Algorithms for Galois Words: Detection, Factorization, and Rotation

Diptarama Hendrian — Tokyo Medical and Dental University

Dominik Köppl — University of Yamanashi

Ryo Yoshinaka — Tohoku University

Ayumi Shinohara — Tohoku University

Ordered Alphabet and Lexicographic Order

Ordered alphabet: $\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$

$$\mathbf{a} < \mathbf{b} < \mathbf{c}$$

Lexicographic Order $<_{lex}$

$$\begin{array}{ll} S = & \mathtt{abbaaaacab} \\ & \;\; \| \;\; \| \;\; \| \;\; \| \;\; \| \;\; \wedge \\ T = & \mathtt{abbaabb} \\ & \;\; \| \;\; \| \;\; \| \\ U = & \mathtt{abb} \end{array}$$

$$U <_{lex} S <_{lex} T$$

Lyndon Words

Definition: Lyndon Words

A word S is a Lyndon word iff $S <_{lex} U$, for any proper non-empty suffix U of S.

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$

$\mathtt{a} < \mathtt{b} < \mathtt{c}$

```
aababaacb
 ababaacb
  babaacb
   abaacb
    baacb
     aacb
      acb
       cb
        b
```

Lyndon Words

Definition: Lyndon Words

A word S is a Lyndon word iff $S <_{lex} U$, for any proper non-empty suffix U of S.

$\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$

$\mathbf{a} < \mathbf{b} < \mathbf{c}$

aababaacb	Sort →	aababaacb
ababaacb		aacb
babaacb		abaacb
abaacb		ababaacb
baacb		acb
aacb		b
acb		baacb
cb		babaacb
b		cb

Infinite Repetition

Let S^ω be the infinite repetition of a word S

$$S = \texttt{aababaacb}$$

$$S^\omega = \underbrace{\texttt{aababaacb}}\underbrace{\texttt{aababaacb}}\underbrace{\texttt{aababaacb}}\ldots$$

$$\boxed{\text{Lexicographic Order on Infinite Repetition } \prec_{lex}}$$

$$X \prec_{lex} Y \Leftrightarrow X^\omega <_{lex} Y^\omega$$

$$\begin{array}{ll} S = & \texttt{abbaaaacab} \\ & \ \ \text{|| || || || || } \wedge \\ T = & \texttt{abbaabb} \\ & \ \ \text{|| || ||} \\ U = & \texttt{abb} \end{array} \qquad \begin{array}{ll} S^\omega = & \texttt{abbaaaacaba}\ldots \\ & \ \ \text{|| || || || || } \wedge \\ T^\omega = & \texttt{abbaabbabba}\ldots \\ & \ \ \text{|| || || || } \wedge \\ U^\omega = & \texttt{abbabbabba}\ldots \end{array}$$

$$U <_{lex} S <_{lex} T \qquad S \prec_{lex} T \prec_{lex} U$$

Lyndon Words

Definition: Lyndon Words

A word S is a Lyndon word iff $S <_{lex} U$, for any proper non-empty suffix U of S

Proposition

A word S is a Lyndon word iff $S \prec_{lex} U$, for any proper non-empty suffix U of S

aababaacb	aababaacb …
aacb	aacbaacba …
abaacb	abaacbaba …
ababaacb	ababaacba …
acb	acbacbacb …
b	bbbbbbbbb …
baacb	baacbbaac …
babaacb	babaacbba …
cb	cbcbcbcbc …

Alternating Order

$\Sigma = \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$

$\mathbf{a} < \mathbf{b} < \mathbf{c}$

Definition: Alternating Order $\prec_{alt}$

Given two words S and T such that $S^\omega \neq T^\omega$, with the first mismatching position j ($S^\omega[1..j-1] = T^\omega[1..j-1]$ and $S^\omega[j] \neq T^\omega[j]$).
Then, we denote $S \prec_{alt} T$ if either
(a) j is odd and $S^\omega[j] < T^\omega[j]$, or
(b) j is even and $S^\omega[j] > T^\omega[j]$.

$$
\begin{aligned}
S^\omega &= \mathbf{abbaaaacaba}\ldots \\
&\ \ \| \ \| \ \| \ \| \ \| \ \wedge \\
T^\omega &= \mathbf{abbaabbabba}\ldots \\
&\ \ \| \ \| \ \| \ \| \ \wedge \\
U^\omega &= \mathbf{abbabbabbab}\ldots
\end{aligned}
$$

$S \prec_{lex} T \prec_{lex} U$

$$
\begin{array}{rcl}
 & & \begin{matrix}1 & 2 & 3 & 4 & 5 & 6\end{matrix} \\
S^\omega &=& \mathbf{abbaaaacaba}\ldots \\
 & & \| \ \| \ \| \ \| \ \| \ \wedge \\
T^\omega &=& \mathbf{abbaabbabba}\ldots \\
 & & \| \ \| \ \| \ \| \ \wedge \\
U^\omega &=& \mathbf{abbabbabbab}\ldots
\end{array}
$$

$\boxed{T \prec_{alt} S} \prec_{alt} U$

Galois Words [Reutenauer, 2005]

Definition: Galois Words

A word S is a Galois word iff $S \prec_{alt} U$, for any proper non-empty suffix U of S.

