Fast Computation of Abelian Runs

Gabriele Fici [1] Tomasz Kociumaka [2] Thierry Lecroq [3] Arnaud Lefebvre [3] Élise Prieur-Gaston [3]

[1] University of Palermo, Italy
[2] University of Warsaw, Poland
[3] Normandie Université, Université de Rouen, LITIS EA 4108, France

SeqBio 2015
November 26th-27th 2015 – Orsay, France

Classical periods and runs

Definition

An integer p is a period of a string w if $w[i] = w[i + p]$ for $0 \leq i \leq |w| - p - 1$.

Example

abaabaab has period 3: aba · aba · ab

Definition

A factor x of w is a run if it has maximal periodicity (cannot be extended to the left nor to the right).

Example

ababa in a run of abaababaa: aba·ab·ab·a·a

Classical periods and runs

Definition

An integer p is a period of a string w if $w[i] = w[i+p]$ for $0 \leq i \leq |w| - p - 1$.

Example

abaabaab has period 3: aba · aba · ab

Definition

A factor x of w is a run if it has maximal periodicity (cannot be extended to the left nor to the right).

Example

ababa in a run of abaababaa: aba·a·ba·ba·a

Abelian period

The Parikh vector $\mathcal{P}_w$ of the string w over the ordered alphabet $\Sigma = \{a_1, a_2, \ldots a_\sigma\}$ is $\mathcal{P}_w = (|w|_{a_1}, |w|_{a_2}, \ldots, |w|_{a_\sigma})$.

E.g. $\mathcal{P}_{\texttt{aacabb}} = (3, 2, 1)$.

Let $|\mathcal{P}_w|$ be the norm of $\mathcal{P}$ defined by $|\mathcal{P}_w| = \sum_{i=1}^{\sigma} |w|_{a_i}$.
E.g. $|\mathcal{P}_{\texttt{aacabb}}| = 6$.

Definition (Constantinescu, Ilie 2006)

A Parikh vector $\mathcal{P}$ is an abelian period for a string w if

$$w = u_0 u_1 \cdots u_{k-1} u_k$$

for some $k > 2$, where $\mathcal{P}_{u_0} \subset \mathcal{P}_{u_1} = \cdots = \mathcal{P}_{u_{k-1}} \supset \mathcal{P}_{u_k}$, and $\mathcal{P}_{u_1} = \mathcal{P}$.

u_0 and u_k are called resp. head and tail of the abelian period.

Example

$(1, 1)$ is the smallest abelian period of $w = \texttt{abaab} = \texttt{a} \cdot \texttt{ba} \cdot \texttt{ab} \cdot \varepsilon$.

Motivations

J. Mendivelso, C. Pino, L. F. Niño, Y. J. Pinzón
Approximate Abelian Periods to Find Motifs in Biological Sequences
CIBB 2014

They analysed gene expressions time series: they identifyied periodic changes in expression levels in the cell-cycle of *Megasphaera cerevisiae*.

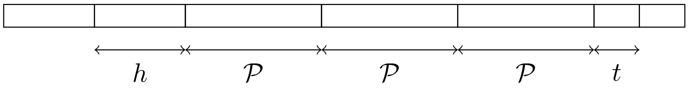
Abelian run

Definition

A substring with abelian period $\mathcal{P}$ is maximal if it cannot be extended to the left nor to the right by one letter keeping the same abelian period $\mathcal{P}$.

Definition

An abelian run of period $\mathcal{P}$ is an occurrence of a maximal substring of period $\mathcal{P}$ containing at least two occurrences of $\mathcal{P}$.



Example

$w =$ abababaaa. The prefix ab · ab · a has abelian period $(1, 1)$ but it is not an abelian run since the prefix a · ba · ba · a has also abelian period $(1, 1)$. This latter is an abelian run of period $(1, 1)$.

The Problems

Problem 1

Given a string w of length n and a Parikh vector $\mathcal{P}$, find all the abelian runs with period $\mathcal{P}$ occurring in w,

Problem 2

Given a string w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

Problem 3

Given a string w of length n, find all the abelian runs occurring in w,

The Problems

Problem 1

Given a string w of length n and a Parikh vector $\mathcal{P}$, find all the abelian runs with period $\mathcal{P}$ occurring in w,
$\rightarrow$ an $O(n)$-time and $O(\sigma + |\mathcal{P}|)$-space algorithm that solves this problem online, i.e., processes positions of the string from left to right and outputs the runs ending in position i when processing position $i + 1$.

