Efficiently Finding All Maximal α-gapped Repeats

Paweł Gawrychowski[1], Tomohiro I[2], Shunsuke Inenaga[3], Dominik Köppl[4], and Florin Manea[5]

1 **Institute of Informatics, University of Warsaw, Poland**
gawry@mimuw.edu.pl
2 **Department of Computer Science, TU Dortmund, Germany**
tomohiro.i@cs.tu-dortmund.de
3 **Department of Informatics, Kyushu University, Japan**
inenaga@inf.kyushu-u.ac.jp
4 **Department of Computer Science, TU Dortmund, Germany**
dominik.koeppl@cs.tu-dortmund.de
5 **Department of Computer Science, Kiel University, Germany**
flm@informatik.uni-kiel.de

Abstract

For $\alpha \geq 1$, an α-gapped repeat in a word w is a factor uvu of w such that $|uv| \leq \alpha|u|$; the two occurrences of a factor u in such a repeat are called *arms*. Such a repeat is called *maximal* if its arms cannot be extended simultaneously with the same symbol to the right nor to the left. We show that the number of all maximal α-gapped repeats occurring in words of length n is upper bounded by $18\alpha n$, allowing us to construct an algorithm finding all maximal α-gapped repeats of a word on an integer alphabet of size $n^{\mathcal{O}(1)}$ in $\mathcal{O}(\alpha n)$ time. This result is optimal as there are words that have $\Theta(\alpha n)$ maximal α-gapped repeats. Our techniques can be extended to get comparable results in the case of α-gapped palindromes, i.e., factors $uvu^{\intercal}$ with $|uv| \leq \alpha|u|$.

1998 ACM Subject Classification G.2.1 Combinatorics, I.1.2 Algorithms

Keywords and phrases combinatorics on words, counting algorithms

Digital Object Identifier 10.4230/LIPIcs.STACS.2016.39

1 Introduction

Gapped repeats and palindromes are repetitive structures occurring in words that were investigated extensively within theoretical computer science (see, e.g., [11, 3, 14, 15, 16, 4, 6, 5, 10, 7, 17] and the references therein) with motivation coming especially from the analysis of DNA and RNA structures, modelling different types of tandem and interspersed repeats as well as hairpin structures; such structures are important in analysing the structural and functional information of the genetic sequences (see, e.g., [11, 3, 15]).

Besides introducing the definitions of (maximal) α-gapped repeats and palindromes, both papers [15, 16] lead to combinatorial and algorithmic problems that extend the classical results obtained for squares and palindromes. In fact, problems like how many maximal α-gapped repeats or palindromes can a word of length n contain, how efficiently can we compute the set of maximal α-gapped repeats or palindromes in a word, how efficiently can we compute the α-gapped repeat or palindrome with the longest arm, were already investigated [15, 3, 16, 10, 17, 5]. In this article we obtain the following results:

- The number of all maximal α-gapped repeats in a word of length n is at most $18\alpha n$.
- We can compute the list of all α-gapped repeats in $\mathcal{O}(\alpha n)$ time for *integer* alphabets.

33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016).
Editors: Nicolas Ollinger and Heribert Vollmer; Article No. 39; pp. 39:1–39:14

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Our techniques can be extended to show that the number of all maximal α-gapped palindromes in a word of length n is upper bounded by $28\alpha n + 7n$; they can be found in $\mathcal{O}(\alpha n)$ time. As there are words of length n that contain $\Theta(\alpha n)$ maximal α-gapped repeats (see [16]), it follows that our obtained bounds on the number of all maximal α-gapped repeats are asymptotically tight, and that we cannot hope for algorithms finding all α-gapped repeats faster in the worst case.

Our results improve those of [16] (as well as those existing in the literature before [16]). There, the authors present an algorithm computing all maximal α-gapped repeats in $\mathcal{O}(\alpha^2 n + \mathrm{occ}(n))$ time for integer alphabets, where $\mathrm{occ}(n)$ is the number of all maximal α-gapped repeats occurring in a word of length n. Further, they proved that $\mathrm{occ}(n) = \mathcal{O}(\alpha^2 n)$.

An alternative proof for the upper bound $\mathrm{occ}(n) = \mathcal{O}(\alpha n)$ was given in the very recent paper [5]. However, compared to that paper, we present a more direct proof of the $\mathcal{O}(\alpha n)$ upper bound as well as a concrete evaluation of the constant hidden by the $\mathcal{O}$-denotation.

The algorithms given in [5, 17] compute all maximal α-gapped repeats of a word in $\mathcal{O}(\alpha n + \mathrm{occ}(n))$. In the light of the upper bound $\mathcal{O}(\alpha n)$ on $\mathrm{occ}(n)$, it follows that these algorithms work in $\mathcal{O}(\alpha n)$ time, but only for constant alphabets. Extending the approach in [10], we devise an algorithm for the same problem with integer alphabets. The algorithm requires a deeper analysis than the one developed in [10] for finding the longest α-gapped repeat, and uses essentially different techniques and data structures than the ones in [5, 17].

A related problem is the computation of all factors with an exponent less than 2 that are maximal wrt. their exponents. This problem was recently investigated in [1].

2 Combinatorics on Words

Let Σ be a finite alphabet; Σ^* denotes the set of all finite words over Σ. The ***length*** of a word $w \in \Sigma^*$ is denoted by $|w|$.

For $v = xuy$ with $x, u, y \in \Sigma^*$, we call x, u and y a ***prefix***, ***factor***, and ***suffix*** of v, respectively. We denote by $w[i]$ the symbol occurring at position i in w, and by $w[i, j]$ the factor of w starting at position i and ending at position j, consisting of the catenation of the symbols $w[i], \ldots, w[j]$, where $1 \leq i \leq j \leq n$; we say that $w[i, j]$ is empty if $i > j$. By $w^\intercal$ we denote the ***mirror image*** of w. A ***period*** of a word w over Σ is a positive integer p such that $w[i] = w[j]$ for all i and j with $i \equiv j \pmod{p}$; a word that has period p is also called ***p-periodic***. Let $per(w)$ be the smallest period of w. A word w with $per(w) \leq \frac{|w|}{2}$ is called ***periodic***; otherwise, w is called ***aperiodic***. It is worth noting that the length of the overlap between two consecutive occurrences of an aperiodic factor v in w is upper bounded by $\frac{|v|}{2}$.

By $\mathcal{I} = [b, e]$ we represent the set of consecutive integers from b to e, for $b \leq e$, and call $\mathcal{I}$ an ***interval***. For an interval $\mathcal{I}$, we use the notations $\mathsf{b}(\mathcal{I})$ and $\mathsf{e}(\mathcal{I})$ to denote the beginning and end of $\mathcal{I}$; i.e., $\mathcal{I} = [\mathsf{b}(\mathcal{I}), \mathsf{e}(\mathcal{I})]$. We write $|\mathcal{I}|$ to denote the length of $\mathcal{I}$; i.e., $|\mathcal{I}| = \mathsf{e}(\mathcal{I}) - \mathsf{b}(\mathcal{I}) + 1$. A ***subword*** u of a word w is a pair $(s, [b, e])$ consisting of a factor s of w and an interval $[b, e]$ in w such that $s = w[b, e]$. While a factor is identified only by a sequence of letters, a subword is also identified by its position in the word. So subwords are always unique, while a word may contain multiple occurrences of the same factor. For two subwords u and $\overline{u}$ of a word w, we write $u = \overline{u}$ if they start at the same position in w and have the same length. We write $u \equiv \overline{u}$ if the factors identifying these subwords are the same. We implicitly use subwords both like factors of w and as intervals contained in $[1, |w|]$, e.g., we write $u \subseteq \overline{u}$ if two subwords $u = (s, [b, e]), \overline{u} = (\overline{s}, [\overline{b}, \overline{e}])$ of w satisfy $[b, e] \subseteq [\overline{b}, \overline{e}]$, i.e., $\mathsf{b}(\overline{u}) \leq \mathsf{b}(u) \leq \mathsf{e}(u) \leq \mathsf{e}(\overline{u})$. Two subwords u and $\overline{u}$ of the same word w are called ***consecutive***, iff $\mathsf{e}(u) + 1 = \mathsf{b}(\overline{u})$.