$\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$

$\mathtt{a} < \mathtt{b} < \mathtt{c}$

Lyndon word

aababaacb ...	ababccaba ...		ababccaba ...
aacbaacba ...	babccabab ...		abaabaaba ...
abaacbaba ...	abccabaab ...	Sort by $\prec_{alt}$	abccabaab ...
ababaacba ...	bccababcc ...		aaaaaaaaa ...
acbacbacb ...	ccabaccab ...		bccababcc ...
bbbbbbbbb ...	cabacabac ...		babccabab ...
baacbbaac ...	abaabaaba ...		bababab ...
babaacbba ...	bababab ...		ccabaccab ...
cbcbcbcbc ...	aaaaaaaaa ...		cabacabac ...

Our Results

Based on Duval's algorithm (1983) on Lyndon words

We propose an online algorithm for the following task

Task	Time Complexity	Working Space
Determining Galois word	$O(n)$	$O(1)$
Computing Galois Factorization	$O(n)$	$O(1)$
Computing Galois Rotation	$O(n)$	$O(1)$

We do not include input and output space in the working space

Our Results

We propose an online algorithm for the following task

Task	Time Complexity	Working Space
Determining Galois word	$O(n)$	$O(1)$
Computing Galois Factorization	$O(n)$	$O(1)$
Computing Galois Rotation	$O(n)$	$O(1)$

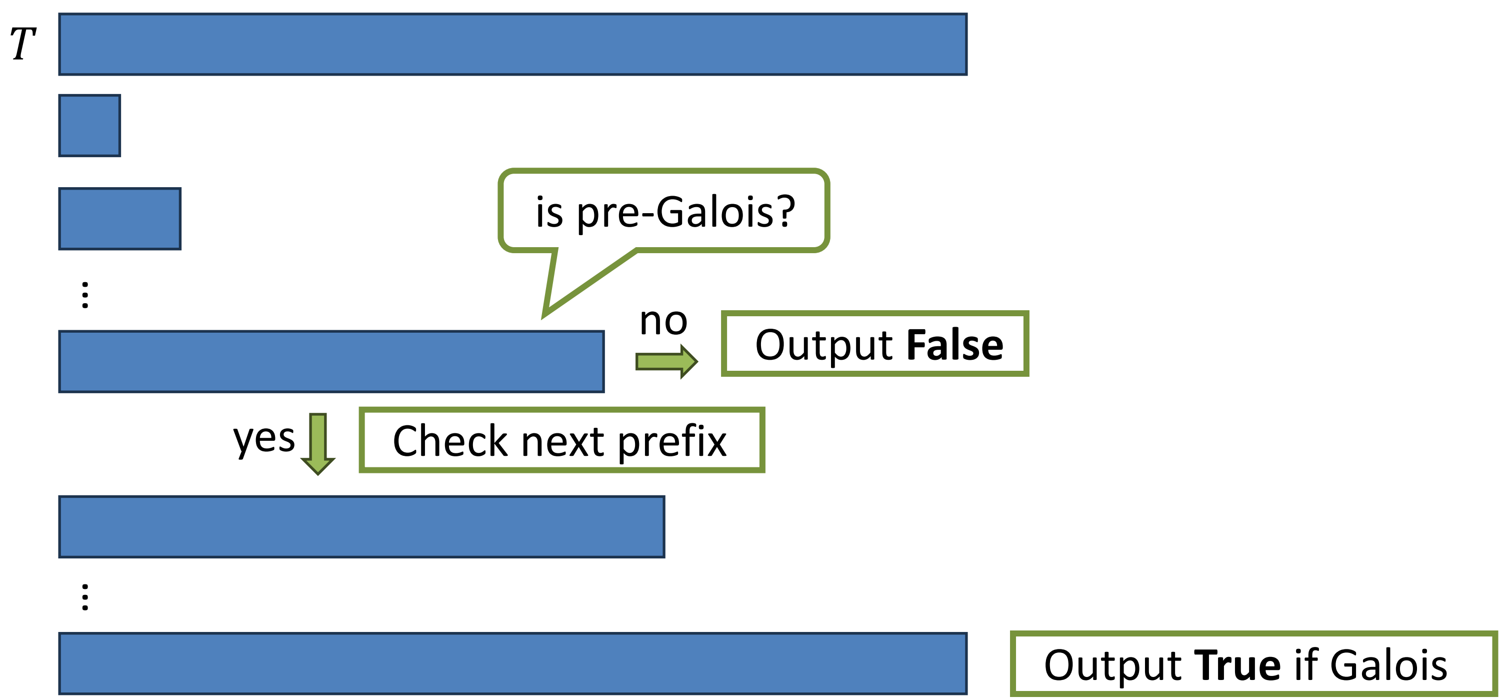
We do not include input and output space in the working space

Determining Galois Word

Definition: Determining Galois Word

Input: A non-empty word T
Output: **True** if T is a Galois word or **False** if T is not a Galois word

Our algorithm checks whether prefixes of T are pre-Galois incrementally



Pre-Galois Words

Definition: Pre-Galois Words

A word T is a pre-Galois word if every proper suffix S of T satisfies one or both of the following conditions:

(a) S is a prefix of T;

(b) $S \succ_{alt} T$.

ababccaba

A Galois word is a pre-Galois word

abaabaab

Not Galois word but a pre-Galois word

Lemma

Let T be a pre-Galois word, any non- empty suffix of T is pre-Galois.
Let S be a word that not pre-Galois, any extension SU of S is not pre-Galois.

