

Efficient Enumeration of Non-Equivalent Squares in Partial Words with Few Holes

Panagiotis Charalampopoulos · Maxime Crochemore · Costas S. Iliopoulos · Tomasz Kociumaka · Solon P. Pissis · Jakub Radoszewski · Wojciech Rytter · Tomasz Waleń

Received: date / Accepted: date

Abstract A word of the form WW for some word $W \in \Sigma^*$ is called a square. A partial word is a word possibly containing holes (also called don't cares). The hole is a special symbol $\diamond \notin \Sigma$ which *matches* any symbol from $\Sigma \cup \{\diamond\}$. A *p-square* is a partial word matching at least one square WW without holes. Two p-squares are called *equivalent* if they match the same set of squares. A p-square is called here *unambiguous* if it matches exactly one square WW without holes. Such p-squares are natural counterparts of classical squares. Let $\text{PSQUARES}_k(n)$ and $\text{USQUARES}_k(n)$ be the maximum number of non-equivalent p-squares and non-equivalent unambiguous p-squares in T over all partial words T of length n with at most k holes. We show asymptotically tight bounds:

$$\text{PSQUARES}_k(n) = \Theta(\min(nk^2, n^2)), \quad \text{USQUARES}_k(n) = \Theta(nk).$$

We present an algorithm that reports all non-equivalent p-squares in $\mathcal{O}(nk^3)$ time for a partial word of length n with k holes, for an integer alphabet. In particular, it runs in linear time for $k = \mathcal{O}(1)$ and its time complexity near-matches the asymptotic bound for $\text{PSQUARES}_k(n)$. We also show an $\mathcal{O}(n)$ -time algorithm that reports all non-equivalent p-squares of a given length.

The paper is a full and improved version of [8].

P. Charalampopoulos · M. Crochemore · C. S. Iliopoulos · S. P. Pissis
Department of Informatics, King's College London, London, UK
E-mail: [panagiotis.charalampopoulos, maxime.crochemore, costas.ilopoulos, solon.pissis]@kcl.ac.uk

T. Kociumaka · Jakub Radoszewski · W. Rytter · T. Waleń
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warsaw, Poland
E-mail: [kociumaka, jrad, rytter, walen]@mimuw.edu.pl

M. Crochemore
Also at Université Paris-Est, France

Keywords partial word · square in a word · approximate period · Lyndon word

1 Introduction

A *word* is a sequence of letters from a given alphabet Σ . By Σ^* we denote the set of all words over Σ . A word of the form $U^2 = UU$, for some word U , is called a *square*. For a word W , a factor is a subword composed of some number of consecutive letters and a *square factor* is a factor of W which is a square. Enumeration of square factors in words is a well-studied topic, both from a combinatorial and from an algorithmic perspective. Obviously, a word W of length n may contain $\Theta(n^2)$ square factors (e.g. $W = a^n$), however, it is known that such a word contains only $\mathcal{O}(n)$ distinct square factors [14,18]; currently the best known upper bound is $\frac{11}{6}n$ [11].

Moreover, all distinct square factors of a word over an integer alphabet can be listed in $\mathcal{O}(n)$ time using the suffix tree [16,1] or the suffix array and the structure of runs (maximal repetitions) in the word [9].

A *partial word* is a sequence of letters from $\Sigma \cup \{\diamond\}$, where \diamond denotes a *hole*, that is, a don't care symbol. Two symbols $a, b \in \Sigma \cup \{\diamond\}$ are said to *match* (denoted as $a \approx b$) if they are equal or one of them is a hole; note that this relation is not transitive. The relation of matching is extended in a natural way to partial words of the same length.

A partial word UV is called a *p-square* if $U \approx V$. Like in the context of words, a *p-square factor* of a partial word T is a factor being a p-square; see [2,7].

We introduce the notion of *equivalence* of p-square factors in partial words. Let $sq\text{-val}(UV)$ denote the set of squares that match the partial word UV :

$$sq\text{-val}(UV) = \{WW : W \in \Sigma^*, WW \approx UV\}.$$

Example 1 Let $\Sigma = \{a, b\}$. Then:

$$\begin{aligned} sq\text{-val}(a\diamond b a\diamond\diamond) &= sq\text{-val}(a\diamond\diamond \diamond\diamond b) = \{(ab)^2, (abb)^2\}, \\ sq\text{-val}(a\diamond\diamond \diamond ab) &= \{(ab)^2\}. \end{aligned}$$

The p-squares UV and $U'V'$ are called *equivalent* if $sq\text{-val}(UV) = sq\text{-val}(U'V')$ (denoted as $UV \equiv U'V'$). For example,

$$a\diamond b a\diamond\diamond \equiv a\diamond\diamond \diamond\diamond b, \text{ but } a\diamond b a\diamond\diamond \not\equiv a\diamond\diamond \diamond ab.$$

Let us assume that Σ is non-unary. We say that $X^2 = XX$ is the *representative* (also called *general form*; see [6]) of a p-square UV , denoted as $repr(UV)$, if

$$XX \approx UV \text{ and } sq\text{-val}(XX) = sq\text{-val}(UV).$$

(In other words, X is the “most general” partial word that matches both U and V .) It can be noted that the representative of a p-square is unique. Then $UV \equiv$

$U'V'$ if and only if $\text{repr}(UV) = \text{repr}(U'V')$. A p-square is called *unambiguous* if its representative does not contain the symbol \diamond and *ambiguous* otherwise.

Example 2

$\text{repr}(a\diamond b a\diamond\diamond) = (a\diamond b)^2$ and the p-square is ambiguous.

$\text{repr}(a\diamond\diamond\diamond ab) = (aab)^2$ and the p-square is unambiguous.

The set of non-equivalent p-square factors in a partial word T is denoted by $\text{psquares}(T)$. Thus, $\text{psquares}(T)$ corresponds to the set of different representatives of p-square factors of T .

Example 3 Let $T = ab\diamond\diamond ba\diamond aaba\diamond b$.

T contains 4 non-equivalent classes of p-squares of length 4:

1. $a\diamond aa$ with representative $(aa)^2$,
2. $ab\diamond\diamond \equiv \diamond ba\diamond \equiv aba\diamond$ with representative $(ab)^2$,
3. $\diamond\diamond ba \equiv ba\diamond a$ with representative $(ba)^2$, and
4. $b\diamond\diamond b$ with representative $(bb)^2$.

T contains 4 equivalence classes of p-squares of length 6 with representatives:

$$(aab)^2, (aba)^2, (baa)^2, (ba\diamond)^2;$$

see also Fig. 1.

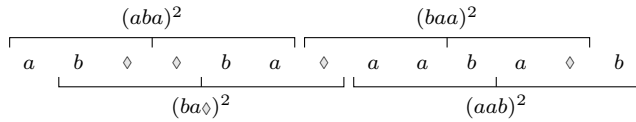


Fig. 1 All non-equivalent p-square factors of length 6 with their representatives in an example partial word.

Overall, we have $|\text{psquares}(T)| = 14$. The remaining 6 representatives are:

$$\diamond\diamond, aa, bb, (aaba\diamond)^2, (abaab)^2, (baaba)^2.$$

Our work is devoted to enumeration of non-equivalent p-square factors in a partial word with a given number $k > 0$ of holes.

Previous results. Alongside [2, 6, 7], we define a *solid square* as a square of a word and a *square subword* of a partial word T as a solid square that matches a factor of T .

Previous studies on squares in partial words were mostly focused on combinatorics. They started with the case of $k = 1$ [6], in which case distinct square subwords correspond to non-equivalent p-square factors. It was shown that a partial word with one hole contains at most $\frac{7}{2}n$ distinct square subwords [4] ($3n$ for binary partial words [17]). Also a generalization of the three squares lemma (see [10]) was proposed for partial words [5]. As for a larger number of holes, the existing literature is devoted mainly to counting the number of

distinct square subwords of a partial word [6,2] or all occurrences of p-square factors [3,2]. On the algorithmic side, [22] proved that the problem of counting distinct square subwords of a partial word is #P-complete and [12,21] and [7] showed quadratic- and nearly-quadratic-time algorithms for finding all occurrences of p-square factors and primitively-rooted p-square factors of a partial word, respectively.

