4	10	16	18	6	12	14	21

$$\forall a \in \Sigma, i \leqslant |w|_a, select_a(w,i) = S[C[ind[a]] + i - 1]$$

The second 'c' in w appears at index

$select_{\mathsf{c}}(w,2) = S[C[ind[\mathsf{c}]] + 2 - 1] = 8$

$select$ function: $O(n^2) \to O(n+\sigma)$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
w	a	g	c	g	a	t	a	c	a	g	a	t	c	t	a	g	c	g	a	c	t

	'a'	'c'	'g'	't'
ind	1	2	3	4

	1	2	3	4	5
C	1	8	13	18	22

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4	10	16	18	6	12	14	21

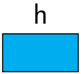
$$\forall a \in \Sigma, i \leqslant |w|_a, select_a(w,i) = S[C[ind[a]] + i - 1]$$

The second 'c' in w appears at index

$select_{\mathsf{c}}(w,2) = S[C[ind[\mathsf{c}]] + 2 - 1] = 8$

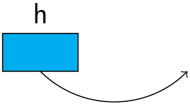
Improvements

Given h



Improvements

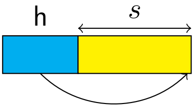
Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



Using the $select$ function, one can find the minimal s such that $\mathcal{P}_w(1, h) \subset \mathcal{P}_w(h + 1, s)$.

Improvements

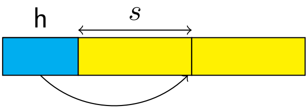
Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



Using the $select$ function, one can find the minimal s such that $\mathcal{P}_w(1, h) \subset \mathcal{P}_w(h + 1, s)$.

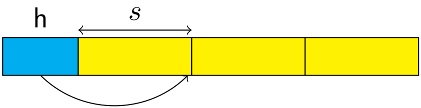
Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



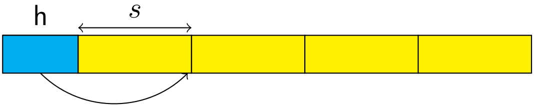
Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



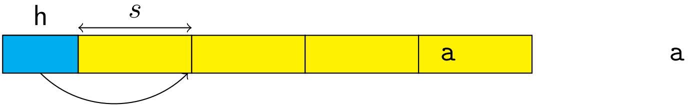
Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



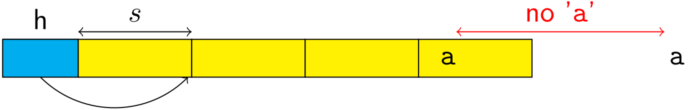
Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



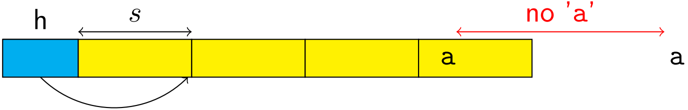
Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



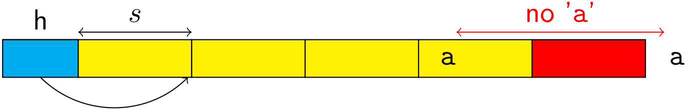
Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



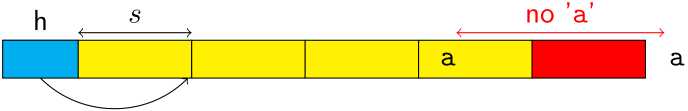
Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?

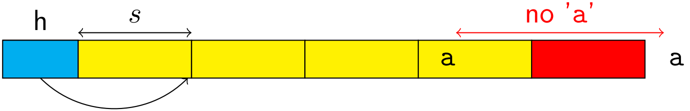


	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4
G														

$$G[h] = max\{S[i+1] - S[i] | h < i < |w|\}$$

Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?

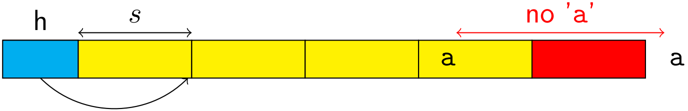


	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4
G													2	

$$G[h] = max\{S[i+1] - S[i] | h < i < |w|\}$$

Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?

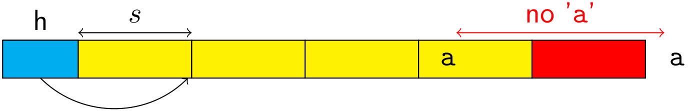


	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4
G												2	2	

$$G[h] = max\{S[i+1] - S[i] | h < i < |w|\}$$

Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?