Periods of Pre-Galois Words

Lemma: Odd Period of Pre-Galois Words

Let T be a pre-Galois word that has an odd period.
Then $T[1..p_o]$ is a Galois word, where p_o is a shortest odd period of T.

Lemma: Even Period of Pre-Galois Words

Let T be a pre-Galois word that has an odd period.
Then $T[1..p_e]$ is a Galois word if primitive,
where p_e is a shortest even period of T.

$p_o = 3$
abaa
$p_e = 4$

No odd period
abab
$p_e = 2$

$p_o = 3$
abaab
No even period

$p_o = 3$
abaabaab
$p_e = 6$

Check pre-Galois Incrementally

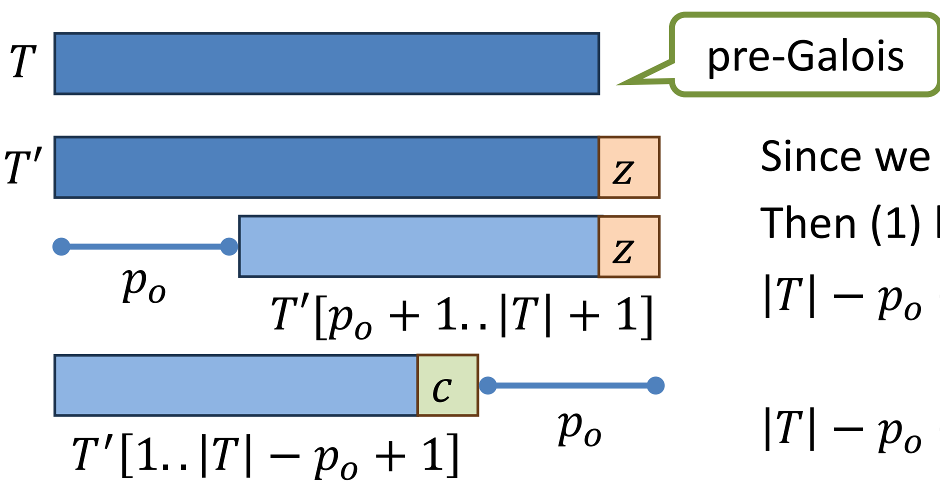
Lemma: Pre-Galois Word

Let T be a pre-Galois word, p_o be the shortest odd period of T (if exists), and p_e be the shortest even period of T (if exists).
Given a symbol z, the extension $T' = T \cdot z$ is a pre-Galois word if and only if both conditions

(1) $T'[1..|T| - p_o + 1] \preccurlyeq_{alt} T'[p_o + 1..|T| + 1]$ and

(2) $T'[1..|T| - p_e + 1] \preccurlyeq_{alt} T'[p_e + 1..|T| + 1]$ hold.

Consider the shortest odd period p_o



Since we have $T'[1..|T| - p_o] = T'[p_o + 1..|T|]$

Then (1) holds iff

$|T| - p_o + 1$ is odd and $c \leq z$

or

$|T| - p_o + 1$ is even and $c \geq z$

Updating p_o (and p_e)

Lemma: Odd Period (Equal Case)

Let T be a pre-Galois word and p_o be the shortest odd period of T (if exists).
Consider a symbol z and $T' = T \cdot z$, such that $z = T'[|T| - p_o + 1]$.
Then p_o is the shortest odd period of T'.

Lemma: Odd Period (Not Equal Case)

Let T be a pre-Galois word and p_o be the shortest odd period of T (if exists).
Consider a symbol z and $T' = T \cdot z$, such that $T'[1..|T| - p_o + 1] \prec_{alt} T'[p_o + 1..|T| + 1]$. Then $|T'|$ is the shortest odd period of T' if $|T'|$ is odd.
Otherwise T' does not have an odd period.