Problem 2

Given a string w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

Problem 3

Given a string w of length n, find all the abelian runs occurring in w,

The Problems

Problem 1

Given a string w of length n and a Parikh vector $\mathcal{P}$, find all the abelian runs with period $\mathcal{P}$ occurring in w,
$\rightarrow$ an $O(n)$-time and $O(\sigma + |\mathcal{P}|)$-space algorithm that solves this problem online, i.e., processes positions of the string from left to right and outputs the runs ending in position i when processing position $i+1$.

Problem 2

Given a string w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,
$\rightarrow$ an $O(np)$-time online algorithm.

Problem 3

Given a string w of length n, find all the abelian runs occurring in w,

The Problems

Problem 1

Given a string w of length n and a Parikh vector $\mathcal{P}$, find all the abelian runs with period $\mathcal{P}$ occurring in w,
$\rightarrow$ an $O(n)$-time and $O(\sigma + |\mathcal{P}|)$-space algorithm that solves this problem online, i.e., processes positions of the string from left to right and outputs the runs ending in position i when processing position $i + 1$.

Problem 2

Given a string w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,
$\rightarrow$ an $O(np)$-time online algorithm.

Problem 3

Given a string w of length n, find all the abelian runs occurring in w,
$\rightarrow$ an $O(n^2)$-time (resp. $O(n^2 \log \sigma)$-time) offline randomized (resp. deterministic) algorithm.

Problem 1: Previous work

Problem 1

Given a string w of length n and a Parikh vector $\mathcal{P}$, find all the abelian runs with period $\mathcal{P}$ occurring in w,

$O(np)$-time and $O(\sigma + p)$ space online solution in

G. Fici, T. L., A. Lefebvre and É. Prieur-Gaston
Online Computation of Abelian Runs
LATA 2015

Problem 1

Problem 1

Given a string w of length n and a Parikh vector $\mathcal{P}$, find all the abelian runs with period $\mathcal{P}$ occurring in w,

Lemma

If $w[i\,.\,.\,j]$ and $w[i'\,.\,.\,j']$ have abelian period $\mathcal{P}$ and if $w[i\,.\,.\,j]$ is properly contained in $w[i'\,.\,.\,j']$ then $w[i\,.\,.\,j]$ is not an abelian run with period $\mathcal{P}$.

Corollary

There is at most 1 abelian run with period $\mathcal{P}$ starting at each position of w.

Problem 1: Anchor

Anchor

Given a string w,
if $w[i..j] = u_0 \cdots u_k$ has abelian period $\mathcal{P}$, with $|\mathcal{P}| = p$
and i_s is the starting position of u_s in w with $1 \leq s \leq k$
then $i_s \bmod p$ is called the anchor of the factorization.

Anchored period

$w[i..j]$ has abelian period $\mathcal{P}$ anchored at k if it has abelian period $\mathcal{P}$ whose anchor is $k \bmod p$.

Anchored run

$w[i..j]$ is a k-anchored run with period $\mathcal{P}$ if it has abelian period $|\mathcal{P}|$ anchored at k and if it is maximal ($w[i-1..j]$ and $w[i..j+1]$ if they exist have no abelian period $|\mathcal{P}|$ anchored at k).

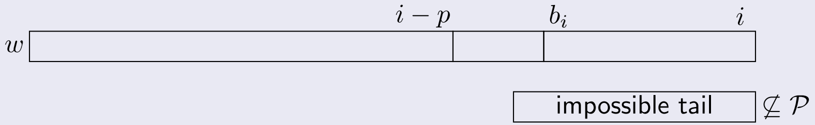
Problem 1

Definition

Let $B_i[k]$ be the starting position of the longest suffix of $w[0..i]$ which has period $\mathcal{P}$ anchored at k (or ∞) for $0 \le k < p$.
Let b_i be the starting position of the longest suffix of $w[0..i]$ whose Parikh vector is contained in or equal to $\mathcal{P}$.

Lemma

$B_i[k \bmod p] \le k$ for $b_i \le k \le i+1$
and
$B_i[k \bmod p] = \infty$ for $i - p + 1 < k < b_i$



$B_i[k \bmod p] = \infty$ for $i - p + 1 < k < b_i$.