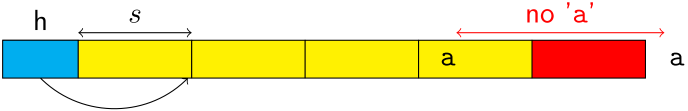


	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4
G											3	2	2	

$$G[h] = max\{S[i+1] - S[i] | h < i < |w|\}$$

Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?

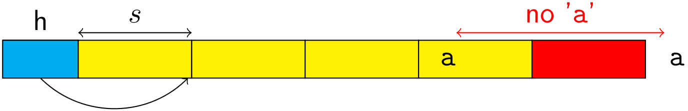


	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4
G										4	3	2	2	

$$G[h] = max\{S[i+1] - S[i] | h < i < |w|\}$$

Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?

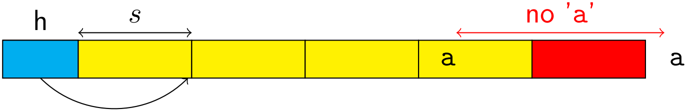


	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4
G									5	4	3	2	2	

$$G[h] = max\{S[i+1] - S[i] | h < i < |w|\}$$

Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?

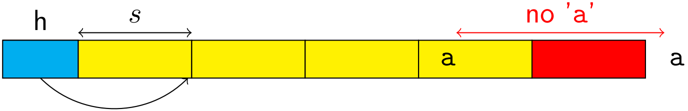


	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4
G								5	5	4	3	2	2	

$$G[h] = max\{S[i+1] - S[i] | h < i < |w|\}$$

Improvements

Given h, what is the minimal p such that (h, p) can be an Abelian period of w?



	1	2	3	4	5	6	7	8	9	10	11	12	13	14
S	1	5	7	9	11	15	19	3	8	13	17	20	2	4
G	5	5	5	5	5	5	5	5	5	4	3	2	2	

$$G[h] = max\{S[i+1] - S[i] | h < i < |w|\}$$

Find the minimal p

For a given h: The minimal p value such that (h, p) can be an Abelian period of w is the maximum between s, $G[h+1]/2$ and $h+1$.

Find the minimal p

For a given h: The minimal p value such that (h, p) can be an Abelian period of w is the maximum between s, $G[h+1]/2$ and $h+1$.

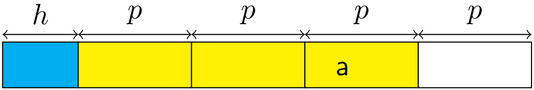
Shift function

Once (h, p) is fixed, a Shift function verifies whether it corresponds to an Abelian period or not.

Shift function

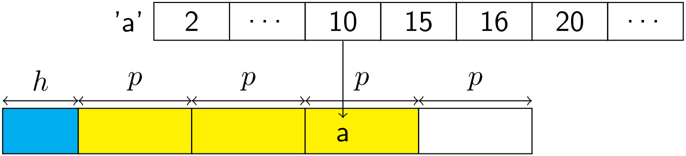
The Shift function is based on the *select* function.

'a'	2	$\cdots$	10	15	16	20	$\cdots$



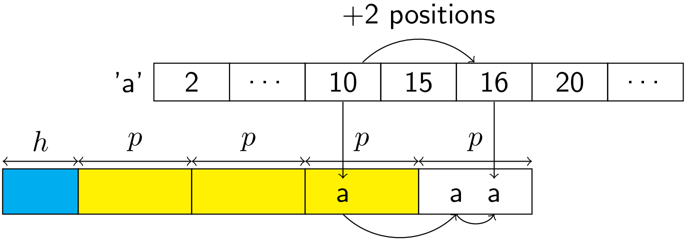
Shift function

The Shift function is based on the *select* function.



Shift function

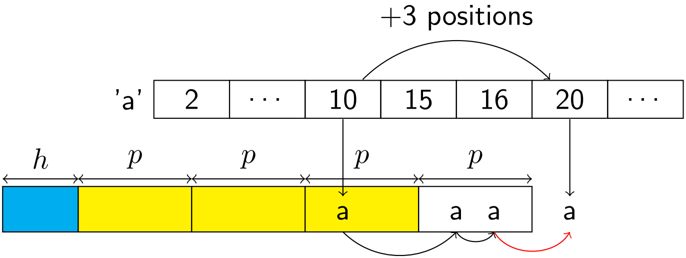
The Shift function is based on the $select$ function.



if the number of a's in the Parikh vector is 2, it's ok!