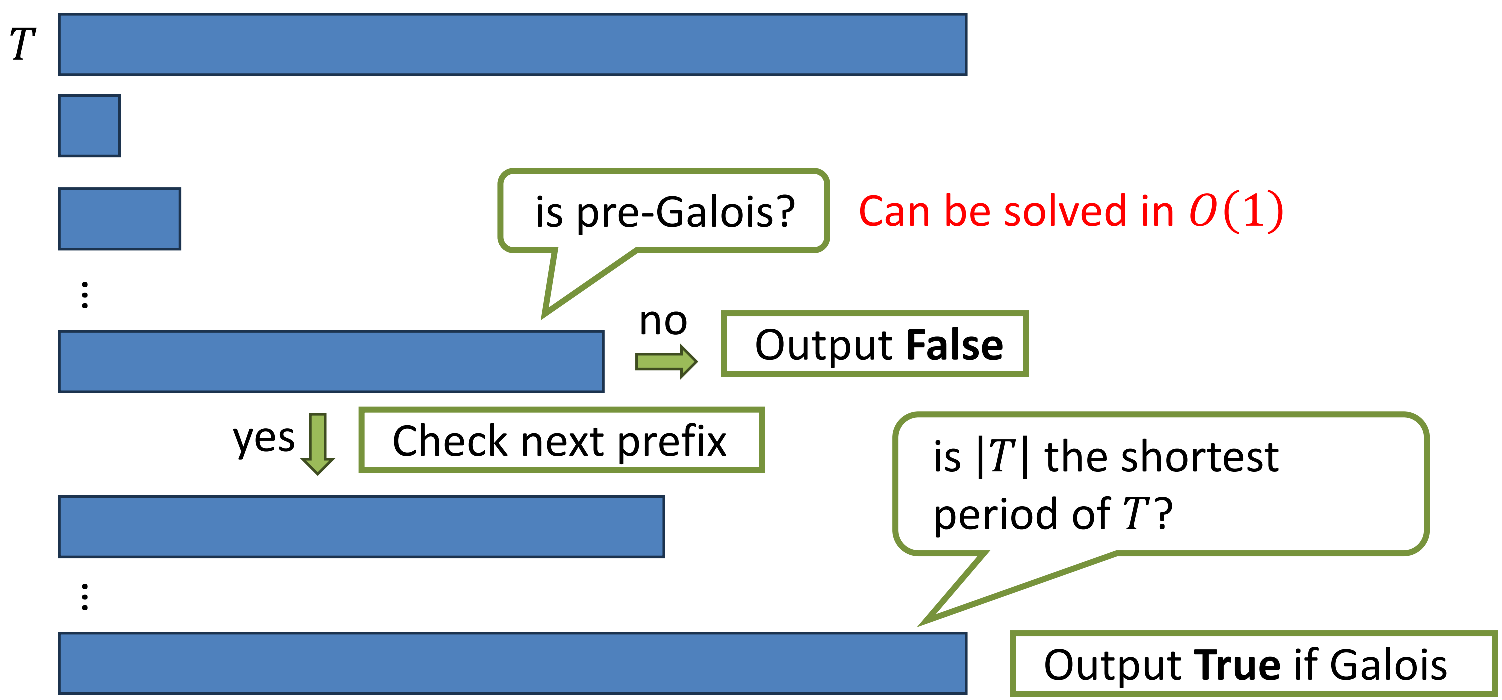
Moreover, if T does not have an odd period, $|T'|$ is the shortest odd period of $T' = T \cdot z$ for any symbol z.

Determining Galois Word

Theorem: Determining Galois Word

Given a word T, we can verify whether T is Galois in $O(|T|)$ time with $O(1)$ working space.

Our algorithm checks whether prefixes of T are pre-Galois incrementally, while maintaining their shortest odd and even periods.



Our Results

We propose an online algorithm for the following task

Task	Time Complexity	Working Space
Determining Galois word	$O(n)$	$O(1)$
Computing Galois Factorization	$O(n)$	$O(1)$
Computing Galois Rotation	$O(n)$	$O(1)$

We do not include input and output space in the working space

Galois Factorization

Definition: Galois Factorization [Reutenauer, 2005]

A factorization $G_1 \cdot G_2 \cdots G_k = T$ of a word T is the Galois factorization of T if G_i is Galois for $1 \leq i \leq k$ and $G_1 \succcurlyeq_{alt} G_2 \succcurlyeq_{alt} \cdots \succcurlyeq_{alt} G_k$.

Proposition: Uniqueness of Galois Factorization [Reutenauer, 2005]

For any word T, there exists a unique Galois factorization of T.