Our combinatorial results. Let $\text{PSQUARES}_k(n)$ and $\text{USQUARES}_k(n)$ be the maximum number of non-equivalent p-squares and non-equivalent unambiguous p-squares in T over all partial words T of length n with at most k holes. We show the following bounds:

$$\text{PSQUARES}_k(n) = \Theta(\min(nk^2, n^2)), \quad \text{USQUARES}_k(n) = \Theta(nk).$$

This work can be viewed as a generalization of the results on partial words with one hole [6,4,17] to k holes.

Our algorithmic results. We present an algorithm that reports all elements of the set $\text{psquares}(T)$ in a partial word of length n with k holes in $\mathcal{O}(nk^3)$ time. In particular, our algorithm runs in linear time for $k = \mathcal{O}(1)$ and its time complexity near-matches the maximum number of non-equivalent p-square factors. We also show an $\mathcal{O}(n)$ -time algorithm that reports all non-equivalent p-squares of a given length. The algorithms assume integer alphabet $\Sigma \subseteq \{1, \dots, n^{\mathcal{O}(1)}\}$. We use recently introduced advanced data structures from [19].

Comparison with the conference version. The paper is an extended version of [8]. As far as combinatorics of p-squares is concerned, the conference version of the paper derived the bound $\text{PSQUARES}_k(n) = \Theta(\min(n^2, nk^2))$. Let $\text{ASQUARES}_k(n)$ be the maximum number of non-equivalent ambiguous p-squares in T over all partial words T of length n with at most k holes. The bound was proved by showing that $\text{ASQUARES}_k(n) = \Theta(\min(n^2, nk^2))$ and that $\text{USQUARES}_k(n) = \mathcal{O}(nk^2)$. As a new contribution here, we present a tight estimation $\text{USQUARES}_k(n) = \Theta(nk)$. This lets us identify ambiguous p-squares as the ones that attain the bound on $\text{PSQUARES}_k(n)$. On the algorithmic side, [8] presented an algorithm computing the set $\text{psquares}(T)$ in $\mathcal{O}(nk^3)$ time. Here the readability of the algorithm has been considerably improved; we also show a linear-time algorithm that reports all non-equivalent p-squares of a specified length.

Structure of the paper. After the Preliminaries comes the algorithmic part of the paper, which is followed by the combinatorial part. In Section 3 we show an $\mathcal{O}(n)$ -time algorithm that reports all non-equivalent p-squares of a specified length and, as an immediate corollary, $\mathcal{O}(nk^2)$ -time computation of all non-equivalent ambiguous p-squares. Then in Section 4 we give an $\mathcal{O}(nk^3)$ -time algorithm for computing all non-equivalent unambiguous p-squares. Asymptotic bounds for ambiguous p-squares and unambiguous p-squares are presented in Sections 5 and 6, respectively.

2 Preliminaries

For a word $W \in \Sigma^*$, by $|W| = n$ we denote the length of W , and by W_i , for $i = 1, \dots, n$, the i th letter of W . For $1 \leq i \leq j \leq n$, by $[i..j]$ and $(i..j)$ we denote integer intervals $\{i, \dots, j\}$ and $\{i + 1, \dots, j\}$, respectively. $W[i..j]$ denotes the *factor* of W equal to $W_i \dots W_j$; we also use the notation $W[I]$, where I is an integer interval. A factor of the form $W[1..j]$ is called a *prefix*, a factor of the form $W[i..n]$ is called a *suffix*.

For a partial word T we use the same notation as for words: $|T| = n$ for its length, T_i for the i th letter, $T[i..j]$ for a factor. If T does not contain holes, then it is called *solid*. The relation \approx of matching on $\Sigma \cup \{\diamond\}$ is defined as: $a \approx a$, $\diamond \approx a$, and $a \approx \diamond$ for all $a \in \Sigma \cup \{\diamond\}$.

We define an operation \odot such that: $a \odot a = a \odot \diamond = \diamond \odot a = a$ for all $a \in \Sigma \cup \{\diamond\}$, and otherwise $a \odot b$ is undefined. Two equal-length partial words S and T are said to *match* (denoted as $S \approx T$) if $S_i \approx T_i$ for all $i = 1, \dots, n$. In this case, we denote

$$S \odot T = S_1 \odot T_1, \dots, S_n \odot T_n.$$

Also note that if UV is a p-square, then $\text{repr}(UV) = (U \odot V)^2$.

If $U \approx T[i..i + |U| - 1]$ for a partial word U , then we say that U occurs in T at position i .

Two equal-length partial words U and V are called *cyclic shifts* if there are partial words X, Y such that $U = XY$ and $V = YX$. We denote this as $\text{rot}(U, |X|) = V$, where $|X|$ is the *shift value*.

For a partial word X , by $\#\diamond(X)$ we denote the number of holes in X . For $1 \leq i \leq n$ and $0 \leq q \leq \log n$, we denote $T_{i,q} = T[i.. \min(n, i + 2^q - 1)]$. We say that $T_{i,q}$ is a *q-basic factor* of the partial word T . In other words, *q-basic factors* are factors of T of length 2^q and suffixes of T of length at most 2^q . By $\mathcal{B}(T)$ we denote the set of all basic factors of T .

Lemma 1 *If T is a partial word of length n with k holes, then*

$$\sum_{W \in \mathcal{B}(T)} \#\diamond(W) \leq 2nk.$$

Proof The number of q -basic factors that contain a given position $i \in \{1, \dots, n\}$ is at most 2^q . Thus the total number of basic factors that contain a given hole position i is at most:

$$\sum_{q=0}^{\lceil \log n \rceil} 2^q \leq 2n. \quad \square$$

We say that a p-square is an *unambiguous p-square* (*u-square*) if its representative is solid and an *ambiguous p-square* (*a-square*) otherwise. By $\text{asquares}(T)$ and $\text{usquares}(T)$ we denote the sets of non-equivalent factors of T being a-squares and u-squares, respectively. Obviously:

Observation 1 $\text{psquares}(T) = \text{asquares}(T) \cup \text{usquares}(T)$

2.1 Periods in Solid and Partial Words

A positive integer q is called a *period* of a word W if $W_i = W_{i+q}$ for all $i = 1, \dots, n - q$. In this case, $W[1..q]$ is called a *string period* of W . A word W is called *periodic* if it has a period q such that $2q \leq |W|$.

A *quantum period* of a partial word T is a positive integer q such that $T_i \approx T_{i+q}$ for all $i = 1, \dots, n - q$. A *deterministic period* of T is an integer q such that there exists a word W such that $W \approx T$ and W has a period q .

The partial word T is called *quantum (deterministically) periodic* if it has a quantum (deterministic) period q such that $2q \leq n$.

For a partial word U and integer $\delta > 0$, we denote

$$\begin{aligned} \text{Mis}_\delta(U) &= \{i \in [\delta + 1..|U|] : U_{i-\delta} \not\approx U_i\}, \\ \text{Holes}(U) &= \{i \in [1..|U|] : U_i = \diamond\}. \end{aligned}$$

We say that p is a *d-approximate quantum period* of a partial word T if $|\text{Mis}_d(T)| \leq d$. Note that a 0-approximate quantum period is exactly a quantum period.

Lemma 2 *Assume that $U \approx V$.*

- (a) *If $i \in \text{Mis}_\delta(U)$, then $i \in \text{Mis}_\delta(V)$ or $i \in \text{Holes}(V)$ or $i - \delta \in \text{Holes}(V)$.*
- (b) *$|\text{Mis}_\delta(U)| \leq |\text{Mis}_\delta(V)| + 2|\text{Holes}(V)|$.*
- (c) *If $\delta \geq \frac{1}{2}|U|$, then $|\text{Mis}_\delta(U)| \leq |\text{Mis}_\delta(V)| + |\text{Holes}(V)|$.*

Proof (a): We have $V_{i-\delta} \approx U_{i-\delta} \not\approx U_i \approx V_i$. This means that $U_{i-\delta}, U_i \in \Sigma$. Hence, if $i \notin \text{Holes}(V)$ and $i - \delta \notin \text{Holes}(V)$, then $V_{i-\delta} = U_{i-\delta}$ and $V_i = U_i$, so $i \in \text{Mis}_\delta(V)$.

Point (b) follows from point (a). Also point (c) follows from point (a). Indeed, if $i \in \text{Mis}_\delta(U)$ in this case, then for each of the positions $i, i - \delta$ in V , if it contains a hole, then it is counted only for the index i . \square

3 Computing All p-Squares of Specified Length and Non-Equivalent Ambiguous p-Squares

In this section we develop an $\mathcal{O}(n)$ -time algorithm that enumerates all non-equivalent p-squares of a half length d in a partial word T of length n . As a corollary, we obtain a simple computation of all non-equivalent ambiguous p-squares in optimal time.