Shift function

The Shift function is based on the *select* function.



if the number of a's in the Parikh vector is 3, it's not ok!

Complexity

Time: $O(n^2 \times \sigma)$

Space: $O(n^2 + \sigma)$

Outline

On-line algorithms

Proposition

If (h, p), $h + p \leqslant |w|$, is not an Abelian period of w then (h, p) is not an Abelian period of wa for any symbol $a \in \Sigma$.

$\Longrightarrow$ skip some periods with an on-line algorithm.

Dynamic programming

When processing position i:
$T[h, p] = j$ iff $w[1\,.\,.\,j]$ is the longest prefix of $w[1\,.\,.\,i]$ having Abelian period (h, p).

Dynamic programming

When processing position i:
$T[h,p] = j$ iff $w[1..j]$ is the longest prefix of $w[1..i]$ having Abelian period (h,p).

$w =$ **abaababa**

1

a

$h \backslash p$	1	2	3	4	5	6	7	8
0	1							
1								
2								
3								

Dynamic programming

When processing position i:
$T[h,p] = j$ iff $w[1..j]$ is the longest prefix of $w[1..i]$ having Abelian period (h,p).

$w =$ **abaababa**

1	2
a	**b**

$h \backslash p$	1	2	3	4	5	6	7	8
0	1	2						
1								
2								
3								

Dynamic programming

When processing position i:
$T[h,p] = j$ iff $w[1\,..\,j]$ is the longest prefix of $w[1\,..\,i]$ having Abelian period (h,p).

$w =$ **abaababa**

1	2	3
a	**b**	**a**

$h \backslash p$	1	2	3	4	5	6	7	8
0	1	3	3					
1		3						
2								
3								

Dynamic programming

When processing position i:
$T[h,p] = j$ iff $w[1..j]$ is the longest prefix of $w[1..i]$ having Abelian period (h,p).

$w =$ **abaababa**

1	2	3	4
a	**b**	**a**	**a**

$h \backslash p$	1	2	3	4	5	6	7	8
0	1	3	4	4				
1		4	4					
2								
3								

Dynamic programming

When processing position i:
$T[h,p] = j$ iff $w[1..j]$ is the longest prefix of $w[1..i]$ having Abelian period (h,p).

$w =$ **abaababa**

1	2	3	4	5
a	**b**	**a**	**a**	**b**

$h \backslash p$	1	2	3	4	5	6	7	8
0	1	3	5	5	5			
1		5	5	5				
2			5					
3								

Dynamic programming

When processing position i:
$T[h,p] = j$ iff $w[1..j]$ is the longest prefix of $w[1..i]$ having Abelian period (h,p).

$w =$ **abaababa**

1	2	3	4	5	6
a	**b**	**a**	**a**	**b**	**a**

$h \backslash p$	1	2	3	4	5	6	7	8
0	1	3	6	6	6	6		
1		6	6	6	6			
2			6	6				
3								

Dynamic programming

When processing position i:
$T[h,p] = j$ iff $w[1..j]$ is the longest prefix of $w[1..i]$ having Abelian period (h,p).

$w =$ **abaababa**

1	2	3	4	5	6	7	8
a	**b**	**a**	**a**	**b**	**a**	**b**	

$h \backslash p$	1	2	3	4	5	6	7	8
0	1	3	7	6	7	7	7	
1		7	6	7	7	7		
2			7	7	7			
3				7				

Dynamic programming

When processing position i:
$T[h,p] = j$ iff $w[1..j]$ is the longest prefix of $w[1..i]$ having Abelian period (h,p).

$w =$ **abaababa**

1	2	3	4	5	6	7	8
a	**b**	**a**	**a**	**b**	**a**	**b**	**a**

$h \backslash p$	1	2	3	4	5	6	7	8
0	1	3	8	6	8	8	8	8
1		8	6	8	8	8	8	
2			8	8	8	8		
3				8	8			

Lists

When processing position i we only store couples (h, p) which are Abelian periods of $w[1..i]$.

Complexity

Both algorithms

Time: $O(n^3 \times \sigma)$

Space: $O(n^2)$

where

- n is the length of w
- σ is the size of the alphabet

Heaps

Proposition

w has s Abelian periods $(h_1, p_1) < (h_2, p_2) < \cdots < (h_s, p_s)$ such that $(|w| - h_i) \mod p_i = t > 0$ for $1 \leq i \leq s$.
If (h_1, p_1) is an Abelian period of wa for any symbol $a \in \Sigma$ then $(h_2, p_2), \ldots, (h_s, p_s)$ are also Abelian periods of wa.