$T =$ **abacabaabacababacabaab**

Galois factorization of T

ab | acabaab | acababacabaab

Lyndon factorization of T

abac | ab | aabacababacab | aab

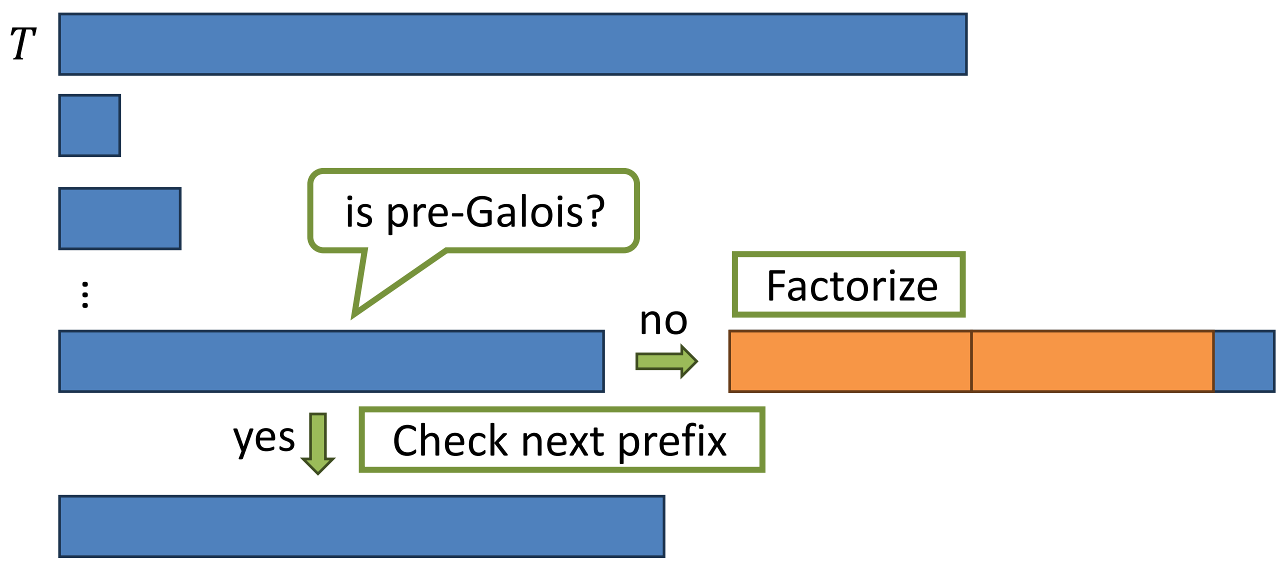
Computing Galois Factorization

Definition: Computing Galois Factorization

Input: A non-empty word T

Output: $(G_1, G_2, \ldots, G_k)$ such that

$$G_1 \cdot G_2 \cdots G_k = T \text{ is the Galois factorization of } T$$



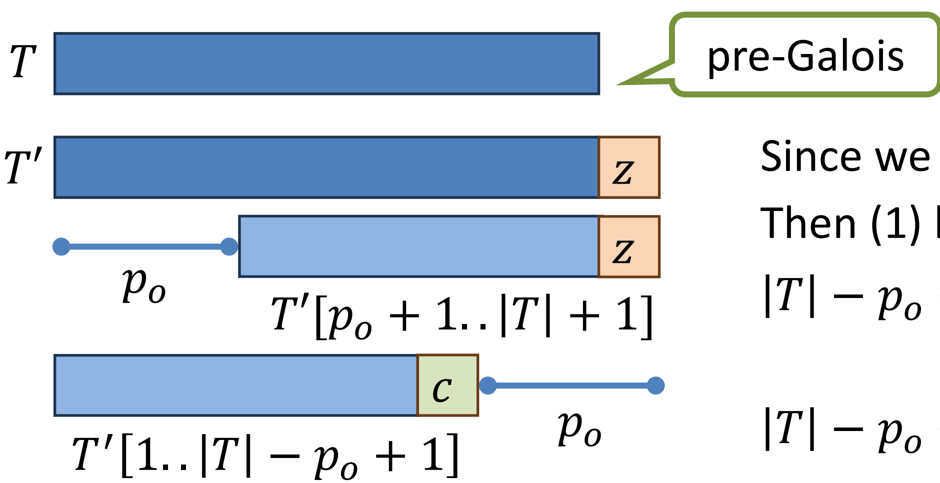
Check pre-Galois Incrementally

Lemma: Pre-Galois Word

Let T be a pre-Galois word, p_o be the shortest odd period of T (if exists), and p_e be the shortest even period of T (if exists).
Given a symbol z, the extension $T' = T \cdot z$ is a pre-Galois word if and only if both conditions

(1) $T'[1..|T| - p_o + 1] \preccurlyeq_{alt} T'[p_o + 1..|T| + 1]$ and
(2) $T'[1..|T| - p_e + 1] \preccurlyeq_{alt} T'[p_e + 1..|T| + 1]$ hold.

Consider the shortest odd period p_o



Since we have $T'[1..|T| - p_o] = T'[p_o + 1..|T|]$
Then (1) holds iff

$|T| - p_o + 1$ is odd and $c \le z$

or

$|T| - p_o + 1$ is even and $c \ge z$

Properties of Galois Factorization

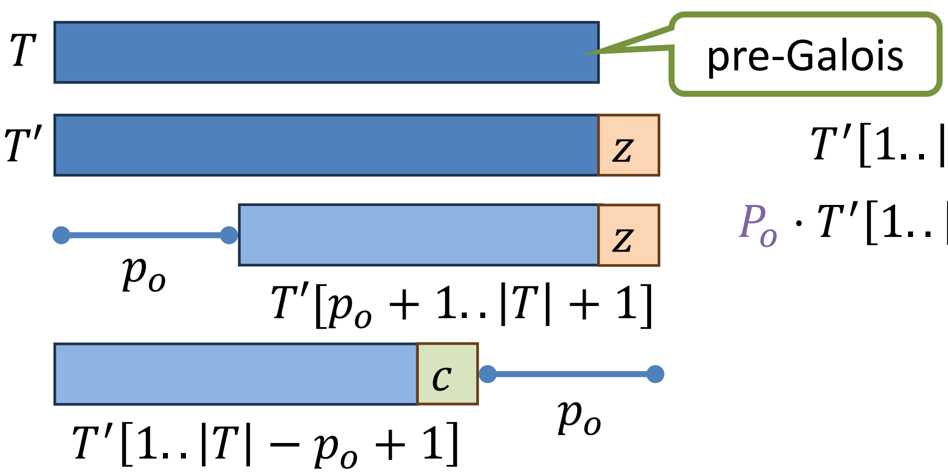
Lemma: First Factor of Galois Factorization [Dolce, et al. 2019]