For a partial word T , we denote by T' a partial word of length $n - d$ such that $T'[i] = T[i] \odot T[i+d]$ for each $i = 1, \dots, n - d$. If $T[i] \odot T[i+d]$ is undefined (since $T[i] \not\approx T[i+d]$), we set the value to a symbol $\# \notin \Sigma$.

Observation 2 (a) *$T[i..i + 2d - 1]$ is a p-square if and only if $T'[i..i + d - 1]$ does not contain the symbol $\#$.*

(b) *If $T[i..i + 2d - 1]$ is a p-square, then $\text{repr}(T[i..i + 2d - 1]) = (T'[i..i + d - 1])^2$.*

Proof (a) If $T'[i..i + d - 1]$ contains the symbol $\#$, this means that $T[j] \not\approx T[j + d]$ for some $j \in [i..i + d - 1]$. Hence, $T[i..i + 2d - 1]$ is not a p-square. Otherwise, $T[j] \approx T[j + d]$ for all $j \in [i..i + d - 1]$. Hence, $T[i..i + 2d - 1]$ indeed is a p-square.

(b) If $T[i..i + 2d - 1]$ is a p-square, then

$$(T'[i..i + d - 1])^2 = (T[i..i + d - 1] \odot T[i + d..i + 2d - 1])^2 = \text{repr}(T[i..i + 2d - 1]).$$

□

Example 4 Let us consider the partial word $T = ab\diamond ba\diamond aaba\diamond b$ from Example 3. For $d = 2$ we construct the following partial word T' :

$$abbabaa\#ab\#$$

from which we conclude that T contains p-squares of half length 2 with representatives:

$$(ab)^2, (bb)^2, (ba)^2, (ab)^2, (ba)^2, (aa)^2, (ab)^2.$$

For $d = 3$ we construct the partial word T' :

$$aba\diamond\#abaab$$

which means that T contains p-squares of half length 3 with representatives:

$$(aba)^2, (ba\diamond)^2, (aba)^2, (baa)^2, (aab)^2.$$

Theorem 3 *All non-equivalent p-squares of half length d in a partial word of length n can be reported (as factors of the partial word) in $\mathcal{O}(n)$ time.*

Proof Let T be a partial word of length n . In $\mathcal{O}(n)$ time we compute T' . Let S_1, \dots, S_q be a partition of T' into maximal factors that do not contain the symbol $\#$. By Observation 2, our task is equivalent to reporting all distinct factors of length d of the partial words S_j . This can be performed by listing all nodes (implicit and explicit) at depth d in the *generalized suffix tree* \mathcal{T} of S_1, \dots, S_q , that is, in the suffix tree of $S_1\#_1 \dots S_q\#_q$, where $\#_1, \dots, \#_q \notin \Sigma$ are distinct symbols. For details, see [15]. As the suffix tree of a word of length n can be constructed in $\mathcal{O}(n)$ time [13], the whole algorithm works in $\mathcal{O}(n)$ time. □

As a corollary we obtain efficient computation of non-equivalent a-squares.

Theorem 4 *For a partial word T of length n with k holes, all elements of the set $\text{asquares}(T)$ can be reported in $\mathcal{O}(nk^2)$ time.*

Proof There are at most k^2 possible lengths of ambiguous p-squares. For each length we use the algorithm of Theorem 3 to report all non-equivalent p-squares. This takes $\mathcal{O}(nk^2)$ time. In the end, for each length we need to filter out unambiguous p-squares. For a specified half length d , it suffices to check, for each p-square $T[i..i + 2d - 1]$ found, if $T'[i..i + d - 1]$ contains a hole. This condition can be checked in $\mathcal{O}(1)$ time if the prefix sums of the sequence $a_i = [T'[i] = \diamond]$ are stored. □

4 Computing All Non-Equivalent Unambiguous p-Squares

We start the description of the algorithm by an abstract lemma that lets us efficiently generate all distinct squares induced by a special family of (solid) words.

4.1 Computing Squares Induced by a Family of Words

For a word S , we define its *primitive root* U as the shortest word such that $U^k = S$ for some integer $k \geq 1$. The *Lyndon root* λ of a word U is the minimal cyclic shift of the shortest string period of U . The notion of a Lyndon root was introduced in the context of runs in [9].

Example 5 The Lyndon root of $U = abaababaababa$ is $aabab$. The word U is periodic and its shortest period is 5.

For a word W and its period q , by $squares(W, q)$ we denote the set of square factors of W of length $2q$. We say that $squares(W, q)$ is the set of squares induced by the word W with the period q . Each square factor in $squares(W, q)$ can be represented in $\mathcal{O}(1)$ space by specifying its occurrence in W .

Lemma 3 *Assume we have a family of possibly unknown words W_1, W_2, \dots, W_N with periods q_1, \dots, q_N , a positive integer k and positive integers $n_i, first_i, \ell_i$ for $i = 1, \dots, N$, such that:*

- (1) $n_i \leq n$ is the length of W_i and $2q_i \leq n_i$;
- (2) all the words W_i for which $2q_i = n_i$ (so-called short words) are distinct;
- (3) for a given q_i , the number of words W_i for which $2q_i < n_i$ (so-called long words) is at most k ;
- (4) $first_i$ is the starting position of the first occurrence of the Lyndon root λ_i of W_i in W_i and ℓ_i is its length;
- (5) any two Lyndon roots λ_i, λ_j can be compared in $O(k)$ time.

Then we can compute the cardinality of the set $SQ = \bigcup_i squares(W_i, q_i)$ and its representation (as sets of intervals in W_i 's) in $O(Nk^2 + nk^3 + |SQ|)$ time.

Proof Let us start with the following observation; see also Fig. 2. The same type of observation was used in [9].

Observation 5 *For every i , the set $squares(W_i, q_i)$ equals*

$$\{rot(\lambda_i^{2q_i/\ell_i}, a) : a = (1 - first_i) \bmod \ell_i, \dots, (1 + n_i - 2q_i - first_i) \bmod \ell_i\}.$$

The above set of integers is denoted by I_i . Note that it forms one cyclic subinterval of $[0.. \ell_i - 1]$ (composed of up to two standard intervals) and that it can be computed in $\mathcal{O}(1)$ time. Each of the elements $a \in I_i$ represents a unique square that is induced by W_i and q_i .

We make two transformations of the set of intervals I_i so that, in the end, each square from the set SQ is induced by exactly one word W_i with period q_i .

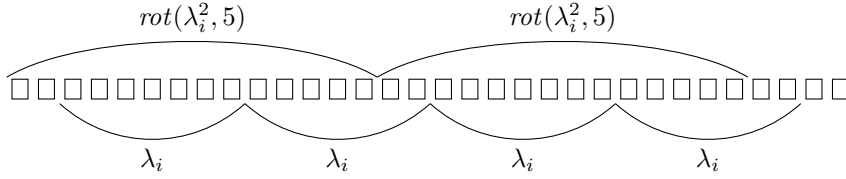


Fig. 2 In this case $n_i = 32$, $q_i = 14$, $\ell_i = 7$, $first_i = 3$. Hence, W_i induces 5 squares being cyclic shifts of λ_i^4 , that is, $I_i = [0, 2] \cup [5, 6]$.

If any of the intervals is made empty, this corresponds to removing the word as unnecessary. The first transformation deals with the long words W_i ; by definition, at most k of them share the same period q_i .

First transformation: For every pair W_i, q_i and W_j, q_j of long words such that $i \neq j$ and $q_i = q_j$, we check if $\lambda_i = \lambda_j$. If $I_i \subseteq I_j$, we dispose of W_i . Likewise, if $I_j \subseteq I_i$, we remove W_j . If none of the two cases holds and still $I_i \cap I_j \neq \emptyset$, we trim I_j to make it disjoint with I_i .

Complexity: All long words can be sorted by their periods in $\mathcal{O}(N + n)$ time by bucket sort. There are $n/2$ buckets and each bucket contains at most k words. For each of the $k(k - 1)/2$ pairs of long words in a bucket, we check equality of their Lyndon roots, which takes $\mathcal{O}(k)$ time per pair and $\mathcal{O}(nk^3)$ time overall. The time complexity of trimming of cyclic intervals is dominated by this step.