Heaps

One heap will be created for each position of w.

Heaps

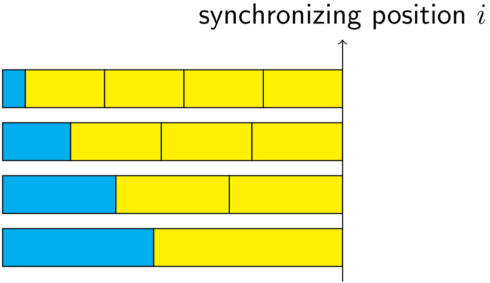
Proposition

w has s Abelian periods $(h_1, p_1) < (h_2, p_2) < \cdots < (h_s, p_s)$ such that $(|w| - h_i) \mod p_i = t > 0$ for $1 \le i \le s$.
If (h_1, p_1) is an Abelian period of wa for any symbol $a \in \Sigma$ then $(h_2, p_2), \ldots, (h_s, p_s)$ are also Abelian periods of wa.

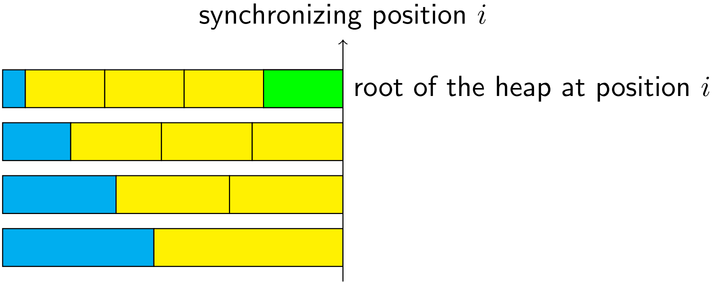
Heaps

One heap will be created for each position of w.