For a word w, we call a triple of consecutive subwords u_λ, v, u_ρ a ***gapped repeat*** with ***period*** $|u_\lambda v|$ and ***gap*** $|v|$ iff $u_\rho \equiv u_\lambda$. A triple of consecutive subwords u_λ, v, u_ρ is called a ***gapped palindrome*** with gap $|v|$ iff $u_\rho \equiv u_\lambda{}^\intercal$. The subwords u_λ and u_ρ are called left and right ***arm***, respectively. For $\alpha \geq 1$, the gapped repeat (palindrome) u_λ, v, u_ρ is called α***-gapped*** iff $|u_\lambda| + |v| \leq \alpha\,|u_\lambda|$. Further, it is called ***maximal*** iff its arms cannot be extended simultaneously to the right nor to the left. Let $\mathcal{G}_\alpha(w)$ (respectively, $\mathcal{G}^\intercal_\alpha(w)$) denote the set of maximal α-gapped repeats (palindromes) in w. The representation of a maximal gapped repeat (palindrome) by the subword $z := w[u_\lambda]w[v]w[u_\rho]$ is not unique – the same subword z can be composed of gapped repeats (palindromes) with different periods (different gaps). Instead, a maximal gapped repeat (palindrome) is uniquely determined by its left arm u_λ and its period (gap). By fixing w, we thus can map u_λ, v, u_ρ injectively to the pair of integers $(\mathsf{e}(u_\lambda), |u_\lambda v|)$ in case of gapped repeats, or to $(\mathsf{e}(u_\lambda), |v|)$ in case of gapped palindromes.

A ***repetition*** in a word w is a periodic factor; a ***run*** is a maximal repetition; the ***exponent*** of a run is the number of times the period fits in that run. For a word w, let $E(w)$ denote the number of runs and the sum of the exponents of runs in w, respectively. The exponent of a run r is denoted by $\exp(r)$. We use the following results from literature:

▶ **Lemma 1** ([2]). *For a word w, $E(w) < 3\,|w|$, and the number of runs is less than $|w|$.*

▶ **Corollary 2** ([5, Conclusions]). *The number of maximal 1-gapped repeats is less than n.*

▶ **Observation 3.** *The mirror image of a gapped repeat (palindrome) is a gapped repeat (palindrome) with the same period. Hence, there exist the bijections $\mathcal{G}_\alpha(w) \sim \mathcal{G}_\alpha(w^\intercal)$ and $\mathcal{G}^\intercal_\alpha(w) \sim \mathcal{G}^\intercal_\alpha(w^\intercal)$.*

2.1 Point Analysis

A pair of positive integers is called a ***point***. We use points to bound the cardinality of a subset of gapped repeats and gapped palindromes by injectively mapping a gapped repeat (palindrome) to a point as stated above. To this end, we show that some vicinity of any point generated by a member of this subset does not contain any point that is generated by another member. This vicinity is given by

▶ **Definition 4.** For any $\gamma \in (0, 1]$, we say that a point (x, y) γ***-covers*** a point (x', y') iff $x - \gamma y \leq x' \leq x$ and $y - \gamma y \leq y' \leq y$.

It is crucial that the γ factor is always multiplied with the y-coordinates. In other words, the number of γ-covers of a point $(\cdot, y)$ correlates with γ and the value y. The main property of this definition is given by

▶ **Lemma 5.** *For any $\gamma \in (0, 1]$, let $S \subset [1, n]^2 \subset \mathbb{N}^2$ be a set of points such that no two distinct points in S γ-cover the same point. Then $|S| < 3n/\gamma$.*

Proof. We estimate the maximal number of points that can be placed in $[1, n]^2 \subset \mathbb{N}^2$ such that their covered points are disjoint. First, the number of points $(\cdot, y) \in [1, n]^2$ with $y < 1/\gamma$ is less than n/γ. Second, if a point $(\cdot, y)$ satisfies $2^l/\gamma \leq y < 2^{l+1}/\gamma$ for some integer $l \geq 0$, the point $(\cdot, y)$ γ-covers at least $2^l \times 2^l$ points, or to put it differently, this point γ-covers at least 2^l points $(\cdot, y')$ with $y - 2^l \leq y' \leq y$. In other words, there are at most $n/(2^l\gamma)$ points in S with $2^l/\gamma \leq y < 2^{l+1}/\gamma$. Hence, $|S| < n/\gamma + \sum_{l=0}^{\infty} n/(2^l\gamma) = 3n/\gamma$. ◀

Kolpakov et al. [16] split the set of maximal α-gapped repeats into three subsets, and studied the maximal size of each subset. They analysed maximal α-gapped repeats by

partitioning them into three subsets: those whose arms are contained in one or two runs, those whose arms contain a periodic prefix or suffix larger than half of the size of the arms, and those belonging to neither of the two subsets.

They showed that the first two subsets contain at most $\mathcal{O}(\alpha n)$ elements. The point analysis is used as a tool for studying the last subset. By mapping a gapped repeat to a point consisting of the end position of its left arm and its period, they showed that the points created by two different maximal α-gapped repeats cannot $\frac{1}{4\alpha}$-cover the same point. By this property, they bounded the size of the last subset by $\mathcal{O}(\alpha^2 n)$. Lemma 5 immediately improves this bound of $\mathcal{O}(\alpha^2 n)$ to $\mathcal{O}(\alpha n)$. Consequently, it shows that the number of maximal α-gapped repeats of a word of length n is $\mathcal{O}(\alpha n)$.

2.2 Upper Bound for the Number of Maximal α-gapped Repeats

We optimize the proof technique from [16] and improve the upper bound of the number of maximal α-gapped repeats in a word of length n from $\mathcal{O}(\alpha n)$ to $18\alpha n$. Unlike [16, 5], we partition the maximal α-gapped repeats differently. We categorize a gapped repeat depending on whether their left arm contains a periodic prefix or not. The two subsets are treated differently. For the ones having a periodic prefix, we think about the number of runs covering this prefix. The other category is analysed by using the results of Section 2.1. We begin with a formal definition of both subsets and analyse the former subset.

Let $0 < \beta < 1$. A gapped repeat $\sigma = u_\lambda, v, u_\rho$ belongs to $\beta\mathcal{P}_\alpha(w)$ iff u_λ contains a periodic prefix of length at least $\beta\,|u_\lambda|$. We call σ ***periodic***. Otherwise $\sigma \in \overline{\beta\mathcal{P}}_\alpha(w)$, where $\overline{\beta\mathcal{P}}_\alpha(w) := \mathcal{G}_\alpha(w) \setminus \beta\mathcal{P}_\alpha(w)$; we call σ ***aperiodic***.