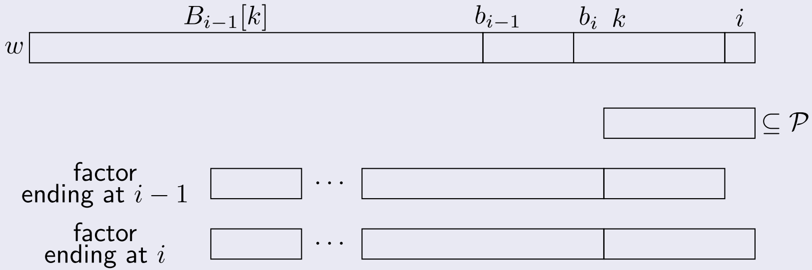
Problem 1

Computation of B_i from b_{i-1}, b_i and B_{i-1}

Lemma

1. $B_i[k \bmod p] = \infty \neq B_{i-1}[k \bmod p]$ for $\max\{i-p+1, b_{i-1}\} \leq k < b_i$
2. $B_i[k \bmod p] = B_{i-1}[k \bmod p]$ for $i-p+1 < k < b_{i-1}$ and for $b_i \leq k \leq i$
3. $B_i[i+1 \bmod p] = \begin{cases} b_i & \text{if } b_i > i-p+1 \\ B_{i-1}[i-p+1 \bmod p] & \text{otherwise} \end{cases}$

Problem 1



$B_i[k \bmod p] = B_{i-1}[k \bmod p]$ for $b_i \leq k \leq i$.

Problem 1

Lemma

$w[b..i-1]$ is a k-anchored run with period $\mathcal{P}$ iff
$B_{i-1}[k \bmod p] = b \leq k - 2p$ and $B_i[k \bmod p] > b$

Problem 1

Lemma

$w[b\,.\,.\,i-1]$ is an abelian run with period $\mathcal{P}$ iff
it is a k-anchored run with period $\mathcal{P}$ and
$B_{i-1}[k' \bmod p] \geq b$ and $B_i[k' \bmod p] > b$
for every k'

Problem 1

$\mathcal{P} = (2, 2)$

	0	1	2	3	4	5	6	7	8	9	10	11
w	a	b	a	a	b	a	b	a	a	b	b	b
$\uparrow$												

	0	1	2	3
B	0	∞	∞	∞

$L = (0)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	0	0	∞	∞

$L = (0, 1)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	0	0	0	∞

$L = (0, 1, 2)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	0	0	0	0

$L = (0, 1, 2, 3)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	∞	0	0	0

$L = (\emptyset, 1, 2, 3)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	0	0	0

$L = (1, 2, 3, 0)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	0	0	0

$L = (1, 2, 3, 0)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	0	0	0

$L = (1, 2, 3, 0)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	0	∞	0

$L = (1, \not{2}, 3, 0)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	0	3	0

$L = (1, 3, 0, 2)$

Problem 1

$\mathcal{P} = (2,2)$

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

	0	1	2	3
B	1	0	3	0

$L = (1,3,0,2)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	0	3	0

$L = (1, 3, 0, 2)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	0	3	0

$L = (1, 3, 0, 2)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	0	3	0

$L = (1, 3, 0, 2)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	∞	3	0

$L = (\not{1}, 3, 0, 2)$

Problem 1

$\mathcal{P} = (2,2)$

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

	0	1	2	3
B	1	6	3	0

$L = (3,0,2,1)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	6	3	0

$L = (3, 0, 2, 1)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	6	3	0

$L = (3, 0, 2, 1)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	6	3	0

$L = (3, 0, 2, 1)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	1	6	3	0

$L = (3, 0, 2, 1)$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	∞	6	3	0

$L = (3, \not{0}, 2, 1)$

Problem 1

$\mathcal{P} = (2, 2)$

	0	1	2	3	4	5	6	7	8	9	10	11
w	a	b	a	a	b	a	b	a	a	b	b	b
												$\uparrow$

	0	1	2	3
B	∞	∞	3	0

$L = (3, 2, \not{1})$

Problem 1

$\mathcal{P} = (2, 2)$

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

	0	1	2	3
B	10	∞	3	0

$L = (3, 2, 0)$

Problem 1

$\mathcal{P} = (2,2)$

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

↑

	0	1	2	3
B	10	∞	∞	0

$L = (3, 0)$ $\qquad$ $\textsc{Output}(0, 3, 1, 11)$

Problem 1

$\mathcal{P} = (2, 2)$

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

$\uparrow$

	0	1	2	3
B	10	∞	∞	∞

$L = (0)$

Problem 1

$\mathcal{P} = (2, 2)$

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

$\uparrow$

	0	1	2	3
B	∞	∞	∞	∞

$L = ()$

Problem 2

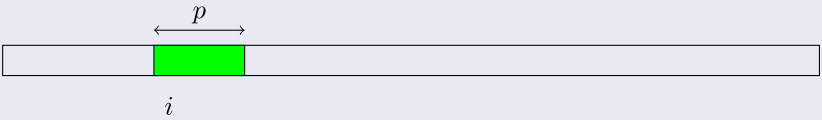
Given a w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