Second transformation: For every short word W_i with period q_i and long word W_j with period $q_j = q_i$, we check if $\lambda_i = \lambda_j$. If so and $I_i \subseteq I_j$, we remove W_i . Note that I_i is a singleton.

Complexity: All words can be sorted by their periods in $\mathcal{O}(N + n)$ time by bucket sort. For each short word W_i , we need to inspect at most k long words and check if their Lyndon roots are equal. This takes $\mathcal{O}(k^2)$ time per short word, $\mathcal{O}(Nk^2)$ time overall. Checking inclusion of elements in cyclic intervals is dominated by this step.

The two transformations take $\mathcal{O}(Nk^2 + nk^3)$ time in total. Afterwards each square is induced by exactly one interval I_i for a word W_i and period q_i , so we can list all the distinct squares in $\mathcal{O}(|SQ|)$ time. \square

For a partial word T , by $ssquares(T)$ we denote the set of distinct solid factors of T being squares. The following fact was already mentioned in the Introduction.

Fact 6 ([1,9,16]) *All distinct squares in a word of length n can be computed in $\mathcal{O}(n)$ time.*

By substituting all holes in a partial word with distinct symbols $\#_1, \dots, \#_k$, we obtain the following corollary.

Corollary 1 *For partial word T of length n , the set $ssquares(T)$ can be computed in $\mathcal{O}(n)$ time.*

The algorithm of [9] actually computes the set $ssquares(T)$ together with all the data in assumption of Lemma 3. These are the short words in the construction.

In the following section we construct a family \mathcal{F} of words (called *sealed fragments*) that represent the u-squares that contain a hole and compute for them the data required in Lemma 3. These are the long words in the construction. Afterwards we list all distinct representatives of u-squares using Lemma 3. Then non-equivalent u-squares are extracted from their representatives.

4.2 Computing a Special Family of Sealed Fragments

If T is a partial word, then U is a *sealed fragment* of T if U is a factor of T with holes substituted by solid symbols. By $unseal(U)$ we denote the original factor of the partial word.

A sealed fragment is always solid. Obviously, a sealed fragment can be represented in space proportional to the number of holes that were substituted. For example, if $T[i..i + 2q - 1]$ is a u-square, then $repr(T[i..i + 2q - 1])$ is a sealed fragment.

If W is a (solid) word, then by a *d-fragment* we mean a concatenation of d factors $W[i_1..j_1] \dots W[i_d..j_d]$. A d -fragment can be represented in $\mathcal{O}(d)$ space. In [19] it was shown that several types of operations on d -fragments can be performed in $\mathcal{O}(d)$ or $\mathcal{O}(d^2)$ time after $\mathcal{O}(n)$ -time preprocessing. We notice here that a sealed fragment of a partial word T with k holes corresponds to a d -fragment with $d = \mathcal{O}(k)$ in a word that corresponds to T where \diamond is treated as an alphabet symbol. Thus the following simple fact is a consequence of Observation 18 from [19] that was stated in terms of d -fragments.

Fact 7 ([19]) *For a partial word of length n with k holes, after $\mathcal{O}(n)$ -time preprocessing, the length of the longest common prefix (or suffix) of any two sealed fragments can be computed in $\mathcal{O}(k)$ time. In particular, equality of sealed fragments can be checked within the same time complexity.*

Definition 1 A family of pairs (W_i, q_i) , where each W_i is a sealed fragment of a partial word T of length n with k holes and q_i is a positive integer, is called an *S-family* if it satisfies the following properties:

- (a) For every i , q_i is a period of W_i and $|W_i| \geq 2q_i$.
- (b) For every i , there are no two holes in $unseal(W_i)$ at distance q_i .
- (c) For every $q = 1, \dots, n$, there are $\mathcal{O}(k)$ sealed fragments with $q_i = q$.
- (d) If X is a non-solid u-square in T , then X is a factor of $unseal(W_i)$ for some W_i with $q_i = \frac{1}{2}|X|$.

The size of an S-family follows from point (c).

Observation 8 *An S-family contains $\mathcal{O}(nk)$ elements and thus can be represented in $\mathcal{O}(nk^2)$ space.*

In the following lemma we provide an algorithm for constructing an S-family. Our approach resembles computing anchored squares in the Main-Lorentz algorithm [20].

Lemma 4 *For a partial word T of length n with k holes, an S-family can be computed in $\mathcal{O}(nk^2)$ time.*

Proof Each non-solid u-square X contains a hole in the first half or in the second half. Below, we construct an S-family for u-squares containing a hole in the second half. A symmetric procedure deals with the u-squares containing a hole in the first half.

For a hole h and integer q , we define the family $\mathcal{S}(q, h)$ of u-squares of length $2q$, which contain h as the leftmost hole in the second half. For each non-empty set $\mathcal{S}(q, h)$, we shall construct a sealed fragment W with period q so that each u-square $X \in \mathcal{S}(q, h)$ is a factor of $\text{unseal}(W)$.

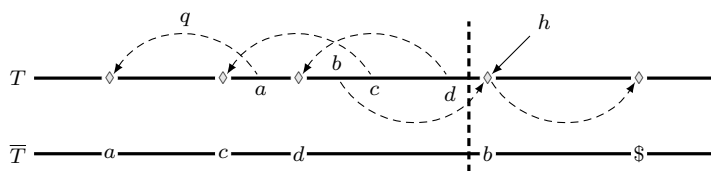


Fig. 3 A partial word T with 5 holes and the corresponding sealed text \bar{T} with holes sealed by 5 (solid) symbols implied by the value of q . The rightmost hole is filled by a special unique marker denoted by $\$$.

First, let us seal the text consistently with the representatives of u-squares in $\mathcal{S}(q, h)$. A hole at position $i < h$ may only be contained in the first half, while a hole at position $i \geq h$ may only be contained in the second half of such a u-square. Thus, we seal the hole $T[i] = \diamond$ with $T[i+q]$ if $i < h$, and with $T[i-q]$ if $i \geq h$. Any remaining hole is sealed with a unique marker (distinct for every hole). This produces a sealed fragment \bar{T} that covers the whole partial word T ; see Fig. 3. Let z be the distance between h and the position of the preceding hole ($z = +\infty$ if there is none). We define W as a maximal fragment of \bar{T} which contains $\bar{T}[h-q..h]$, is contained in $\bar{T}[h-q-\min(q-1, z-1)..h+q-1]$, and has period q . If $|W| < 2q$, there is no u-square of the desired type and we can discard W .

The fragment W is unique and it can be retrieved in $\mathcal{O}(k)$ time using Fact 7. Indeed, it suffices to compute the longest common prefix P of $\bar{T}[h-q..n]$ and $\bar{T}[h..n]$, the longest common suffix S of $\bar{T}[1..h-q]$ and $\bar{T}[1..h]$, and take the possibly trimmed fragment $S\bar{T}[h-q+1..h-1]P$; see Fig. 4. We may need to

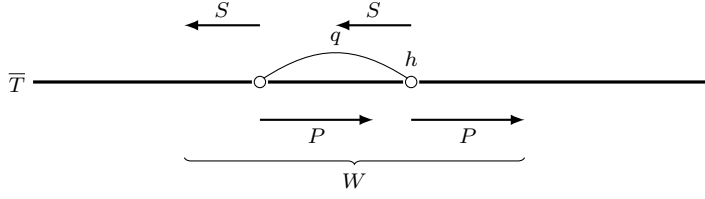


Fig. 4 The fragment W with period q anchored at h is computed using an operation of modified longest common extension and its reversed version. We have $|S| \leq \min(q-1, z-1)$, where z is the distance between h and the position of the preceding hole, and $|P| \leq q-1$.

trim S so that its length exceeds neither $q-1$ (so that the hole at position h is contained in the right half of the square) nor $z-1$ (so that h is the leftmost hole in the right half). Similarly, we may need to trim P to the length $q-1$. In total, the construction takes $\mathcal{O}(nk^2)$ time.