▶ **Lemma 6.** *Let w be a word, $\alpha > 1$ and $0 < \beta < 1$ two real numbers. Then $|\beta\mathcal{P}_\alpha(w)|$ is at most $2\alpha E(w)/\beta$.*

Proof. Let $\sigma = (u_\lambda, v, u_\rho) \in \beta\mathcal{P}_\alpha(w)$. By definition, the left arm u_λ has a periodic prefix s_λ of length at least $\beta\,|u_\lambda|$. Let r_λ denote the run that generates s_λ, i.e., $s_\lambda \subseteq r_\lambda$ and they both have the common shortest period p. By the definition of gapped repeats, there is a right copy s_ρ of s_λ contained in u_ρ with $s_\rho = w[\mathsf{b}(s_\lambda) + |u_\lambda v|\,, \mathsf{e}(s_\lambda) + |u_\lambda v|] \equiv s_\lambda$.

Let r_ρ be a run generating s_ρ (it is possible that r_ρ and r_λ are identical). By definition, r_ρ has the same period p as r_λ. In the following, we will see that σ is uniquely determined by r_λ and the period $q := |u_\lambda v|$, if σ is a periodic gapped repeat. We will fix r_λ and pose the question how many maximal periodic gapped repeats can be generated by r_λ.

Since σ is maximal, $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$ or $\mathsf{b}(u_\rho) = \mathsf{b}(r_\rho)$ must hold; otherwise we could extend σ to the left. We analyse the case $\mathsf{b}(s_\lambda) = \mathsf{b}(r_\lambda)$, the other is treated exactly in the same way by symmetry. The gapped repeat σ is identified by r_λ and the period q. We fix r_λ and count the number of possible values for the period q. Given two different gapped repeats σ_1 and σ_2 with respective periods q_1 and q_2 such that the left arms of both are generated by r_λ, the difference between q_1 and q_2 must be at least p.

Since $|u_\lambda| \le |s_\lambda|\,/\beta$ and σ is α-gapped, $1 \le q \le |s_\lambda|\,\alpha/\beta \le |r_\lambda|\,\alpha/\beta$. Then the number of possible periods q is bounded by $|r_\lambda|\,\alpha/(\beta p) = \exp(r_\lambda)\alpha/\beta$. Therefore the number of maximal α-gapped repeats is bounded by $\alpha E(w)/\beta$ for the case $\mathsf{b}(u_\lambda) = \mathsf{b}(r_\lambda)$. Summing up we get the bound $2\alpha E(w)/\beta$. ◀

Remembering the results of Section 2.1, we map gapped repeats to their respective points. By using the period as the y-coordinate, one can show Lemma 7.

▶ **Lemma 7.** *Given a word w, and two real numbers $\alpha > 1$ and $2/3 \le \beta < 1$. The points mapped by two different maximal gapped repeats in $\overline{\beta\mathcal{P}}_\alpha(w)$ cannot $\frac{1-\beta}{\alpha}$-cover the same point.*

Proof. Let $\sigma = u_\lambda, v, u_\rho$ and $\overline{\sigma} = \overline{u_\lambda}, \overline{v}, \overline{u_\rho}$ be two different maximal gapped repeats in $\overline{\beta\mathcal{P}_\alpha(w)}$. Set $u := |u_\lambda| = |u_\rho|$, $\overline{u} := |\overline{u_\lambda}| = |\overline{u_\rho}|$, $q := |u_\lambda v|$ and $\overline{q} := |\overline{u_\lambda v}|$. We map the maximal gapped repeats σ and $\overline{\sigma}$ to the points $(\mathsf{e}(u_\lambda), q)$ and $(\mathsf{e}(\overline{u_\lambda}), \overline{q})$, respectively. Assume, for the sake of contradiction, that both points $\frac{1-\beta}{\alpha}$-cover the same point (x, y).

Let $z := |\mathsf{e}(u_\lambda) - \mathsf{e}(\overline{u_\lambda})|$ be the difference of the endings of both left arms, and $s_\lambda := w[[\mathsf{b}(u_\lambda), \mathsf{e}(u_\lambda)] \cap [\mathsf{b}(\overline{u_\lambda}), \mathsf{e}(\overline{u_\lambda})]]$ be the overlap of u_λ and $\overline{u_\lambda}$. Let $s := |s_\lambda|$, and let s_ρ (resp. $\overline{s_\rho}$) be the right copy of s_λ based on σ (resp. $\overline{\sigma}$).

Sub-Claim: The overlap s_λ is not empty, and $s_\rho \neq \overline{s_\rho}$

Sub-Proof. Assume for this sub-proof that $\mathsf{e}(u_\lambda) < \mathsf{e}(\overline{u_\lambda})$ (otherwise exchange σ with $\overline{\sigma}$, or yield the contradiction $\sigma = \overline{\sigma}$). By combining the $(1-\beta)/\alpha$-cover property with the fact that $\overline{\sigma}$ is α-gapped, we yield $\mathsf{e}(\overline{u_\lambda}) - \overline{u} \leq \mathsf{e}(\overline{u_\lambda}) - \overline{q}/\alpha \leq \mathsf{e}(\overline{u_\lambda}) - \overline{q}(1-\beta)/\alpha \leq x \leq \mathsf{e}(u_\lambda) < \mathsf{e}(\overline{u_\lambda})$. So the subword $w[\mathsf{e}(u_\lambda)]$ is contained in $\overline{u_\lambda}$. If $s_\rho = \overline{s_\rho}$, then we get a contradiction to the maximality of σ: By the above inequality, $w[\mathsf{e}(u_\lambda) + 1]$ is contained in $\overline{u_\lambda}$, too. Since $\overline{\sigma}$ is a gapped repeat, the character $w[\mathsf{e}(u_\lambda) + 1]$ occurs in $\overline{u_\rho}$, exactly at $w[\mathsf{e}(u_\rho) + 1]$. ◀

So $q \neq \overline{q}$. Without loss of generality let $q < \overline{q}$. Then

$$\overline{q} - \frac{\overline{q}(1-\beta)}{\alpha} \leq y \leq q \leq \overline{q}. \tag{1}$$

$$\text{So the difference of both periods is } 0 < \delta := \overline{q} - q \leq \overline{q}(1-\beta)/\alpha \leq \overline{u}(1-\beta). \tag{2}$$

$$\text{Eq. (1) also yields that } u \geq q/\alpha \geq \frac{\overline{q}}{\alpha}(1 - \frac{1-\beta}{\alpha}) \geq \overline{q}\beta/\alpha. \tag{3}$$

Since $s_\rho = [\mathsf{b}(s_\lambda) + q, \mathsf{e}(s_\lambda) + q]$ and $\overline{s_\rho} = [\mathsf{b}(s_\lambda) + \overline{q}, \mathsf{e}(s_\lambda) + \overline{q}]$, we have $\mathsf{b}(\overline{s_\rho}) - \mathsf{b}(s_\rho) = \delta$.

By case analysis, we show that u_λ or $\overline{u_\lambda}$ has a periodic prefix, which leads to the contradiction that σ or $\overline{\sigma}$ are in $\beta\mathcal{P}_\alpha(w)$.

1. Case: $\mathsf{e}(u_\lambda) \leq \mathsf{e}(\overline{u_\lambda})$. Since $\mathsf{e}(\overline{u_\lambda}) - \overline{q}(1-\beta)/\alpha \leq x \leq \mathsf{e}(u_\lambda) \leq \mathsf{e}(\overline{u_\lambda})$,

$$z = \mathsf{e}(\overline{u_\lambda}) - \mathsf{e}(u_\lambda) \leq \overline{q}(1-\beta)/\alpha \leq \overline{u}(1-\beta). \tag{4}$$

1a. Sub-Case: $\mathsf{b}(u_\lambda) \leq \mathsf{b}(\overline{u_\lambda})$. By Eq. (4), we get $s = \overline{u} - z \geq \overline{u}\beta$. It follows from Eq. (2) and $2/3 \leq \beta < 1$ that $s/\delta \geq \overline{u}\beta/\overline{u}(1-\beta) = \beta/(1-\beta) \geq 2$, which means that s_ρ and $\overline{s_\rho}$ overlap at least half of their common length, so s_λ is periodic. Since s_λ is a prefix of $\overline{u_\lambda}$ of length $s \geq \overline{u}\beta$, $\overline{\sigma}$ is in $\beta\mathcal{P}_\alpha(w)$, a contradiction.