k-anchored runs

Problem 2

Given a w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

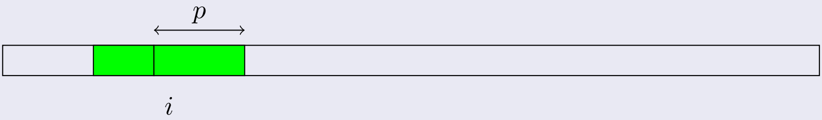
k-anchored runs



Problem 2

Given a w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

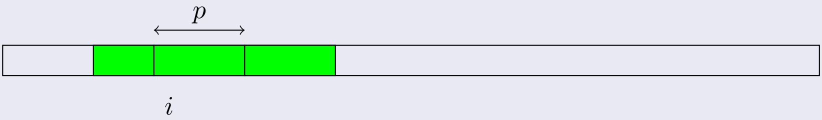
k-anchored runs



Problem 2

Given a w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

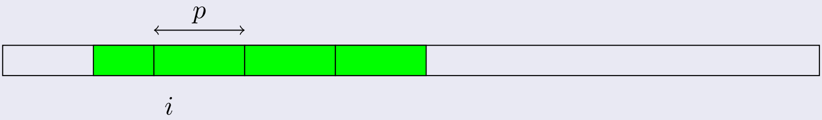
k-anchored runs



Problem 2

Given a w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

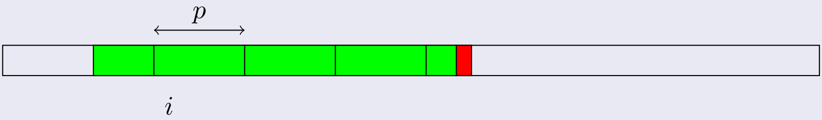
k-anchored runs



Problem 2

Given a w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

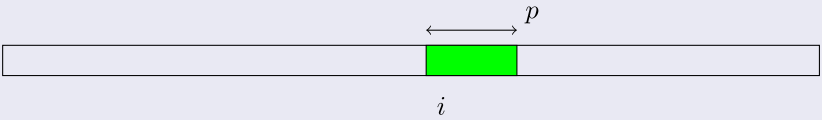
k-anchored runs



Problem 2

Given a w of length n and an integer p, find all the abelian runs with period $\mathcal{P}$ occurring in w such that $|\mathcal{P}| = p$,

k-anchored runs



Problem 2: Idea

Run algorithm for Problem 1 in parallel for each of the p possible anchors.

Problem 2: Idea

Run algorithm for Problem 1 in parallel for each of the p possible anchors.

$\rightarrow O(np)$ time algorithm

Problem 3: Previous work

Matsuda, Inenaga, Bannai & Takeda, PSC 2014

- Computation of all abelian squares using [Cummings & Smyth, 1997]:

$$L_i = \{j \mid \mathcal{P}_{w[i-j+1..i]} = \mathcal{P}_{w[i+1..i+j]}, 0 \leq j \leq \min\{i+1, n-i\}\}$$

  The L_i's are stored in a 2-dimensional boolean array L of size $\lfloor n/2 \rfloor \times (n-1)$: $L[j,i] = 1$ if $j \in L_i$ and $L[j,i] = 0$ otherwise.
- For each $1 \leq j \leq \lfloor n/2 \rfloor$ all maximal abelian repetitions of period length j are computed in $O(n)$: the j-th row of L is scanned in increasing order of the column index.
- clever computation of heads and tails.