Let us verify that this construction indeed satisfies the condition of Definition 1. For each hole we construct just one sealed fragment, so the condition (c) is satisfied. Clearly, W has period q and $|W| \geq 2q$, which yields point (a). Moreover, if $X = T[i..j] \in \mathcal{S}(q, h)$, then $\text{repr}(X) = \bar{T}[i..j]$, so (by maximality) $\text{repr}(X)$ is contained in W , and X is contained in $\text{unseal}(W)$. This gives point (d). Finally, we shall prove that $\text{unseal}(W)$ does not contain two holes at distance q (condition (b)). Suppose that the holes are at positions i and $i+q$. Observe that one of the holes is sealed with a unique marker, which contradicts $\bar{T}[i] = \bar{T}[i+q]$. This completes the proof. \square

Example 6 Consider the partial word $T = ab\lozenge ba\lozenge aaba\lozenge b$ from Example 3 and $q = 2$. For the first hole we obtain the following word \bar{T} :

$$ab|abbabaabbb$$

with the original positions of holes underlined. The computed sealed fragment is $W = abab$. For the second hole we obtain the word \bar{T} :

$$abb|bbabaabbb$$

and the sealed fragment $bbbb$. For the third hole \bar{T} equals:

$$abbaba|baabbb$$

and the sealed fragment is $ababa$ so $\mathcal{S}(2, 8) = \{\lozenge ba\lozenge, ba\lozenge a\}$. Finally, for the fourth hole \bar{T} equals:

$$abbaba@aaba|bb$$

and the sealed fragment is $abab$.

Henceforth we denote by \mathcal{F} the S-family constructed in Lemma 4. In order to transform it into an instance of Lemma 3, we need to compute the Lyndon roots of the sealed fragments W_i (that is, the values first_i and ℓ_i).

4.3 Lyndon Roots of Sealed Fragments

We will show how to compute Lyndon roots λ_i of sealed fragments $(W_i, q_i) \in \mathcal{F}$. Obviously, a Lyndon root of a sealed fragment can be represented in the same space complexity as the sealed fragment itself.

Let us start with the following fact that encapsulates Theorems 20 and 23 from [19].

Fact 9 ([19]) *For a word of length n , after $\mathcal{O}(n)$ -time preprocessing,*

- (a) *the length of the lexicographically minimal suffix of a d -fragment can be computed in $\mathcal{O}(d^2)$ time;*
- (b) *the shift value of the minimal cyclic shift of a d -fragment can be computed in $\mathcal{O}(d^2)$ time.*

As a consequence of Fact 9(a) we obtain:

Observation 10 *For a word of length n , after $\mathcal{O}(n)$ -time preprocessing, the length of the lexicographically maximal suffix of a d -fragment can be computed in $\mathcal{O}(d^2)$ time.*

Proof To compute the maximal suffix instead of the minimal suffix, we reverse the lexicographic order on the alphabet and append the d -fragment in question with a letter that is greater than all the letters from Σ . \square

Fact 9(a) and Observation 10 provide us with the following toolbox for sealed fragments.

Lemma 5 ([19]) *For a partial word of length n with k holes, after $\mathcal{O}(n)$ -time preprocessing,*

- (a) *the length of the lexicographically maximal suffix of a sealed fragment can be computed in $\mathcal{O}(k^2)$ time.*
- (b) *the shift value of the minimal cyclic shift of a sealed fragment can be computed in $\mathcal{O}(k^2)$ time.*

Lemma 6 *If W is a periodic sealed fragment and q is its period (not necessarily shortest) such that $2q \leq |W|$, then the length of the Lyndon root of W and its first occurrence in W can be computed in $\mathcal{O}(k^2)$ time after $\mathcal{O}(n)$ -time preprocessing.*

Proof Let $s \in [0..q-1]$ be the shift value of the minimal cyclic shift of $W[1..2q]$ and $i = s + 1$. It can be computed in $\mathcal{O}(k^2)$ time using Lemma 5(a). We know that the Lyndon root λ of W starts at the position s and that its length ℓ divides q .

We then use Lemma 5(b) to find the starting position i' of the maximal suffix of $W[i+1..|W|]$ with the reversed lexicographic order of the alphabet. If $W[i'..|W|]$ is a prefix of $W[i..|W|]$, then $\ell = i' - i$, and otherwise $\ell = q$. We check this condition in $\mathcal{O}(k)$ time using Fact 7. Finally, we return $s \bmod \ell$ and ℓ ; see Fig. 5. \square

c	b	a	b	c	b	a	b	c	b	a	b	c	b	a	b	c	b
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Fig. 5 We have $|W| = 18$ and $q = 8$. The minimal cyclic shift of $W[1..16]$ is $(abc)^4$ and starts, e.g., at position $i = 7$, so the shift value is $s = 6$. Then, the maximal suffix over reversed alphabet of $W[8..18]$ starts at position $i' = 11$. We have $\ell = i' - i = 4$ and $s \bmod \ell = 2$. The Lyndon root of W is $abcb$.

By point (a) of the definition of an S-family we immediately obtain:

Corollary 2 *The Lyndon roots of all sealed fragments $(W_i, q_i) \in \mathcal{F}$ can be computed in $\mathcal{O}(nk^3)$ time after $\mathcal{O}(n)$ -time preprocessing.*

With this missing puzzle we are ready to conclude the algorithm for reporting all unambiguous p-square factors of a partial word.

Theorem 11 *For a partial word T of length n with k holes, all elements of the set $usquares(T)$ can be reported in $\mathcal{O}(nk^3)$ time.*

Proof We construct a family of sealed fragments that consists of the solid p-squares $ssquares(T)$ and an S-family \mathcal{F} . By Corollary 1 and Lemma 4, this family can be constructed in $\mathcal{O}(nk^2)$ time. We compute Lyndon roots of all the sealed fragments in $\mathcal{O}(nk^3)$ time using Corollary 2. For each solid p-square we may compute its Lyndon root in $\mathcal{O}(k^2)$ time using Lemma 6; we can also use the Lyndon roots as computed in [9].

The constructed family satisfies the assumption of Lemma 3 with $N = \mathcal{O}(nk)$. (Actually, if for any sealed factor (W_i, q_i) of the S-family \mathcal{F} we have $|W_i| = 2q_i$, we need to check if it equals any of the solid squares of the same length and, if so, remove it, so that no two short words repeat.) This lemma lets us report all the distinct representatives of u-squares in $\mathcal{O}(nk^3 + |SQ|)$ time. The total number of u-squares that will be generated is $\mathcal{O}(nk)$ due to Theorem 16. This gives the final complexity of the algorithm. \square

5 Combinatorial Bounds for Ambiguous p-Squares

Let T be a partial word of length n with k holes. The upper bound in the case of a-squares is straightforward.

Theorem 12 *If T is a partial word of length n with k holes, then $asquares(T) = \mathcal{O}(nk^2)$.*

Proof The number of possible lengths of a-squares is at most $\binom{k}{2}$, since we have $\binom{k}{2}$ possible distances between the k holes. Consequently, the number of p-squares with such lengths is at most nk^2 . \square

Let us proceed to the lower bound proof. We say that a set A of positive integers is an (m, t) -cover if the following conditions hold:

- (1) For each $d \geq m$, A contains at most one pair of elements with difference d ;
(2) $|\{ |j - i| \geq m : i, j \in A \}| \geq t$.

Example 7

$\{1, 2, 3, 6, 9, 12\}$ is a $(3, 9)$ -cover.

$\{1, 2, 3, 11, 14, 17\}$ is a $(8, 9)$ -cover.

For a set $A \subseteq [1..n]$ we denote by $W_{A,n}$ the partial word of length n over the alphabet Σ such that $W_{A,n}[i] = \diamond \Leftrightarrow i \in A$, and $W_{A,n}[i] = a$ otherwise.