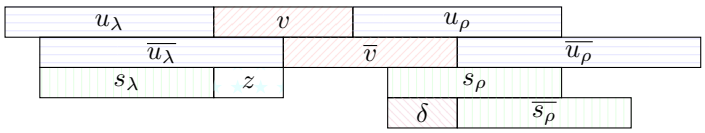

Figure 1 Sub-Case 1a.

1b. Sub-Case: $\mathsf{b}(u_\lambda) > \mathsf{b}(\overline{u_\lambda})$. We conclude that $s_\lambda = u_\lambda$. It follows from Eqs. (2) and (3) and $2/3 \leq \beta < 1$ that $s/\delta \geq \overline{q}\alpha\beta/(\overline{q}\alpha(1-\beta)) = \beta/(1-\beta) \geq 2$, which means that $s_\lambda = u_\lambda$ is periodic. Hence σ is in $\beta\mathcal{P}_\alpha(w)$, a contradiction.

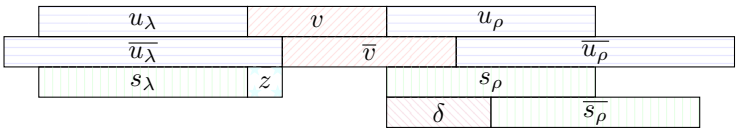

Figure 2 Sub-Case 1b.

2. Case: $\mathsf{e}(u_\lambda) > \mathsf{e}(\overline{u_\lambda})$. Since $\mathsf{e}(u_\lambda) - q(1-\beta)/\alpha \le x \le \mathsf{e}(\overline{u_\lambda}) \le \mathsf{e}(u_\lambda)$,

$$z = \mathsf{e}(u_\lambda) - \mathsf{e}(\overline{u_\lambda}) \le q(1-\beta)/\alpha \le \overline{q}(1-\beta)/\alpha \le \overline{u}(1-\beta). \tag{5}$$

2a. Sub-Case: $\mathsf{b}(u_\lambda) \le \mathsf{b}(\overline{u_\lambda})$. We conclude that $s_\lambda = \overline{u_\lambda}$. It follows from Eq. (2) and $2/3 \le \beta < 1$ that $s/\delta \ge \overline{u}/(\overline{u}(1-\beta)) = 1/(1-\beta) \ge 3 > 2$, which means that $s_\lambda = \overline{u_\lambda}$ is periodic. Hence $\overline{\sigma}$ is in $\beta\mathcal{P}_\alpha(w)$, a contradiction.

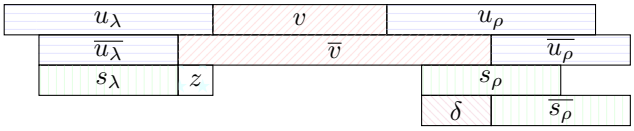

Figure 3 Sub-Case 2a.

2b. Sub-Case: $\mathsf{b}(u_\lambda) > \mathsf{b}(\overline{u_\lambda})$. By Eq. (5) we get $z \le q(1-\beta)/\alpha \le u(1-\beta)$ and hence $s = u - z \ge u\beta$. If $\delta \le s/2$, s_ρ and $\overline{s_\rho}$ overlap at least half of their common length, which leads to the contradiction that u_λ has a periodic prefix s_λ of length at least $u\beta$. Otherwise, let us assume that $s/2 < \delta$. By Eqs. (2) and (3) we get $u/\delta \ge \overline{q}\alpha\beta/(\overline{q}\alpha(1-\beta)) = \beta/(1-\beta) \ge 2$ with $2/3 \le \beta < 1$. Hence, δ is upper bounded by $u/2$; so u_ρ has a periodic prefix of length at least 2δ (since $2\delta > s \ge u\beta$), a contradiction.

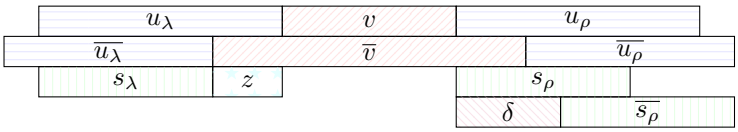

Figure 4 Sub-Case 2b.

◀

The next lemma follows immediately from Lemmas 5 and 7.

▶ **Lemma 8.** *For $\alpha > 1$, $2/3 \le \beta < 1$ and a word w of length n, $\left|\overline{\beta\mathcal{P}}_\alpha(w)\right| < 3\alpha n/(1-\beta)$.*

▶ **Theorem 9.** *Given a word w of length n, and a real number $\alpha > 1$. Then $|\mathcal{G}_\alpha(w)| < 18\alpha n$.*

Proof. Combining the results of Lemmas 6 and 8, $|\mathcal{G}_\alpha(w)| = |\beta\mathcal{P}_\alpha(w)| + \left|\overline{\beta\mathcal{P}}_\alpha(w)\right| < 2\alpha E(w)/\beta + 3\alpha n/(1-\beta)$ for $2/3 \le \beta < 1$. Applying Lemma 1, the term is upper bounded by $6\alpha n/\beta + 3\alpha n/(1-\beta)$. The number is minimal for $\beta = 2/3$, yielding the bound $18\alpha n$. ◀

With Corollary 2 we obtain the result of Theorem 9 for $\alpha \ge 1$.

We can bound the number of maximal α-gapped palindromes by similar proofs to $28\alpha n + 7n$. This bound solves an open problem in [15], where Kolpakov and Kucherov

conjectured that the number of α-gapped palindromes with $\alpha \geq 2$ in a string is linear. We briefly explain the main differences and similarities needed to understand the relationship between gapped repeats and palindromes. Let σ be a maximal α-gapped repeat (or α-gapped palindrome). If the gapped repeat (palindrome) has a periodic prefix s_λ generated by some run, the right arm has a periodic prefix (suffix) s_ρ generated by a run of the same period. Since σ is maximal, both runs have to obey constraints that are similar in both cases, considering whether σ is a gapped repeat or a gapped palindrome. So it is easy to change the proof of Lemma 6 in order to work with palindromes. Like with aperiodic gapped repeats, we can apply the point analysis to the aperiodic α-gapped palindromes, too. Our main idea is to map a gapped palindrome u_λ, v, u_ρ injectively to the pair of integers $(\mathsf{e}(u_\lambda), |v|)$, exchanging the period with the size of the gap. Details are provided in the full version of the paper [9].

3 Finding All Maximal α-gapped Repeats

The computational model we use to design and analyse our algorithms is the standard unit-cost RAM with logarithmic word size, which is generally used in the analysis of algorithms. In the upcoming algorithmic problems, we assume that the words we process are sequences of integers. In general, if the input word has length n then we assume its letters are in $\{1, \ldots, n\}$, so each letter fits in a single memory-word. This is a common assumption in stringology (see, e.g., the discussion in [12]). For a word w, $|w| = n$, we build in $\mathcal{O}(n)$ time the suffix array as well as data structures allowing us to retrieve in constant time the length of the longest common prefix of any two suffixes $w[i, n]$ and $w[j, n]$ of w, denoted $LCP_w(i, j)$ (the subscript w is omitted when there is no danger of confusion). In what follows, such structures are called *LCP* data structures (see, e.g., [12, 11]). We begin with a simple lemma.