$\longrightarrow O(n^2)$ offline computation of anchored runs

Problem 3: Previous work

L for $w =$ abaababaabbb

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

Problem 3: Previous work

L for w = abaababaabbb

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

Problem 3: Previous work

L for $w =$ abaababaabbb

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

Problem 3: Previous work

L for $w =$ abaababaabbb

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

Problem 3: Previous work

L for $w =$ abaababaabbb

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

Problem 3: Previous work

L for w = abaababaabbb

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

Problem 3: Previous work

L for $w =$ abaababaabbb

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

Problem 3: Previous work

L for $w =$ abaababaabbb

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

Problem 3: Previous work

L for $w =$ abaababaabbb

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

Problem 3: Previous work

L for $w =$ abaababaabbb

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

Problem 3: All the runs

- start with the Matsuda *et al* algorithm
- filter out anchored runs which are properly contained in another anchored run with the same period

solution: naming (giving an identifier for every period)
assign to each factor of w an identifier so that 2 factors are abelian-equivalent iff their identifiers are equal

diff-representation

Idea

	a	b	c	
w	0	0	0	
a	1	0	0	$v[0] = 1$
b	1	1	0	$v[1] = 1$
a	2	1	0	$v[0] = 2$
a	3	1	0	$v[0] = 3$
b	3	2	0	$v[1] = 2$
a	4	2	0	$v[0] = 4$
b	4	3	0	$v[1] = 3$
a	5	3	0	$v[0] = 5$
a	6	3	0	$v[0] = 6$
b	6	4	0	$v[1] = 4$
b	6	5	0	$v[1] = 5$
b	6	6	0	$v[1] = 6$

diff-representation

Idea

	a	b	c	
w	0	0	0	
a	1	0	0	$v[0] = 1$
b	1	1	0	$v[1] = 1$
a	2	1	0	$v[0] = 2$
a	3	1	0	$v[0] = 3$
b	3	2	0	$v[1] = 2$
a	4	2	0	$v[0] = 4$
b	4	3	0	$v[1] = 3$
a	5	3	0	$v[0] = 5$
a	6	3	0	$v[0] = 6$
b	6	4	0	$v[1] = 4$
b	6	5	0	$v[1] = 5$
b	6	6	0	$v[1] = 6$

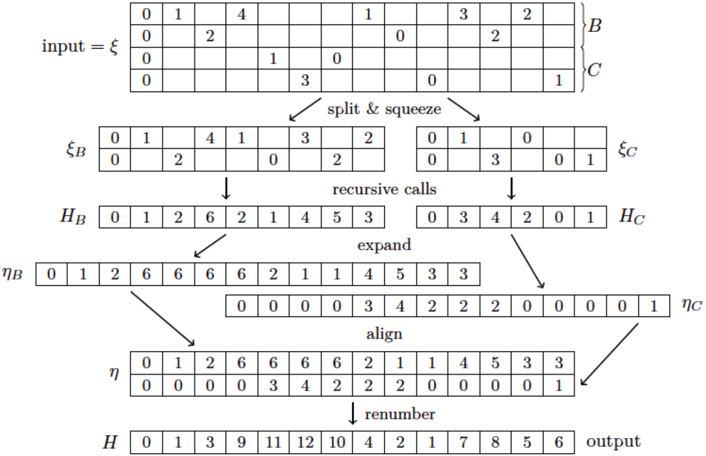
The diff-representations of all the factors of w can be computed in $O(n^2)$ time.

Naming

Theorem [Kociumaka, Radoszewsji & Rytter, STACS 2013]

Given a sequence of vectors of dimension r represented using a diff-representation of size m, the integer vector equality problem can be solved in

- $O(m + r \log m)$ time deterministically
- $O(m + r)$ time using a Monte Carlo algorithm


input = ξ
B
C
split & squeeze
ξ_B
ξ_C
recursive calls
H_B
0 1 2 6 2 1 4 5 3
0 3 4 2 0 1
H_C
expand
η_B
0 1 2 6 6 6 6 2 1 1 4 5 3 3
0 0 0 0 3 4 2 2 2 0 0 0 0 1
η_C
align
η
renumber
H
0 1 3 9 11 12 10 4 2 1 7 8 5 6
output

KRR2013

Problem 3

Given a string w of length n, all the abelian runs occurring in w, can be found in $O(n^2)$-time (resp. $O(n^2 \log \sigma)$-time) by an offline randomized (resp. deterministic) algorithm.

Conclusions

Given a string w of length n we designed:

- an online algorithm that finds all the abelian runs with period $\mathcal{P}$ occurring in w in time $O(n)$ and space $O(\sigma + |\mathcal{P}|)$
- an online algorithm that finds all the abelian runs with period $\mathcal{P}$ such that $|\mathcal{P}| = p$ occurring in w in time $O(np)$
- an offline randomized (resp. deterministic) algorithm that finds all the abelian runs occurring in w in time $O(n^2)$ (resp. $O(n^2 \log \sigma)$)

Any idea on the number of abelian runs in a string?

Thank you for your attention