Incremental computation of the set of period sets

Eric Rivals

Incremental algorithms for computing the set of period sets

Eric Rivals

LIRMM, CNRS Univ Montpellier

28 nov. 2024

pos.	0	1	2	3	4	5	6	7	
u	a	b	a	a	b	a	b	a	border lg
	a	b	a	-	-	a	b	a	3
	a	-	-	-	-	-	-	a	1

Periods, borders, and period set

Ex Word $u = \texttt{abbabbbabba}$ and $n = 11$

pos.	0	1	2	3	4	5	6	7	8	9	10	
u	a	b	b	a	b	b	b	a	b	b	a	border lg
	a	b	b	a	-	-	-	a	b	b	a	4
	a	-	-	-	-	-	-	-	-	-	a	1

- abbabbbabba of length 11 has two non trivial suffix-prefix overlaps i.e., borders: abba and a
- Its set of periods is $P = \{0, 7, 10\}$

Algorithm:
Computing the period set $P(u)$ of a word u of length n takes $\Theta(n)$ time [2, 7, 3].

Gamma Γ_n

- For each word u of Σ^n there is a unique period set (PS): $P(u)$ is a subset of $[0,1,\ldots,n-1]$
- But not all such subsets are period sets (e.g. $\{0,3,8\}$ for $n=9$)

Definition (Gamma)

$\Gamma_n := \{Q \subset [0,1,\ldots,n-1] :$ there exists $u \in \Sigma^n$ such that $P(u) = Q\}$
κ_n : the cardinality of Γ_n

Several words share the same period set: ex. length $n=4$

Period set Γ_4	# words $\sigma = 2$
$\{0,1,2,3\}$	2
$\{0,2\}$	2
$\{0,3\}$	6
$\{0\}$	6

$\Sigma = \{\mathtt{a},\mathtt{b}\}$			Period set
aaba	bbab		
abaa	babb	$\rightarrow$	$\{0,3\}$
abba	baab		

Computation of Γ_n

$$P : \begin{array}{ccc} \Sigma^n & \longrightarrow & \Gamma_n \\ u & \longmapsto & P(u) \end{array}$$

Mapping P is surjective, but not injective.

Question: How to enumerate Γ_n?

1. Naive algorithm: take each word of Σ^n, compute its period set complexity $O(n * 2^n)$ – but with which Σ?
2. Dynamic programming algorithm [4, 5] mimics predicate Ξ and stores Γ_1 until $\Gamma_{n/2}$ to compute Γ_n where predicate Ξ checks if a set is a period set in $O(n)$ time [1]

Incremental enumeration

Consequences of GO characterization property of period sets

Lemma

- *If P is a period set of Γ_n, then $P \setminus \{n-1\}$ belongs to Γ_{n-1}.*
- *Let Q be a period set of Γ_n. Then, there exists P in Γ_{n-1} such that Q equals either P or $P \cup \{n-1\}$.*

Algorithm 1: IncrementalGamma(length $n > 1$; set Γ_{n-1})

Output: Γ_n: the set of period sets for length n;

$G := \emptyset$; // G: variable to store Γ_n
for *all* $P \in \Gamma_{n-1}$ **do**
    if `certify`(P, n) **then** insert P in G ;
    $Q := P \cup \{n-1\}$ // build extension P with period $n-1$;
    if `certify`(Q, n) **then** insert Q in G ;
return G;

Theorem

The two following statements are true.

1. *Algorithm 1 using any certification correctly computes Γ_n from Γ_{n-1}.*
2. *Using the predicate Ξ as certification function, it runs in $O(n \times \kappa_n)$ time and $O(n)$ space.*

Other results

- Algorithm 1 is embarrassingly parallelizable.
- Constructive certification algorithm: it yields a witness for any period set.
- When length n increases, the dynamics of period sets is a binary tree.