Lemma 7 *Assume that $A \subseteq [1..n]$ is an (m, t) -cover such that $m = \Theta(n)$, $|A| = k$, and $t = \Omega(k^2)$. Let $\Sigma = \{a, b\}$ be the alphabet. Then*

$$\text{asquares}(a^{n-2} \cdot W_{A,n} \cdot a^{n-2}) = \Omega(n \cdot k^2).$$

Proof Each even-length factor of $a^{n-2} \cdot W_{A,n} \cdot a^{n-2}$ is a p-square. Let \mathcal{Z} be the set of these factors X which contain two positions $i < j$ containing holes with $j - i \geq m$ and $|X| = 2(j - i)$. As A is an (m, t) -cover, i and j are determined uniquely by $d = j - i$. Then all elements of \mathcal{Z} are pairwise non-equivalent a-squares. The size of \mathcal{Z} is $\Omega(mt)$ which is $\Omega(n \cdot k^2)$. \square

Theorem 13 *For every positive integer n and $k \leq \sqrt{2n}$, there is a partial word of length n with k holes that contains $\Omega(nk^2)$ non-equivalent a-square factors.*

Proof Due to Lemma 7, it is enough to construct a suitable set A . By monotonicity, we may assume that k and n are even. We take:

$$A = [1.. \frac{k}{2}] \cup \{j \cdot \frac{k}{2} + \frac{n}{2} : 1 \leq j \leq \frac{k}{2}\}.$$

We claim that A is an $(\frac{n}{2}, t)$ -cover for $t = \Omega(k^2)$. Indeed, take any $i, j \in [1.. \frac{k}{2}]$. Then $j \cdot \frac{k}{2} + \frac{n}{2} - i \geq \frac{n}{2}$ and all such values are distinct; hence, $t = \frac{k^2}{4}$. The thesis follows from the claim. \square

Example 8 Let us consider the $(8, 9)$ -cover from Example 7, which is a subset of $[1..n]$ for $n = 17$, and the partial word $a^{n-2} \cdot W_{A,n} \cdot a^{n-2}$:

$$a^{15} \diamond \diamond \diamond a a a a a a a a \diamond a a \diamond a a \diamond a^{15}.$$

This partial word contains all a-squares with representatives being cyclic shifts of $(\diamond a^i)$ for $i = 7, \dots, 15$.

6 Combinatorial Bounds for Unambiguous p-Squares

The following theorem shows a lower bound construction. Afterwards we design an upper bound that asymptotically matches this lower bound.

Theorem 14 *For every positive integers n and k , $k \leq \frac{1}{3}n$, there is a partial word of length n with k holes that contains $\Omega(nk)$ non-equivalent u-square factors.*

Proof Let us consider the following partial word over the alphabet $\{a, b\}$:

$$W_m = a^{m-1}ba^{m-k}\diamond^k a^{m-1}.$$

Then for every $i \in [1..k]$, W_m has $m - k + i$ u-square factors of half length $m - k + i$ containing the letter b ; see also Fig. 6. Altogether the number of such u-squares is:

$$\sum_{i=1}^k m - k + i = \Omega(nk),$$

where $n = 3m - 1 = |W_m|$. If n gives a different remainder modulo 3, we can pad W_m with the letter a . \square

$$\begin{array}{cccccccc}
 & & & b & a & a & \diamond & \diamond & a & a & a \\
 & & & a & b & a & a & \diamond & \diamond & a & a \\
 & & a & a & b & a & a & \diamond & \diamond & a & \\
 a & a & a & b & a & a & \diamond & \diamond & & & \\
 W_m: & a & a & a & b & a & a & \diamond & \diamond & a & a & a
 \end{array}$$

Fig. 6 The u-square factors of half length 4 with the letter b in $W_m = a^{m-1}ba^{m-k}\diamond^k a^{m-1}$ for $m = 4$, $k = 2$.

If X is a partial word, then by $LONG(X)$ we denote the set of all p-squares of length at least $\frac{1}{2}|X|$ which occur in X as a prefix.

If A is a set of numbers, $|A| \geq 2$, then we denote

$$\text{mingap}(A) = \min\{|b - a| : a, b \in A, a \neq b\}.$$

If \mathcal{Z} is a set of partial words, then $\text{mingap}(\mathcal{Z})$ denotes $\text{mingap}(\{|S| : S \in \mathcal{Z}\})$.

Lemma 8 (Three p-Squares Lemma) *Let X be a partial word with k holes. Assume that the set $LONG(X)$ contains at least three elements. Then $\delta = \text{mingap}(LONG(X))/2$ is a $12k$ -approximate quantum period of the longest p-square in $LONG(X)$.*

Proof Let $B, C \in LONG(X)$ be p-squares such that $|B| - |C| = 2\delta$. Also let A and D be the longest and the shortest element of $LONG(X)$, respectively. Let $|A| = 2a$, $|B| = 2b$, $|C| = 2c$, $|D| = 2d$. We aim to show that $\text{Mis}_\delta(A) \leq 12k$. We consider two cases, depending on whether $B \neq A$ or $B = A$.

Case $B \neq A$: Let us consider the following intervals (see Fig. 7):

$$I_1 = (0..c] \quad I_2 = (c - \delta..b] \quad I_3 = (c..b + c] \quad I_4 = (2c..2b] \quad I_5 = (b + c..2a].$$

Let $m_q = |\{i \in I_q : i - \delta \in I_q, X_{i-\delta} \not\approx X_i\}|$. We show the following inequalities:

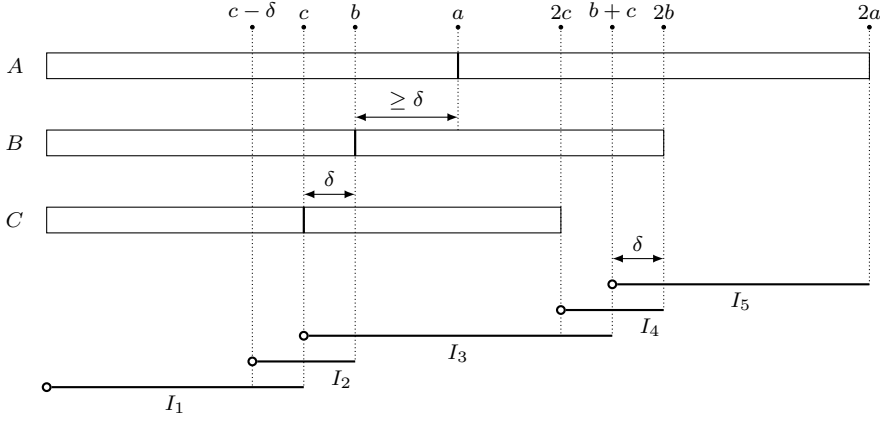


Fig. 7 $B \neq A$

- (I) $m_1 \leq k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap I_1$. Note that $X_i \approx X_{i+c} \approx X_{i+c-b} = X_{i-\delta}$ due to p-squares B and C , respectively. Hence, $X_i \not\approx X_{i-\delta}$ may hold only if $X_{i+c} = \diamond$.
- (II) $m_3 \leq k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap I_3$. Note that $b < i \leq b+c$. Hence, $X_i \approx X_{i-b} \approx X_{i-b+c} = X_{i-\delta}$ due to p-squares B and C , respectively. Consequently, $X_i \not\approx X_{i-\delta}$ may hold only if $X_{i-b} = \diamond$.
- (III) $m_4 \leq m_1 + k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap I_4$. Note that $a < i - \delta < i \leq 2a$. Let $J = (2c - a..2b - a)$. Note that $X[I_4] \approx X[J]$ due to p-square A and that $J \subseteq I_1$. We apply Lemma 2(c) to $X[I_4]$ and $X[J]$ to conclude.
- (IV) $m_2 \leq m_4 + k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap I_2$. Note that $c - \delta < i - \delta < i \leq b$. Note that $X[I_2] \approx X[I_4]$ due to p-square B . We apply Lemma 2(c) to $X[I_2]$ and $X[I_4]$ to conclude.
- (V) $m_4 + m_5 \leq m_1 + m_2 + m_3 + 2k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap (I_4 \cup I_5)$. Note that $a < i - \delta < i \leq 2a$. Let $J = (2c - a..a)$. Note that $X[I_4 \cup I_5] \approx X[J]$ due to p-square A and that $J \subseteq I_1 \cup I_2 \cup I_3$. We apply Lemma 2(b) to $X[I_4 \cup I_5]$ and $X[J]$ to conclude.

We conclude that $|\text{Mis}_\delta(A)| = m_1 + m_2 + m_3 + (m_4 + m_5) \leq k + 3k + k + 7k = 12k$.