▶ **Lemma 10.** *Given a word w, $|w| = n$, we can process it in $\mathcal{O}(n)$ time such that, for each $i, p \leq n$, we can return in $\mathcal{O}(1)$ time the longest factor of period p starting at position i in w.*

Let w be a word and v be a factor of w with $per(v) = p$. Further, let z be a subword of length $\ell|v|$ of w. An occurrence of v in z is a subword $(v, [i, i + |v| - 1])$ of z; we say that v ***occurs*** at position i in z. For an easier presentation of our algorithm, we distinguish between two types of occurrences of v in z. On the one hand, we have the so-called ***single occurrences***. If v is aperiodic, then all its occurrences in z are single occurrences; there are $\mathcal{O}(\ell)$ such occurrences (see, e.g., [13]). If v is periodic, then a subword $(v, [i, i + |v| - 1])$ of z starting at position i in z is a single occurrence if v occurs neither at position $i - p$ nor at position $i + p$ in z. On the other hand, we have ***occurrences of v within a run*** of z, whose period is $p = per(v)$. That is, the subword $(v, [i, i + |v| - 1])$ starting at position i in z is an occurrence of v within a run if v occurs either at $i - p$ or at $i + p$. We say that $(v, [i, i + |v| - 1])$ is the ***first occurrence*** of v in a run of period p of z if v does not occur at $i - p$ but occurs at $i + p$. Note that there are $\mathcal{O}(\ell)$ runs containing occurrences of v in z, or, equivalently, $\mathcal{O}(\ell)$ first occurrences of v in runs of period p.

Consequently, the occurrences of v in z can be succinctly represented as follows. For the single occurrences we just store their starting position. The occurrences of v in a run r can be represented by the starting position of the first occurrence of v in r, together with the period of v, since the starting positions of the occurrences of v in r form an arithmetic progression of period p.

In our approach, basic factors (i.e., factors of length 2^k, for $k \geq 1$) of the input word are important. For some integer $c \geq 2$, the occurrences of the basic factor $w[i, i + 2^k - 1]$ in a subword of length $c2^k$ can be represented in a compact manner: $\mathcal{O}(c)$ positions of the single

occurrences of $w[i, i + 2^k - 1]$ and $\mathcal{O}(c)$ first occurrences of $w[i, i + 2^k - 1]$ in runs, together with the period of $w[i, i + 2^k - 1]$. We need the next lemma (see [10, 13]).

▶ **Lemma 11.** *Given a word w of length n and an integer $c \geq 2$, we can process w in time $\mathcal{O}(n \log n)$ such that given any basic factor $y = w[i, i + 2^k - 1]$ and any subword of w $(z, [j, j + c2^k - 1])$, with $k \geq 0$, we can compute in $\mathcal{O}(\log \log n + c)$ time the representation of all the (single and within runs) occurrences of y in z.*

We now focus on short basic factors of words. The constant 16 occurring in the following considerations can be replaced by any other constant; we just use it here so that we can apply these results directly in the main proofs of this section.

Given a word v and some integer $\beta \geq 16$ with $|v| = \beta \log n$, as well as a basic factor $y = v[i2^k + 1, (i + 1)2^k]$, with $i, k \geq 0$ and $i2^k + 1 > (\beta - 16) \log n$ (so occurring in the suffix of length $16 \log n$ of v), the occurrences of y in v can be represented as $\mathcal{O}(\beta)$ bit-sets, each containing $\mathcal{O}(\log n)$ bits, the 1-bits marking the starting positions of the occurrences of y in v. The next result can be shown using tools developed in [8] (see also [10]).

▶ **Lemma 12.** *Given a word v and an integer $\beta > 16$, with $|v| = \beta \log n$, we can process v in $\mathcal{O}(\beta \log n)$ time such that given any basic factor $y = v[i2^k + 1, (i + 1)2^k]$ with $i, k \geq 0$ and $i2^k + 1 > (\beta - 16) \log n$, we can find in $\mathcal{O}(\beta)$ time the $\mathcal{O}(\beta)$ bit-sets, each storing $\mathcal{O}(\log n)$ bits, characterizing all the occurrences of y in v.*

In the context of the previous lemma, once the occurrences of y in v are computed, given a subword z of v of length $|z| = c|y|$, for some $c \geq 1$, we can obtain in $\mathcal{O}(c)$ time both the single occurrences of y in z and the occurrences of y within runs of z. We just have to select (by bitwise operations on the bit-sets encoding the factors of v that overlap z) the positions where y occurs (so the positions of the 1-bits in those bit-sets). For each two consecutive such occurrences of y we detect whether they are part of a run in v and then skip over all the occurrences of y from that run (and the corresponding parts of the bit-sets) before looking again for 1-bits in the bit-sets; for the positions that form a run we store the first occurrence of y and its period, while for the single occurrences we store the position of that occurrence.

Now we can begin the presentation of the algorithm finding all the maximal α-gapped repeats of a word. We first show how to find maximal repeats with short arms.

▶ **Lemma 13.** *Given a word w and $\alpha \geq 1$, we can find all the maximal α-gapped repeats u_λ, u', u_ρ occurring in w, with $|u_\rho| \leq 16 \log n$, in time $\mathcal{O}(\alpha n)$.*

Proof. If a maximal α-gapped repeat u_λ, u', u_ρ (where we denote by u the underlying factor of both arms) has $|u| \leq 16 \log n$, we get that u_ρ must be completely contained in a subword $(w', [m \log n + 1, (m + 17) \log n])$, for some m with $\frac{n}{\log n} - 17 \geq m \geq 0$. By fixing the interval where u_ρ may occur (that is, fix m), we also fix the place where u_λ may occur. Indeed, the entire subword u_λ, u', u_ρ is completely contained in the factor $x_m = (w'', [(m - 16\alpha) \log n + 1, (m + 17) \log n])$ (or, in the factor $x_m = (w'', [1, (m + 17) \log n])$ if $(m - 16\alpha) \log n + 1 < 1$).

Hence, we look for maximal α-gapped repeats u_λ, u', u_ρ completely contained in x_m with u_ρ completely contained in the suffix of length $16 \log n$ of x_m; then we repeat this process for all m. To begin with, we process x_m as in Lemma 12, and construct *LCP*-structures for it.

Now, once we fixed the subword x_m of w where we search the maximal α-gapped repeats, we try to fix also their length. That is, we find all maximal α-gapped repeats u_λ, u', u_ρ with $2^{k+1} \leq |u| \leq 2^{k+2}$ completely contained in x_m with u_ρ completely contained in the suffix of length $16 \log n$ of x_m; we execute this process for all $0 \leq k \leq \log(16 \log n)$. Note that all the

maximal α-gapped repeats with arms shorter than 2 (occurring anywhere in the word w) can be trivially found in $\mathcal{O}(\alpha n)$ time.