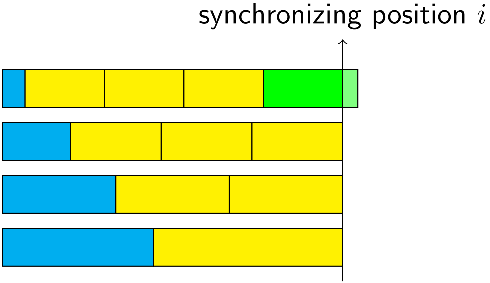
Synchronizing positions and heaps



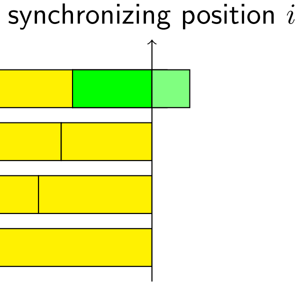
Synchronizing positions and heaps



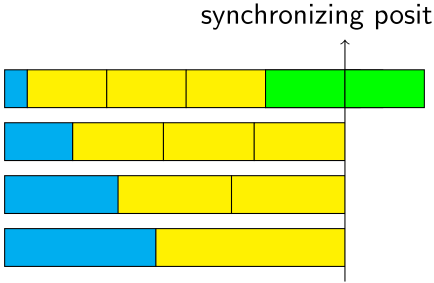
Synchronizing positions and heaps



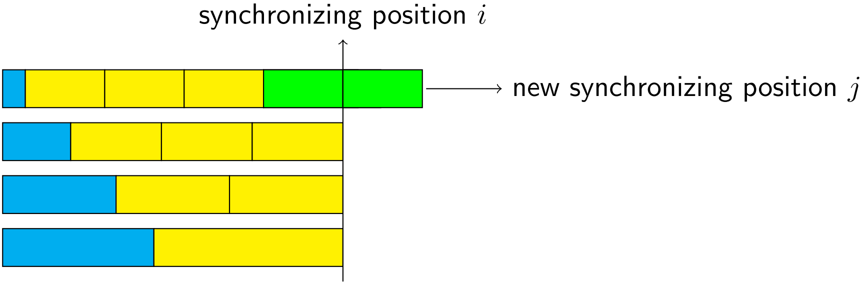
Synchronizing positions and heaps



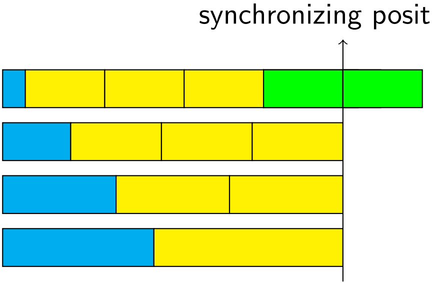
Synchronizing positions and heaps



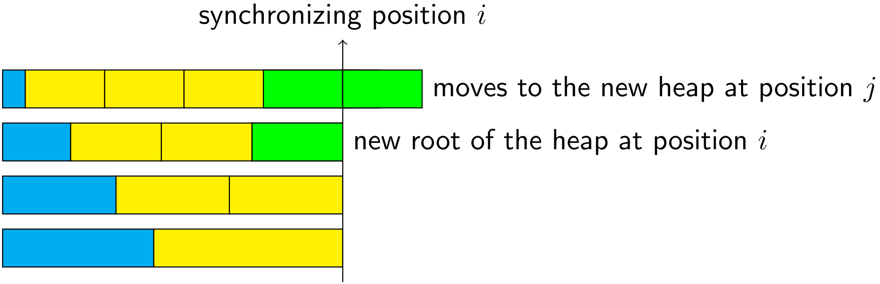
Synchronizing positions and heaps



Synchronizing positions and heaps



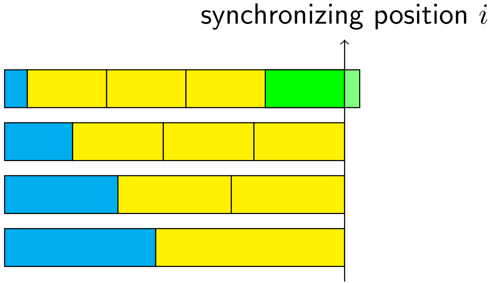
Synchronizing positions and heaps



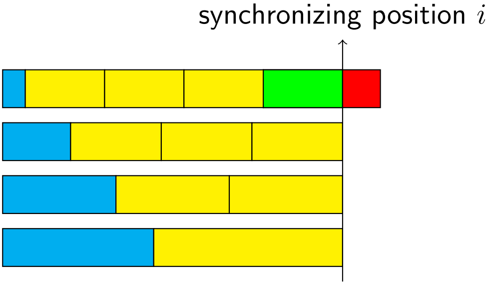
New heaps

Each new heap created at each position j is initialized with all the "trivial" periods (h, p) such that $h + p = j$ and $\mathcal{P}(w[1..h]) \subset \mathcal{P}(w[h+1..j])$.