Let $G_1 \cdot G_2 \cdots G_k = T$ be the Galois factorization of a word T of length n.
Let P be shortest non-empty prefix of T such that
$P \succcurlyeq_{alt} T$ if $|P|$ is even and $P \preccurlyeq_{alt} T$ if $|P|$ is odd.
Then we have

$$P = \begin{cases} G_1^2 & \text{if } |G_1| \text{ is odd, } m \text{ is even, and } m < k, \\ G_1 & \text{otherwise,} \end{cases}$$

where m is the multiplicity of G_1, i.e., $G_i = G_1$ for $i \leq m$, but $G_{m+1} \neq G_1$.

Consider the shortest odd period p_o and let $P_o = T[1..p_o]$



$$T'[1..|T| - p_o + 1] \succ_{alt} T'[p_o + 1..|T| + 1]$$

$$P_o \cdot T'[1..|T| - p_o + 1] \prec_{alt} P_o \cdot T'[p_o + 1..|T| + 1]$$

$$P_o \prec_{alt} T$$

Properties of Galois Factorization

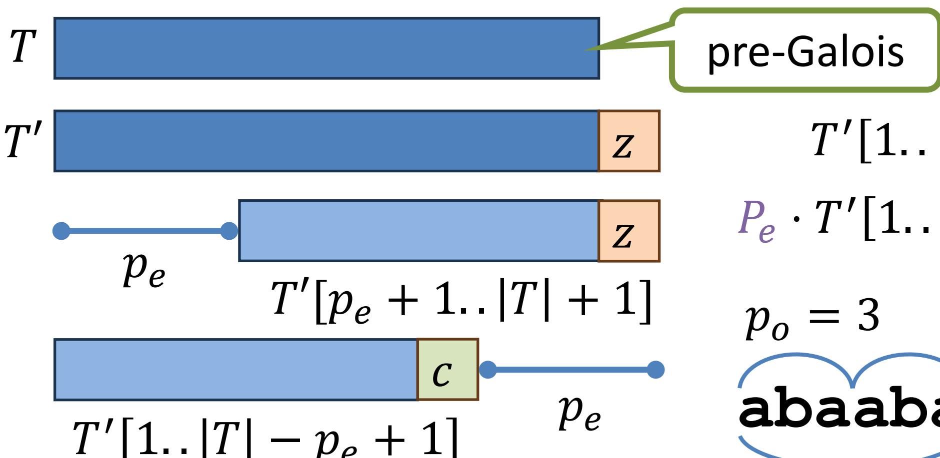
Lemma: First Factor of Galois Factorization [Dolce, et al. 2019]

Let $G_1 \cdot G_2 \cdots G_k = T$ be the Galois factorization of a word T of length n.
Let P be shortest non-empty prefix of T such that
$P \succcurlyeq_{alt} T$ if $|P|$ is even and $P \preccurlyeq_{alt} T$ if $|P|$ is odd.
Then we have

$$P = \begin{cases} G_1^2 & \text{if } |G_1| \text{ is odd, } m \text{ is even, and } m < k, \\ G_1 & \text{otherwise,} \end{cases}$$

where m is the multiplicity of G_1, i.e., $G_i = G_1$ for $i \le m$, but $G_{m+1} \neq G_1$.