Case $B = A$: Let us consider the following intervals (see Fig. 8):

$$I'_1 = (0..c] \quad I'_2 = (c - \delta..b] \quad I'_3 = (c..b + c] \quad I'_4 = (2c..2b].$$

Let $m'_q = |\{i \in I'_q : i - \delta \in I'_q, X_{i-\delta} \not\approx X_i\}|$. We show the following inequalities:

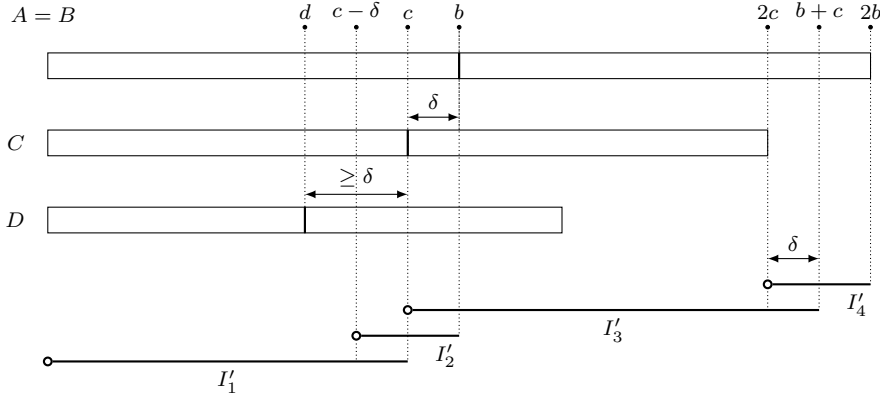


Fig. 8 $B = A$

- (I) $m'_1 \leq k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap I'_1$. Note that $X_i \approx X_{i+c} \approx X_{i+c-b} = X_{i-\delta}$ due to p-squares B and C , respectively. Hence, $X_i \not\approx X_{i-\delta}$ may hold only if $X_{i+c} = \diamond$.
- (II) $m'_3 \leq k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap I'_3$. Note that $b < i < b+c$. Hence, $X_i \approx X_{i-b} \approx X_{i-b+c} = X_{i-\delta}$ due to p-squares B and C , respectively. Consequently, $X_i \not\approx X_{i-\delta}$ may hold only if $X_{i-b} = \diamond$.
- (III) $m'_2 \leq m'_1 + k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap I'_2$. Note that $d < c - \delta < i - \delta < i \leq b \leq 2d$. Let $J = (c - \delta - d..b - d]$. Note that $X[I'_2] \approx X[J]$ due to p-square D and that $J \subseteq I'_1$. We apply Lemma 2(c) to $X[I'_2]$ and $X[J]$ to conclude.
- (IV) $m'_4 \leq m'_2 + k$:
 Assume that $i \in \text{Mis}_\delta(A) \cap I'_4$. Note that $X[I'_4] \approx X[I'_2]$ due to p-square B . We apply Lemma 2(c) to $X[I'_4]$ and $X[I'_2]$ to conclude.

We conclude that $|\text{Mis}_\delta(A)| = m'_1 + m'_2 + m'_3 + m'_4 \leq k + 2k + k + 3k = 7k$. \square

Recall that a deterministic period of a partial word X is an integer q such that there exists a (solid) word W such that $W \approx X$ and W has a period q . In the following lemma we show that if the set $\text{LONG}(X)$ is large enough, then the majority of its elements have strong periodic properties.

Lemma 9 *Let X be a partial word with k holes. Assume that the set $\text{LONG}(X)$ contains at least $16k + 3$ elements. Then $\delta = \text{mingap}(\text{LONG}(X))/2$ is a deterministic period of all p-squares from $\text{LONG}(X)$ excluding possibly the $2k + 1$ longest ones.*

Proof Let $\text{LONG}'(X)$ be the set $\text{LONG}(X)$ without the $2k + 1$ longest elements, A be the longest p-square in $\text{LONG}(X)$, and B be the longest p-square in $\text{LONG}'(X)$. We start by a proof of a weaker property. In the proof we will use the fact that $|\text{Mis}_\delta(A)| \leq 12k$ (Lemma 8).

Claim δ is a quantum period of B .

Proof Assume to the contrary that B does not have quantum period δ , i.e., that $\text{Mis}_\delta(B) \neq \emptyset$. Let i be the minimum index in $\text{Mis}_\delta(B)$.

1. Let us count the p-squares from $LONG(X)$ that contain the position i in the first half. Let $C \in LONG(X)$, $|C| = 2c$, be such a p-square. Then $X_{i+c} \approx X_i \not\approx X_{i-\delta} \approx X_{i+c-\delta}$. Hence, either at least one of the positions X_{i+c} and $X_{i+c-\delta}$ contains a hole ($2k$ possibilities), or $X_{i+c} \not\approx X_{i+c-\delta}$ which means that $i \in \text{Mis}_\delta(A)$ ($12k$ possibilities due to Lemma 8). Therefore, there can be at most $14k$ such p-squares.

2. Let us count the p-squares from $LONG(X)$ that contain $i - \delta$ in the first half and i in the second half. There can be at most one such p-square. Otherwise there would be two p-squares in $LONG(X)$ whose halves' lengths differ by less than δ , contradicting the definition of δ .

3. Let us count the p-squares from $LONG(X)$ that contain both positions $i - \delta$ and i in the second half. Let $C \in LONG(X)$, $|C| = 2c$, be such a p-square. Then $X_{i-c} \approx X_i \not\approx X_{i-\delta} \approx X_{i-c-\delta}$. Hence, at least one of the positions X_{i-c} and $X_{i-c-\delta}$ contains a hole (they cannot form a mismatch, as i was selected as the minimal index). This gives $2k$ possibilities for such a p-square.

4. We will show that there are no p-squares from $LONG(X)$ that do not contain the position i . If such a p-square existed, then we would have $|X|/2 < i - \delta < i \leq |B|$, so $i - \delta$ and i would be contained in right halves of all p-squares that are at least as long as B . There are $2k + 1$ of them, which contradicts point 3.

Each p-square in $LONG(X)$ accounts to one of the categories 1-4. We have shown that there can be at most $16k + 1$ p-squares in $LONG(X)$ which contradicts the assumptions of the lemma. This completes the proof of the claim. \square

Now we strengthen the previous claim and prove that δ is a deterministic period of B . This will conclude the proof since all the p-squares in $LONG'(X)$ are prefixes of B .

Assume that this is not true and let d be minimal such that $B_{i-d\delta} \not\approx B_i$ and let i be the minimal such index i . Hence, $B_{i-\delta} = \dots = B_{i-(d-1)\delta} = \diamond$. Therefore, $d \leq k + 1$, and by the claim, $d \geq 2$. Moreover, $k > 0$.

1. Let us count the p-squares $C \in LONG(X)$, $|C| = 2c$, that contain i in the first half. Let $j = i + c$. If $j > |B|$, then $C \in LONG(X) \setminus LONG'(X)$ and there are $2k + 1$ such p-squares. Otherwise, there can be at most $3k$ p-squares $C \in LONG'(X)$ for which any of the positions $j - d\delta$, $j - \delta$, j contains a hole. Assume otherwise. Then $B_{j-d\delta} = B_{i-d\delta} \not\approx B_i = B_j$ and $B_{j-\delta} \neq \diamond$. Hence, $B_{j-\delta} \not\approx B_{j-d\delta}$ or $B_{j-\delta} \not\approx B_j$, either of which contradicts the way d was selected. In total, there can be $5k + 1$ of the considered p-squares.

2. Let us count the p-squares from $LONG(X)$ that contain $i - d\delta$ in the first half and i in the second half. There can be at most d of them, as otherwise there would be two p-squares in $LONG(X)$ whose halves' lengths differ by less than δ , a contradiction. Hence, the number of such p-squares is at most $k + 1$.

3. Let us count the p-squares $C \in LONG(X)$, $|C| = 2c$, that contain both positions $i - d\delta$ and i in the second half. Let $j = i - c$. There can be at most $2k$ such p-squares C for which any of the positions $j - d\delta$, j contains a hole. Assume otherwise. Then $B_{j-d\delta} \not\approx B_j$ which contradicts the definition of i .

4. Let us count the p-squares that contain the position $i - d\delta$ in the second half and do not contain the position i . Using the same argument as in 2, we see that there are at most $k + 1$ of them.

5. Finally, we will show that there are no p-squares in $LONG(X)$ that do not contain the position $i - d\delta$. If such a p-square existed, then both positions $i - d\delta$ and i would be contained in right halves of all p-squares from $LONG(X) \setminus LONG'(X)$. There are $2k + 1$ of them, which contradicts point 3.

Each p-square in $LONG(X)$ accounts to one of the categories 1-5. We have shown that there can be at most $9k + 3$ p-squares in $LONG(X)$ which contradicts the assumptions of the lemma, as $k > 0$. This completes the proof of the lemma. \square

By $\mathcal{U}\text{-Pref}(X)$ we denote the set of unambiguous p-squares in $LONG(X)$ that occur in X only as a prefix.