Synchronizing positions and heaps



Synchronizing positions and heaps



Synchronizing positions and heaps

Example

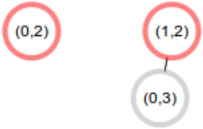
$$w = \mathtt{a}$$

Example

$w = \texttt{ab}$

Example

$$w = \texttt{aba}$$



Example

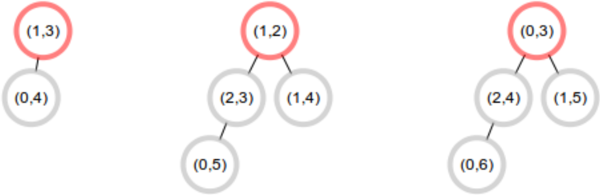
$w = \texttt{abaa}$

Example

$$w = \texttt{abaab}$$

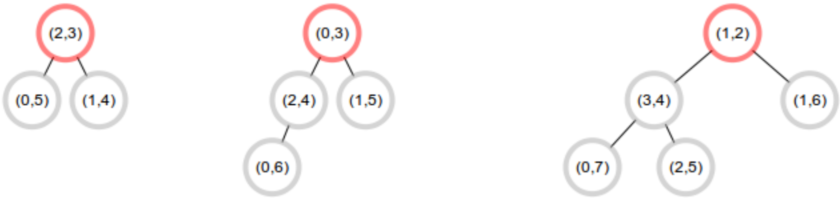
Example

$w = \texttt{abaaba}$



Example

$$w = \texttt{abaabab}$$



Example

$w =$ `abaababa`

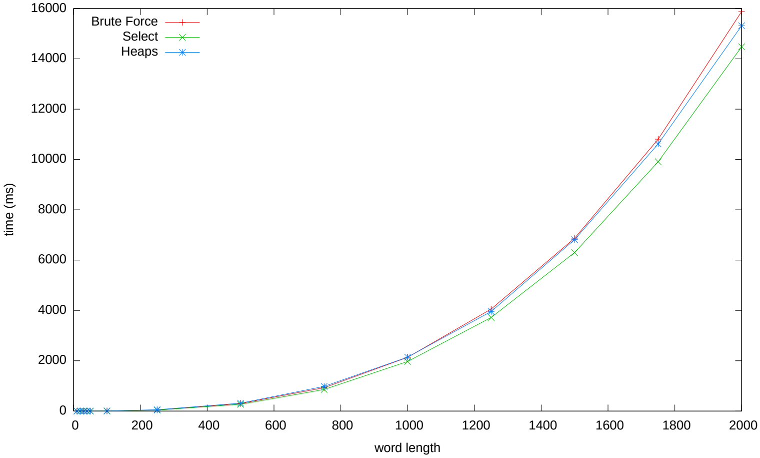
Complexity

Time: $O(n^2 \times (n \log n) \times \sigma)$
Space: $O(n^2)$

Outline

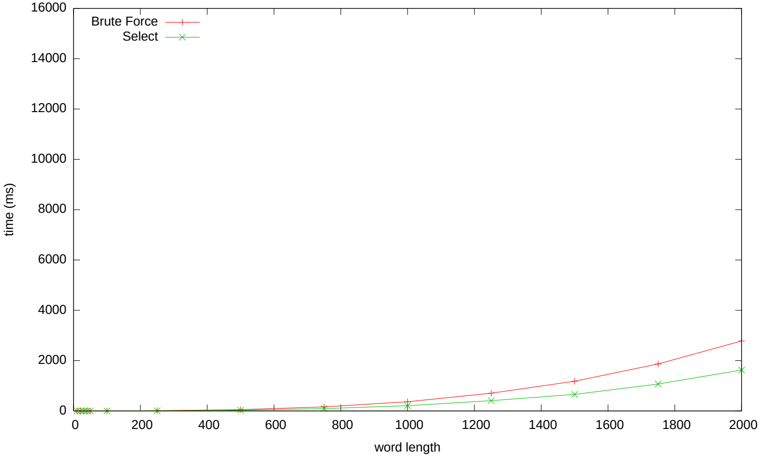
Results

Running times on alphabet size 2 and on random words of lengths 10 up to 2000.



Results

Running times on alphabet size 2 and on random words of lengths 10 up to 2000. Only computing (h, p) such that $h + 2p \leqslant |w|$.



Outline

Conclusion

Conclusion

Different algorithms to compute all the Abelian periods of a word

Perspectives

Set of cutting positions

$$C(h,p) = \{h + kp \mid k \geq 0 \text{ and } h + kp \leq |w|\}$$

Non-deducible Abelian period

An Abelian period (h, p) is non-deducible for the word w if for any other Abelian period (h_0, p_0) of w one has $C(h, p) \nsubseteq C(h_0, p_0)$.
Otherwise (h, p) is deducible.

Perspectives

Perspectives

- improve complexity
- compute the non-deducible Abelian periods
- Abelian periods in Fibonacci words
- Abelian periods in Sturmian words
- compute the Abelian border array
- applications such as looking for repeated regions with same nucleotides composition

References

S. Avgustinovich, J. Karhumäki, and S. Puzynina:
On Abelian versions of Critical Factorization Theorem,
in Proceedings of the 13th Mons Theoretical Computer Science Days, 2010.

F. Blanchet-Sadri, A. Tebbe, and A. Veprauskas:
Fine and Wilf's theorem for abelian periods in partial words,
in Proceedings of the 13th Mons Theoretical Computer Science Days, 2010.

P. Burcsi, F. Cicalese, G. Fici, and Z. Lipták:
On table arrangements, scrabble freaks, and jumbled pattern matching,
in Fun with Algorithms, 5th International Conference, FUN 2010, P. Boldi and L. Gargano, eds., vol. 6099 of Lecture Notes in Computer Science, Springer, 2010, pp. 89–101.

References

S. Constantinescu and L. Ilie:
Fine and Wilf's theorem for abelian periods.
Bull. Europ. Assoc. Theor. Comput. Sci., 89 2006, pp. 167–170.

L. J. Cummings and W. F. Smyth:
Weak repetitions in strings.
J. Combinatorial Mathematics and Combinatorial Computing, 24 1997, pp. 33–48.

M. Domaratzki and N. Rampersad:
Abelian primitive words.
arXiv, 1006.4104 2010.

References

G. Navarro and V. Mäkinen:
Compressed full-text indexes.
ACM Comput. Surv., 39 April 2007.

A. Samsonov and A. Shur:
On abelian repetition threshold,
in Proceedings of the 13th Mons Theoretical Computer Science Days, 2010.