Consider the shortest even period p_e and let $P_e = T[1..p_e]$



$$T'[1..|T| - p_e + 1] \succ_{alt} T'[p_e + 1..|T| + 1]$$

$$P_e \cdot T'[1..|T| - p_e + 1] \succ_{alt} P_e \cdot T'[p_e + 1..|T| + 1]$$

$p_o = 3$

$P_e \succ_{alt} T$

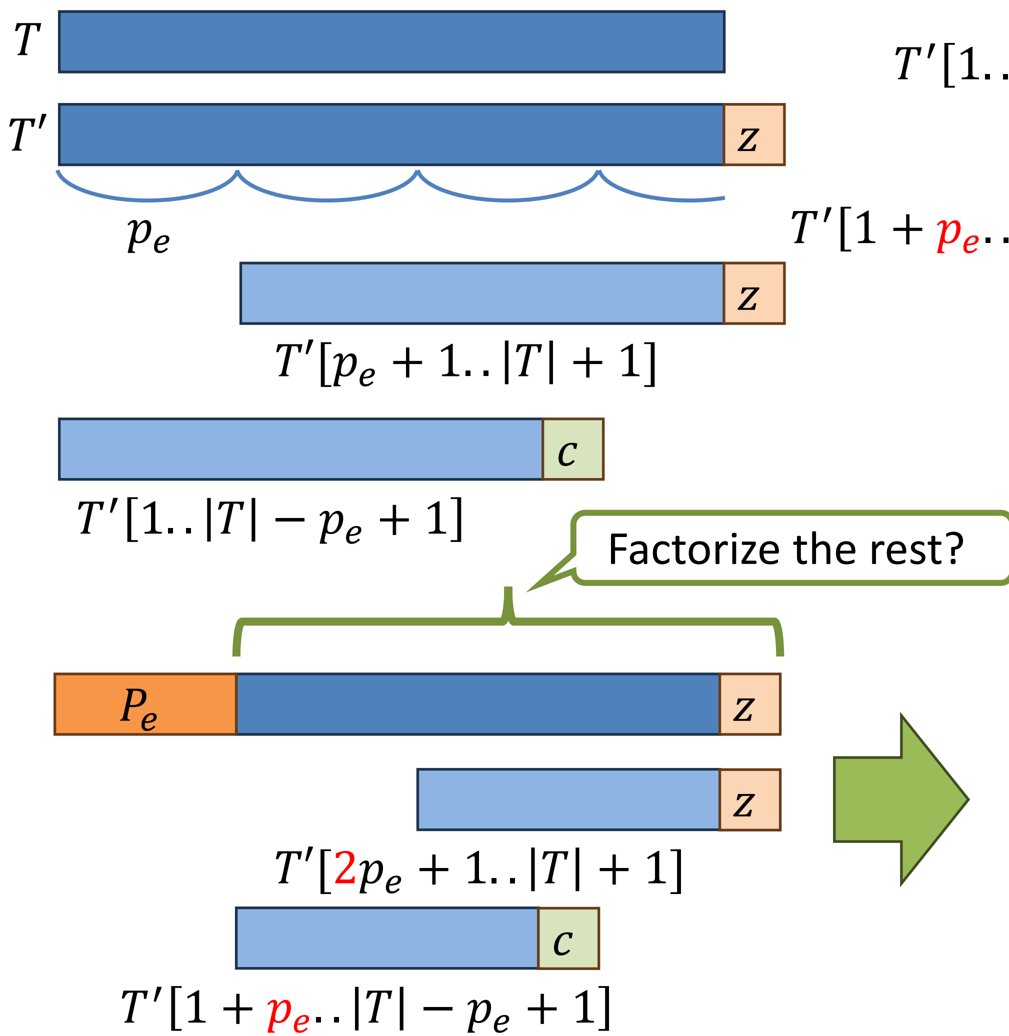
abaabaab

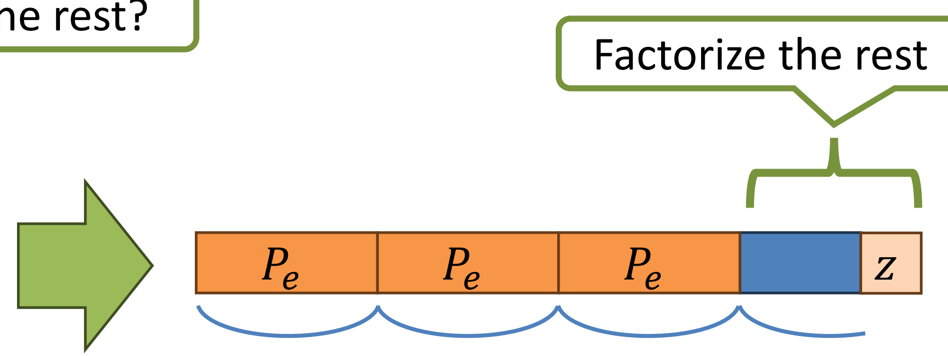
$p_e = 6$

Factorizing prefixes

Consider the shortest even period p_e and let $P = P_e = T[1..p_e]$

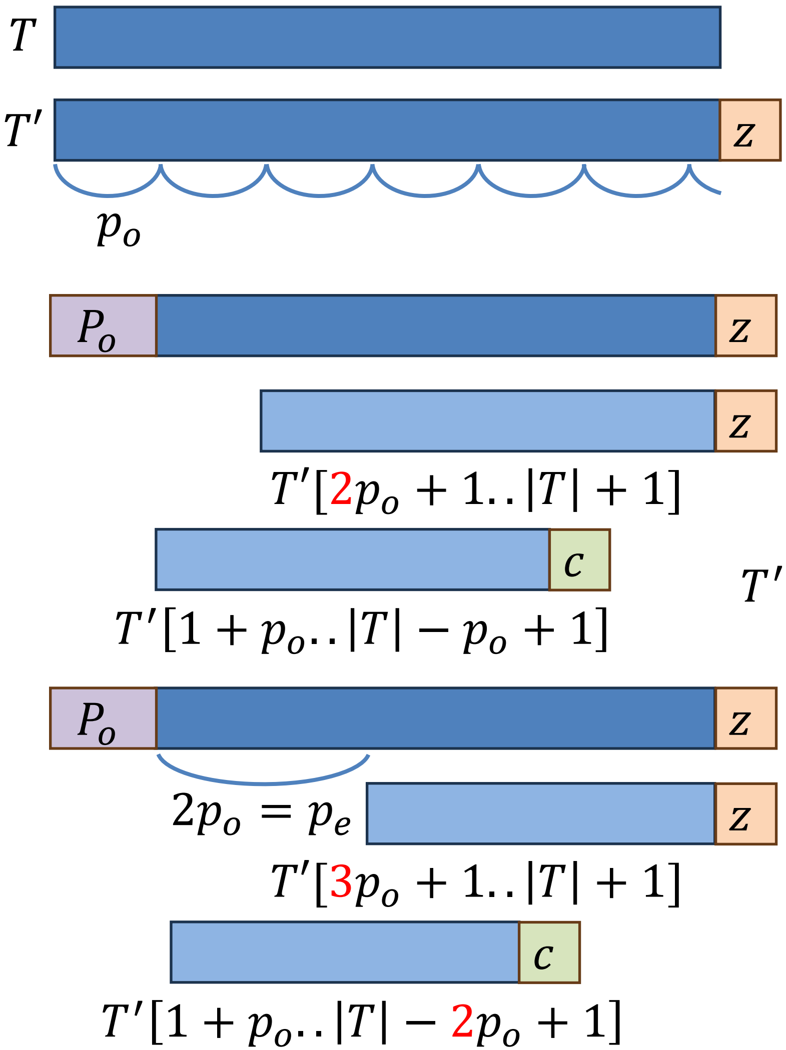
$$T'[1..|T| - p_e + 1] \succ_{alt} T'[p_e + 1..|T| + 1]$$
$$\Downarrow$$
$$T'[1 + p_e..|T| - p_e + 1] \succ_{alt} T'[2p_e + 1..|T| + 1]$$





Factorizing prefixes

Consider the shortest odd period p_o and let $P = P_o = T[1..p_o]$



$$T'[1..|T| - p_o + 1] \succ_{alt} T'[p_o + 1..|T| + 1]$$

$$\Downarrow$$

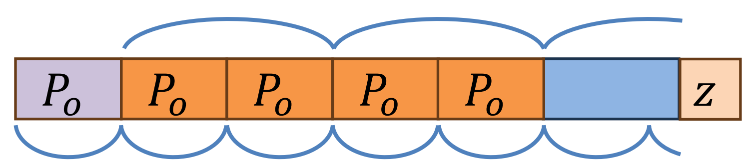
$$T'[1 + p_o..|T| - p_o + 1] \prec_{alt} T'[2p_o + 1..|T| + 1]$$

Lemma

If $|T| \geq 3p_o, p_e = 2p_o$

$$\Downarrow$$

$$T'[1 + p_o..|T| - 2p_o + 1] \succ_{alt} T'[3p_o + 1..|T| + 1]$$



Computing Galois Factorization

Definition: Computing Galois Factorization

Input: A non-empty word T
Output: $(G_1, G_2, \ldots, G_k)$ such that
$$G_1 \cdot G_2 \cdots G_k = T \text{ is the Galois factorization of } T$$

Theorem: Computing Galois Factorization