Since we can find occurrences of basic factors in x_m efficiently, we try to build a maximal gapped repeat by extending gapped repeats whose arms contain a basic factor. To this end, we analyse some *subwords of* x_m: If $2^{k+1} \leq |u| \leq 2^{k+2}$ then u_ρ contains at least one subword $(y, [j2^k + 1, (j+1)2^k])$ starting within its first 2^k positions. A copy of the factor y occurs also within the first 2^k positions of u_λ (with the same offset with respect to the starting position of u_λ as the offset of the occurrence of y with respect to the starting position of u_ρ). So, finding the respective copy of y from u_λ helps us discover the place where u_λ actually occurs. Indeed, assume that we identified the copy of y from u_λ, and assume that this copy is $(y, [\ell + 1, \ell + |y|])$; we try to build u_λ and u_ρ around these two occurrences of y, respectively. Hence, in order to identify u_λ and u_ρ we compute the longest factor p of x_m that ends both at $j2^k$ and at ℓ and the longest factor s that starts both at $(j+1)2^k + 1$ and at $\ell + |y| + 1$. Now, if $\ell + |y| + |s| \leq j2^k - |p|$ then u_λ is obtained by concatenating p and s around $x_m[\ell + 1, \ell + |y|]$ while u_ρ is obtained by concatenating p and s to the left and, respectively, right of $x_m[j2^k + 1, (j+1)2^k]$; otherwise, the two occurrences of y do not determine a maximal repeat. Moreover, the repeat we determined is a valid solution of our problem only if its right arm contains position $j2^k + 1$ of x_m within its first 2^k positions.

Now we explain how to determine efficiently the copy of y around which we try to build u_λ. As $|u| < 2^{k+2}$ and $|y| = 2^k$ we get that the copy of y that corresponds to u_λ should be completely contained in the subword of x_m of length $\alpha 2^{k+2}$ ending at position $j2^k$. As said above, we already processed x_m to construct the data structures from Lemma 12. Therefore, we can obtain in $\mathcal{O}(\alpha)$ time a representation of all the occurrences of y inside the factor of length $\alpha 2^{k+2}$ ending at position $j2^k$. These occurrences can be single occurrences and occurrences within runs. There are $\mathcal{O}(\alpha)$ single occurrences, and we can process each of them individually, as explained, to find the maximal α-gapped repeat they determine together with the occurrence of y from u_ρ. However, it is not efficient to do the same for the occurrences of y within runs. For these (which are also $\mathcal{O}(\alpha)$ many) we proceed as follows.

Assume we have a repetition of y's inside the factor of x_m of length $\alpha 2^{k+2}$ ending at position $j2^k$. Let ℓ be the starting position of the first occurrence of y in this repetition and let p be the period of y. Now, using Lemma 10 we can determine the maximal p-periodic subword (a run of period p) r_λ of x_m containing this repetition of y-occurrences. Similarly, we can determine the maximal p-periodic subword (a run of period p) r_ρ that contains the occurrence of y from u_ρ (i.e., $x_m[j2^k + 1, (j+1)2^k]$). To determine efficiently the α-gapped repeats that contain $x_m[j2^k + 1, (j+1)2^k]$ in the right arm and a corresponding occurrence of y from r_λ in the left arm we analyse several cases.

Assume u_ρ starts at a position of r_ρ, other than its first one. Then u_λ should also start at the first position of r_λ (or we could extend both arms to the left, a contradiction to the maximality of the repeat). If u_ρ ends at a position to the right of r_ρ, then u_λ also ends at a position to the right of r_λ, and, moreover, the suffix of u_λ occurring after the end of r_λ and the suffix of u_ρ occurring after the end of r_ρ are equal, and can be computed by a longest common prefix query on x_m. This means that u_λ can be determined exactly (we know where it starts and where it ends) so u_ρ can also be determined exactly (we know where it ends), and we can check if the obtained repeat is indeed a maximal α-gapped repeat, and the arms fulfil the required length conditions (i.e., their length is between 2^{k+1} and 2^{k+2}, the right arm contains position $j2^k + 1$ of x_m within its first 2^k positions). If u_ρ ends exactly at the same position as r_ρ, then u_ρ is periodic of period p. We compute the longest p-periodic prefix u' of r_λ which is also a suffix of r_ρ. Since u_λ is longer than p, the α-gapped repeats under consideration have

the left arm $u_\lambda := r_\lambda[1..|u'| - pi]$ and the right arm $u_\rho := r_\rho[|r_\rho| - (|u'| - pi) + 1..|r_\rho|]$ for $i \geq 0$ such that the gap $v := (w, [\mathsf{e}(u_\lambda) + 1, \mathsf{b}(u_\rho) - 1])$ respects the condition $|u_\lambda v| \leq \alpha\,|u_\lambda|$. Clearly, we can output in $\mathcal{O}(1)$ time each such repeat.

The final case is when u_ρ ends at a position of r_ρ, other than its last position. In that case, we get that $u_\lambda = r_\lambda$ (or, otherwise, we could extend both arms to the right). Essentially, this means that we know exactly where u_λ is located and its length (and we continue only if this length is between 2^{k+1} and 2^{k+2}); so u_λ denotes a factor $z^h z'$ for some z of length p. Now, looking at the run r_ρ, we can get easily the position of the first occurrence of z in that run, and the position of its last occurrence. If the first occurrence is ℓ', then the occurrences of z have their starting positions ℓ', $\ell' + p, \ldots, \ell' + tp$ for some t. As we know the length of u_λ and the fact that u_λ, u', u_ρ is α-gapped, we can determine in constant time the values $0 \leq i \leq t$ such that u_ρ may start at position $\ell' + ip$, the repeat we obtain is α-gapped, and u_ρ contains position $j2^k + 1$ of x_m within its first 2^k positions. If u_ρ is a prefix of r_ρ we also have to check that we cannot extend simultaneously u_ρ and u_λ to the left; if u_ρ is a suffix of r_λ we have to check that we cannot extend simultaneously u_ρ and u_λ to the right. Then we can return the maximal α-gapped repeats we constructed.

The cases when u_ρ starts at the first position of r_ρ or when it starts at a position to the left of r_ρ can be treated similarly, and as efficiently.

This concludes our algorithm. Its correctness follows from the explanations above. Moreover, we can ensure that our algorithm finds and outputs each maximal repeat exactly once; this clearly holds when we analyse the repeats of x_m for each m separately. However, when moving from x_m to x_{m+1} we must also check that the right arm of each repeat we find is not completely contained in x_m (so, already found). This condition can be easily imposed in our search: when constructing the arms determined by a single occurrence of y, we check the containment condition separately; when constructing a repeat determined by a run of y-occurrences, we have to impose the condition that the right arm extends out of x_m when searching the starting positions of the possible arms.

Next, we compute the complexity of the algorithm. Once we fix m, k, and j, our process takes $\mathcal{O}(\alpha + N_{j,m,k})$ time, where $N_{j,m,k}$ is the number of maximal α-gapped repeats determined for the fixed m, j, k. So, the time complexity of the algorithm is:
$\mathcal{O}(n + \sum_{0 \leq m \leq n/\log n}(16\alpha \log n + \sum_{0 \leq k \leq \log(16 \log n)}(\sum_{j \leq 16 \log n/2^k}(\alpha + N_{j,m,k})))) = \mathcal{O}(\alpha n)$,
as the total number of maximal α-gapped repeats is $\mathcal{O}(\alpha n)$ and we need $\mathcal{O}(|x_m|)$ preprocessing time for each x_m and $\mathcal{O}(n)$ preprocessing time for w. ◀

Next, we find all maximal α-gapped repeats with longer arms.