Distribution of the nb of PS in Γ_{60}

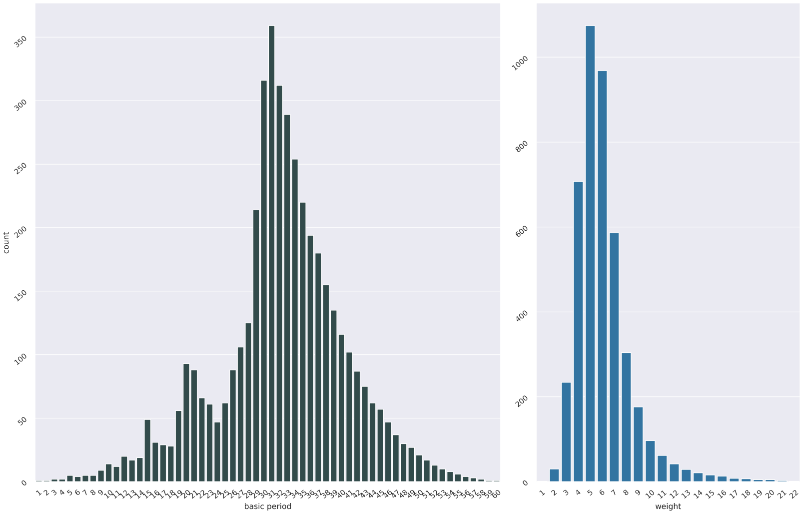

Figure: Left: distribution by basic period. Right: distribution by weight (# of periods) arXiv > cs > arXiv:2410.12077

Thank you for your attention!

Questions?

Conclusion and open questions

Take home message

1. Algorithms for
   - incremental enumeration of Γ_n in $O(n)$ space
   - binary realization of a subset of $[0, 1, \ldots, n-1]$
2. Notion of *dynamics* of period sets

Open questions and future work

- Optimal algorithms for computing the lattice structure
- Explain distribution of PS according to basic period
- Explore dynamics of PS with increasing n
- Generalizations to 2D-words

Support:
Co-financed by the Connecting Europe Facility of the European Union

Bibliography

Leo J. Guibas and Andrew M. Odlyzko.
Periods in strings.
J. of Combinatorial Theory series A, 30:19–42, 1981.
doi:10.1016/0097-3165(81)90038-8.

Dan Gusfield, Gad M. Landau, and Baruch Schieber.
An efficient algorithm for the All Pairs Suffix-Prefix Problem.
Inf Proc Letters, 41(4):181–185, 1992.
URL: http://www.sciencedirect.com/science/article/pii/002001909290176V, doi:10.1016/0020-0190(92)90176-V.

Jihyuk Lim and Kunsoo Park.
A fast algorithm for the All-Pairs Suffix–Prefix problem.
Theoretical Computer Science, 698:14–24, 2017.
URL: http://www.sciencedirect.com/science/article/pii/S0304397517305510, doi:10.1016/j.tcs.2017.07.013.

Eric Rivals and Sven Rahmann.
Combinatorics of periods in strings.
In *Proc. of ICALP 2001*, page 615–626. Springer Berlin Heidelberg, 2001.
URL: http://dx.doi.org/10.1007/3-540-48224-5_51, doi:10.1007/3-540-48224-5_51.

Eric Rivals and Sven Rahmann.
Combinatorics of periods in strings.
Journal of Combinatorial Theory, Series A, 104(1):95–113, Oct 2003.
URL: `http://dx.doi.org/10.1016/S0097-3165(03)00123-7`,
`doi:10.1016/s0097-3165(03)00123-7`.

Eric Rivals, Michelle Sweering, and Pengfei Wang.
Convergence of the Number of Period Sets in Strings.
In Kousha Etessami, Uriel Feige, and Gabriele Puppis, editors, *50th International Colloquium on Automata, Languages, and Programming (ICALP 2023)*, volume 261 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 100:1–100:14, Dagstuhl, Germany, 2023.
Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
URL: `https://drops.dagstuhl.de/opus/volltexte/2023/18152`,
`doi:10.4230/LIPIcs.ICALP.2023.100`.

William F. Smyth.
Computating Pattern in Strings.
Pearson - Addison Wesley, 2003.