Given a word T, we can compute the Galois factorization of T in $O(|T|)$ time with $O(1)$ working space.

Our Results

We propose an online algorithm for the following task

Task	Time Complexity	Working Space
Determining Galois word	$O(n)$	$O(1)$
Computing Galois Factorization	$O(n)$	$O(1)$
Computing Galois Rotation	$O(n)$	$O(1)$

We do not include input and output space in the working space

Computing Galois Rotation

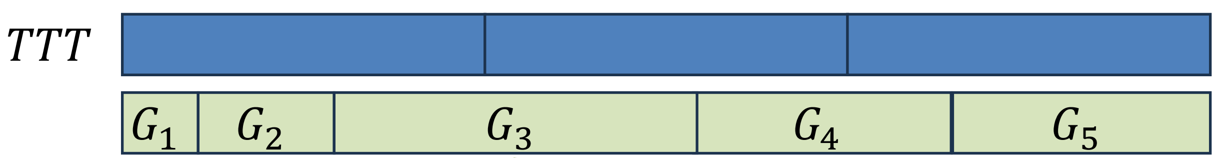
Definition: Galois Rotation

Let T be a primitive word. A rotation $R = VU$ is the **Galois rotation** of $T = UV$ if R is a Galois word.

Theorem: Computing Galois Factorization

Given a word T, we can compute the Galois rotation of T in $O(|T|)$ time with $O(1)$ working space.

The algorithm performs Galois factorization on TTT



There exists a Galois factor of length at least $|T|$, whose prefix of length $|T|$ is also a Galois word

$T = \mathbf{bca|abca}$

$TT = \mathbf{bc|a|abc|abca|abca}$

$TTT = \mathbf{bc|a|abc|abcaabc|abca|abca}$

Conclusion

We propose an online algorithm for the following task

Task	Time Complexity	Working Space
Determining Galois word	$O(n)$	$O(1)$
Computing Galois Factorization	$O(n)$	$O(1)$
Computing Galois Rotation	$O(n)$	$O(1)$

Future Work

- Enumeration of Galois words
- Algorithms for general Lyndon words