▶ **Lemma 14.** *Given a word w and $\alpha \geq 1$, we can find all the maximal α-gapped repeats u_λ, u', u_ρ occurring in w, with $|u_\rho| > 16 \log n$, in time $\mathcal{O}(\alpha n)$.*

Proof. The general approach in proving this lemma is similar to that used in the proof of the previous result. Essentially, when identifying a new maximal α-gapped repeat, we try to fix the place and length of the right arm u_ρ of the respective repeat, which restricts the place where the left arm u_λ occurs. This allows us to fix some long enough subword of w as being part of the right arm, detect its occurrences that are possibly contained in the left arm, and, finally, to efficiently identify the actual repeat. The main difference is that we cannot use the result of Lemma 12, as we have to deal with repeats with arms longer than $16 \log n$. Instead, we will use the structures constructed in Lemma 11. However, to get the stated complexity, we cannot apply this lemma directly to the word w, but rather to an encoded variant of w.

Thus, the first step of the algorithm is to construct a word w', of length $\frac{n}{\log n}$, whose symbols, called ***blocks***, encode $\log n$ consecutive symbols of w grouped together. That is, the

first block of the new word corresponds to $w[1, \log n]$, the second one to $w[\log n + 1, 2 \log n]$, and so on. Hence, we have two versions of the word w: the original one, and the one where it is split in blocks. It is not hard to see that the blocks can be encoded into numbers between 1 and n in linear time. Indeed, we build the suffix array and LCP-data structures for w, and then we cluster together the suffixes of the suffix array that share a common prefix of length at least $\log n$. Then, all the suffixes of a cluster are given the same number (between 1 and n), and a block is given the number of the suffix starting with the respective block.

We can now construct in $\mathcal{O}(n)$ time the suffix arrays and LCP-data structures for both w and w', as well as the data structures of Lemma 11 for the word w'.

Now, we guess the length of the arms of the repeat. We try to find the maximal α-gapped repeats u_λ, u', u_ρ of w with $2^{k+1} \log n \leq |u_\lambda| \leq 2^{k+2} \log n$, $k \leq \log \frac{n}{\log n} - 2$. We fix k and split again the word w, this time in factors of length $2^k \log n$, called ***k*-blocks**. Assume that each split is exact (padding the word with some new symbols ensures this).

Now, if a maximal α-gapped repeat u_λ, u', u_ρ with $2^{k+1} \log n \leq |u_\lambda| \leq 2^{k+2} \log n$ exists, then it contains an occurrence of a k-block within its first $2^k \log n$ positions. So, let z be a k-block and assume that it is the first k-block occurring in u_ρ (in this way fixing a range where u_ρ may occur). Obviously, if u_ρ contains z, then u_λ also contains an occurrence of z; however, this occurrence is not necessarily starting at a position $j \log n + 1$ for some $j \geq 0$ (so, it is not necessarily a sequence of blocks). But, at least one of the factors of length $2^{k-1} \log n$ starting within the first $\log n$ positions of z (which are not necessarily sequences of blocks) must correspond, in fact, to a sequence of blocks from the left arm u_λ. So, let us fix now a factor y of length $2^{k-1} \log n$ that starts within the first $\log n$ positions of z (we try all of them in the algorithm, one by one). As said, the respective occurrence of y from u_ρ is not necessarily a sequence of blocks (so it cannot be mapped directly to a factor of w'). But, we look for an occurrence of y starting at one of the $\alpha 2^{k+2} \log n$ positions to the left of z, corresponding to a sequence of blocks, and assume that the respective occurrence is exactly the occurrence of y from u_λ.

By binary searching the suffix array of w' (using LCP-queries on w to compare the factors of $\log n$ symbols of y and the blocks of w', at each step of the search) we try to detect a factor of w' that encodes a word equal to y. Assume that we can find such a sequence y' of 2^{k-1} blocks of w' (otherwise, y cannot correspond to a sequence of blocks from u_λ, so we should try other factors of z instead). Using Lemma 11 for w', we get in $\mathcal{O}(\log \log |w'| + \alpha)$ time a representation of the occurrences of y' in the range of $\alpha 2^{k+2}$ blocks of w' occurring before the blocks of z; this range corresponds to an interval of w with a length of $\alpha 2^{k+2} \log n$.

Further, we process these occurrences of y' just like in the previous lemma. Namely, the occurrences of y' in that range are either single occurrences or occurrences within runs. Looking at their corresponding factors from w, we note that each of these factors fixes a possible left arm u_λ; this arm, together with the corresponding arm u_ρ can be constructed just like before. In the case of single occurrences (which are at most $\mathcal{O}(\alpha)$, again), we try to extend both the respective occurrence and the occurrence of y from u_ρ both to the left and, respectively, to the right, simultaneously, and see if we can obtain in this way the arms of a valid maximal α-gapped repeat. Note that we must check also that the length of the arm of the repeat is between 2^{k+1} and 2^{k+2}, and that z is the first k-block of the right arm. As before, complications occur when the occurrences of y' are within runs. In this case, the run of occurrences of y' does not necessarily give us the period of y, but a multiple of this period that can be expressed also as a multiple of $\log n$ (or, in other words, the minimum period of y is a multiple of the block-length). This, however, does not cause any problems, as the factor y from u_ρ should always correspond to a block sequence from u_λ, so definitely to one of the factors encoded in the run of occurrences of y'.

Therefore, by determining the maximal factor that contains y and has the same period as the repetition of y'-occurrences (with the period measured within w), we can perform a very similar analysis to the corresponding one from the case when we searched maximal α-gapped repeats with arms shorter than $16 \log n$.

It remains to prove that each maximal gapped repeat is counted only once. Essentially, the reason for this is that for two separate factors y_1 and y_2 (of length $2^{k-1} \log n$) occurring in the first $\log n$ symbols of z we cannot get occurrences of the corresponding factors y_1' and y_2' that define the same repeat; in that case, the distance between y_1' and y_2' should be at least one block, so the distance between y_1 and y_2 should be at least $\log n$, a contradiction. Similarly, if we have a factor y occurring in the first $\log n$ symbols of some k-block z_1 such that this factor determines an α-gapped maximal repeat, then the same maximal repeat cannot be determined by a factor of another k-block, since z_1 is the first k-block of u_ρ.

The correctness of the algorithm described above follows easily from the explanations given in the proofs of the last two lemmas. Let us evaluate its complexity. The preprocessing phase (construction of w' and of all the needed data structures) takes $\mathcal{O}(n)$ time. Further, we can choose k (and implicitly an interval for the length of the arms of the repeats) such that $k \leq \log \frac{n}{\log n} - 2$. After choosing k, we can choose a k-block z in $\frac{n}{2^k \log n}$ ways. Further, we analyse each factor y of length $2^{k-1} \log n$ starting within the first $\log n$ positions of the chosen k-block z. For each such factor y we find in $\mathcal{O}(\log \frac{n}{\log n} + \log\log n + \alpha)$ time the representation of the occurrences of the block encoding the occurrence of y from u_λ. From each of the $\mathcal{O}(\alpha)$ single occurrences we check whether it is possible to construct a maximal α-gapped repeat in $\mathcal{O}(1)$ time. We also have $\mathcal{O}(\alpha)$ occurrences of the block encoding y in runs, and each of them is processed in $\mathcal{O}(N_{z,y})$ time, where $N_{z,y}$ is the number of maximal α-gapped repeats we find for some z and y. Overall, this adds up to a total time of $\mathcal{O}(n \log n + \alpha n)$, as the total number of maximal α-gapped repeats in w is upper bounded by $\mathcal{O}(\alpha n)$. If $\alpha \geq \log n$, the statement of the lemma follows. If $\alpha < \log n$, we proceed as follows.

Initially, we run the algorithm only for $k > \log\log n$ and find the maximal α-gapped repeats $u_\lambda u' u_\rho$ with $2^{\log\log n} \log n \leq |u_\lambda|$, in $\mathcal{O}(\alpha n)$ time. Further, we search maximal α-gapped repeats with shorter arms. Now, $|u_\lambda|$ is upper bounded by $2^{\log\log n + 1} \log n = 2(\log n)^2$, so $|u_\lambda u' u_\rho| \leq \ell_0$, for $\ell_0 = \alpha \cdot 2(\log n)^2 + 2(\log n)^2 = 2(\alpha+1)(\log n)^2$. Such an α-gapped repeat $u_\lambda u' u_\rho$ is, thus, contained in (at least) one factor of length $2\ell_0$ of w, starting at a position of the form $1 + m\ell_0$ for $m \geq 0$. So, we take the factors $w[1 + m\ell_0, (m+2)\ell_0]$ of w, for $m \geq 0$, and apply for each such factor, separately, the same strategy as above to detect the maximal α-gapped repeats contained completely in each of them. To this end, we first encode the factors $w[1 + m\ell_0, (m+2)\ell_0]$ of w so that each of them corresponds to an integer in $[0, 2\ell_0]$; we attain the encoding of the factors $w[1 + m\ell_0, (m+2)\ell_0]$ by sorting the symbols of w in linear time, and then using their order. The total time needed to do that is $\mathcal{O}\Big(n + \alpha\ell_0 \frac{n}{\ell_0} + N_{\ell_0}\Big) = \mathcal{O}(\alpha n)$, where N_{ℓ_0} is the number of repeats we find; moreover, we can easily ensure that a maximal repeat is not output twice (that is, ensure always that the gapped repeats we produce were not already contained in a previously processed interval). Hence, we find all maximal α-gapped repeats $u_\lambda u' u_\rho$ with $2^{\log\log(2\ell_0)} \log(2\ell_0) \leq |u|$. This means we find all the maximal α-gapped repeats with $|u| \geq 2^{\log\log(2\ell_0)+1} \log(2\ell_0)$. Since $2^{\log\log(2\ell_0)+1} \log(2\ell_0) \leq 16 \log n$ (for n large enough, as $\alpha \leq \log n$), we can apply Lemma 13 for gapped repeats with an arm-length smaller than $2^{\log\log(2\ell_0)+1} \log(2\ell_0)$. ◀

Putting together the results of Lemmas 13 and 14 we get the following theorem.

▶ **Theorem 15.** *Given a word w and $\alpha \geq 1$, we can compute $\mathcal{G}_\alpha(w)$ in time $\mathcal{O}(\alpha n)$.*

By a completely similar approach we can compute $\mathcal{G}^{\intercal}_{\alpha}(w)$, generalizing the algorithm of [15]. To this end, we construct *LCP*-structures for $ww^{\intercal}$ (allowing us to test efficiently whether a factor $w[i,j]^{\intercal}$ occurs at some position in w). When we search the α-gapped palindromes u_λ, v, u_ρ (with $u_\rho \equiv u_\lambda{}^{\intercal}$), we split again w in blocks and k-blocks, for each $k \leq \log|w|$, to check whether there exists such an u_λ, v, u_ρ with $2^k \leq |u_\lambda| \leq 2^{k+1}$. This search is conducted pretty much as in the case of repeats, only that now when we fix some factor y of u_ρ, we have to look for the occurrences of $y^{\intercal}$ in the factor of length $\mathcal{O}(\alpha|u_\rho|)$ preceding it; the *LCP*-structures for $ww^{\intercal}$ are useful for this, because, as explained above, they allow us to efficiently search the mirror images of factors of w inside w. Thus, given a word w and $\alpha \geq 1$, we can compute $\mathcal{G}^{\intercal}_{\alpha}(w)$ in time $\mathcal{O}(\alpha n)$.

Acknowledgement. The work of Florin Manea was supported by the DFG grant 596676.

References

1 Golnaz Badkobeh, Maxime Crochemore, and Chalita Toopsuwan. Computing the maximal-exponent repeats of an overlap-free string in linear time. In *SPIRE 2012. Proceedings*, volume 7608 of *Lecture Notes in Computer Science*, pages 61–72. Springer, 2012.

2 Hideo Bannai, Tomohiro I, Shunsuke Inenaga, Yuto Nakashima, Masayuki Takeda, and Kazuya Tsuruta. The Runs Theorem. *CoRR*, abs/1406.0263, 2014. URL: `http://arxiv.org/abs/1406.0263`.

3 Gerth Stølting Brodal, Rune B. Lyngsø, Christian N. S. Pedersen, and Jens Stoye. Finding maximal pairs with bounded gap. In *CPM*, volume 1645 of *LNCS*, pages 134–149. Springer, 1999.

4 Maxime Crochemore, Costas S. Iliopoulos, Marcin Kubica, Wojciech Rytter, and Tomasz Walen. Efficient algorithms for two extensions of LPF table: The power of suffix arrays. In *Proc. SOFSEM 2010*, volume 5901 of *LNCS*, pages 296–307, 2010.

5 Maxime Crochemore, Roman Kolpakov, and Gregory Kucherov. Optimal searching of gapped repeats in a word. *ArXiv e-prints 1309.4055*, 2015. `arXiv:1309.4055`.

6 Maxime Crochemore and German Tischler. Computing longest previous non-overlapping factors. *Inf. Process. Lett.*, 111(6):291–295, February 2011.

7 Marius Dumitran and Florin Manea. Longest gapped repeats and palindromes. In *Proc. MFCS 2015*, volume 9234 of *LNCS*, pages 205–217. Springer, 2015.

8 Pawel Gawrychowski. Pattern matching in Lempel-Ziv compressed strings: Fast, simple, and deterministic. In *Proc. ESA*, volume 6942 of *LNCS*, pages 421–432, 2011.

9 Pawel Gawrychowski, Tomohiro I, Shunsuke Inenaga, Dominik Köppl, and Florin Manea. Efficiently finding all maximal α-gapped repeats. *ArXiv e-prints abs/1509.09237*, 2015.

10 Pawel Gawrychowski and Florin Manea. Longest α-gapped repeat and palindrome. In *Proc. FCT 2015*, volume 9210 of *LNCS*, pages 27–40. Springer, 2015.

11 Dan Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology.* Cambridge University Press, New York, NY, USA, 1997.

12 Juha Kärkkäinen, Peter Sanders, and Stefan Burkhardt. Linear work suffix array construction. *J. ACM*, 53:918–936, 2006.

13 Tomasz Kociumaka, Jakub Radoszewski, Wojciech Rytter, and Tomasz Walen. Efficient data structures for the factor periodicity problem. In *Proc. SPIRE*, volume 7608 of *LNCS*, pages 284–294, 2012.

14 Roman Kolpakov and Gregory Kucherov. Finding repeats with fixed gap. In *Proc. SPIRE*, pages 162–168, 2000.

15 Roman Kolpakov and Gregory Kucherov. Searching for gapped palindromes. *Theoretical Computer Science*, 410(51):5365–5373, 2009. Combinatorial Pattern Matching. doi:10.1016/j.tcs.2009.09.013.

16 Roman Kolpakov, Mikhail Podolskiy, Mikhail Posypkin, and Nickolay Khrapov. Searching of gapped repeats and subrepetitions in a word. In *Proc. CPM*, volume 8486 of *LNCS*, pages 212–221, 2014.

17 Yuka Tanimura, Yuta Fujishige, Tomohiro I, Shunsuke Inenaga, Hideo Bannai, and Masayuki Takeda. A faster algorithm for computing maximal α-gapped repeats in a string. In *Proc. SPIRE 2015*, volume 9309 of *LNCS*, pages 124–136. Springer, 2015.