Lemma 10 *Let X be a partial word with k holes. Then $|\mathcal{U}\text{-Pref}(X)| < 16k + 3$.*

Proof Assume to the contrary that $|\mathcal{U}\text{-Pref}(X)| \geq 16k + 3$. Let us recall that $\mathcal{U}\text{-Pref}(X) \subseteq LONG(X)$ so the assumptions of Lemma 9 are satisfied.

Let $\mathcal{U}\text{-Pref}'(X)$ be the set $\mathcal{U}\text{-Pref}(X)$ without the $2k + 1$ longest elements. By Lemma 9, each p-square in $\mathcal{U}\text{-Pref}'(X)$ has a deterministic period $\delta = \text{mingap}(LONG(X))/2$.

Let us assume that $B = X[1..2a] \in \mathcal{U}\text{-Pref}'(X)$ and let W^2 be its (solid) representative. Then $C = X[1 + \delta..2a + \delta]$ is a p-square, as it matches W^2 due to the deterministic period δ . If $X[2a + 1..2a + \delta]$ did not contain a hole, then C would be another occurrence of a u-square with representative W^2 . This would contradict the assumption that $B \in \mathcal{U}\text{-Pref}(X)$.

Note that the fragments of the form $X[2a + 1..2a + \delta]$ for $X[1..2a] \in \mathcal{U}\text{-Pref}'(X)$ are pairwise disjoint due to the definition of δ . What follows is that $|\mathcal{U}\text{-Pref}'(X)| \leq k$ and $|\mathcal{U}\text{-Pref}(X)| \leq 3k + 1$, a contradiction. \square

We say that a solid square W^2 has a *solid occurrence* in T if T contains a factor *equal to* W^2 . By the following fact, there are at most $2n$ non-equivalent p-square factors of T with solid occurrences.

Fact 15 ([14, 18, 11]) *Every position of a (solid) word contains at most two rightmost occurrences of squares.*

In the proof of the upper bound on the number of u-squares we separately count u-squares that have a solid occurrence and those that do not. In the latter case, we use Lemma 10, which lets us bound $|\mathcal{U}\text{-Pref}(X)|$ by $19k$ in case that $k > 0$.

Theorem 16 *If T is a partial word of length n with k holes, then $usquares(T) = \mathcal{O}(nk)$.*

Proof Let us recall that by $\mathcal{B}(T)$ we denote the set of all basic factors of T . If $T[i..i + \ell - 1]$ is a rightmost occurrence of a u-square V , then $V \in \mathcal{U}\text{-Pref}(W)$ for some $W \in \mathcal{B}(T)$. (In particular, $W = T_{i,q}$ for $q = \lceil \log \ell \rceil$.) This lets us bound $usquares(T)$ as follows:

$$\begin{aligned} usquares(T) &\leq \sum_{W \in \mathcal{B}(T)} |\mathcal{U}\text{-Pref}(W)| \\ \{\text{Fact 15}\} &\leq 2n + \sum_{W \in \mathcal{B}(T): \#_{\diamond}(W) > 0} |\mathcal{U}\text{-Pref}(W)| \\ \{\text{Lemma 10}\} &< 2n + 19 \sum_{W \in \mathcal{B}(T)} \#_{\diamond}(W) \\ \{\text{Lemma 1}\} &\leq 2n + 38nk. \end{aligned}$$

This concludes the proof. \square

Acknowledgements. Tomasz Kociumaka is supported by Polish budget funds for science in 2013-2017 as a research project under the ‘Diamond Grant’ program, grant no. DI2012 017942. Jakub Radoszewski, Wojciech Rytter, and Tomasz Waleń are supported by the Polish National Science Center, grant no. 2014/13/B/ST6/00770.

References

1. Bannai, H., Inenaga, S., Köppl, D.: Computing all distinct squares in linear time for integer alphabets. In: J. Kärkkäinen, J. Radoszewski, W. Rytter (eds.) 28th Annual Symposium on Combinatorial Pattern Matching, CPM 2017, *LIPICs*, vol. 78, pp. 22:1–22:18. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2017). DOI 10.4230/LIPICs.CPM.2017.22
2. Blanchet-Sadri, F., Bodnar, M., Nikkel, J., Quigley, J.D., Zhang, X.: Squares and primitivity in partial words. *Discrete Applied Mathematics* **185**, 26–37 (2015). DOI 10.1016/j.dam.2014.12.003
3. Blanchet-Sadri, F., Jiao, Y., Machacek, J.M., Quigley, J., Zhang, X.: Squares in partial words. *Theoretical Computer Science* **530**, 42–57 (2014). DOI 10.1007/978-3-642-31653-1_36
4. Blanchet-Sadri, F., Mercaş, R.: A note on the number of squares in a partial word with one hole. *Informatique Théorique et Applications* **43**(4), 767–774 (2009). DOI 10.1051/ita/2009019
5. Blanchet-Sadri, F., Mercaş, R.: The three-squares lemma for partial words with one hole. *Theoretical Computer Science* **428**, 1–9 (2012). DOI 10.1016/j.tcs.2012.01.012
6. Blanchet-Sadri, F., Mercaş, R., Scott, G.: Counting distinct squares in partial words. *Acta Cybernetica* **19**(2), 465–477 (2009)
7. Blanchet-Sadri, F., Nikkel, J., Quigley, J.D., Zhang, X.: Computing primitively-rooted squares and runs in partial words. In: J. Kratochvíl, M. Miller, D. Froncek (eds.) *Combinatorial Algorithms, IWOC A 2014, Lecture Notes in Computer Science*, vol. 8986, pp. 86–97. Springer (2014). DOI 10.1007/978-3-319-19315-1_8
8. Charalampopoulos, P., Crochemore, M., Iliopoulos, C.S., Kociumaka, T., Pissis, S.P., Radoszewski, J., Rytter, W., Waleń, T.: Efficient enumeration of non-equivalent squares in partial words with few holes. In: Y. Cao, J. Chen (eds.) *Computing and Combinatorics - 23rd International Conference, COCOON 2017, Proceedings, Lecture Notes in Computer Science*, vol. 10392, pp. 99–111. Springer (2017). DOI 10.1007/978-3-319-62389-4_9

9. Crochemore, M., Iliopoulos, C.S., Kubica, M., Radoszewski, J., Rytter, W., Waleń, T.: Extracting powers and periods in a word from its runs structure. *Theoretical Computer Science* **521**, 29–41 (2014). DOI 10.1016/j.tcs.2013.11.018
10. Crochemore, M., Rytter, W.: Squares, cubes, and time-space efficient string searching. *Algorithmica* **13**(5), 405–425 (1995). DOI 10.1007/BF01190846
11. Deza, A., Franek, F., Thierry, A.: How many double squares can a string contain? *Discrete Applied Mathematics* **180**, 52–69 (2015). DOI 10.1016/j.dam.2014.08.016
12. Diaconu, A., Manea, F., Tiseanu, C.: Combinatorial queries and updates on partial words. In: M. Kutylowski, W. Charatonik, M. Gębala (eds.) *Fundamentals of Computation Theory, FCT 2009, Lecture Notes in Computer Science*, vol. 5699, pp. 96–108. Springer (2009). DOI 10.1007/978-3-642-03409-1_10
13. Farach, M.: Optimal suffix tree construction with large alphabets. In: FOCS, pp. 137–143. IEEE Computer Society (1997)
14. Fraenkel, A.S., Simpson, J.: How many squares can a string contain? *Journal of Combinatorial Theory. Series A* **82**(1), 112–120 (1998). DOI 10.1006/jcta.1997.2843
15. Gusfield, D.: *Algorithms on Strings, Trees, and Sequences - Computer Science and Computational Biology*. Cambridge University Press (1997)
16. Gusfield, D., Stoye, J.: Linear time algorithms for finding and representing all the tandem repeats in a string. *Journal of Computer and System Sciences* **69**(4), 525–546 (2004). DOI 10.1016/j.jcss.2004.03.004
17. Halava, V., Harju, T., Kärki, T.: On the number of squares in partial words. *RAIRO – Theoretical Informatics and Applications* **44**(1), 125–138 (2010). DOI 10.1051/ita/2010008
18. Ilie, L.: A simple proof that a word of length n has at most $2n$ distinct squares. *Journal of Combinatorial Theory. Series A* **112**(1), 163–164 (2005). DOI 10.1016/j.jcta.2005.01.006
19. Kociumaka, T.: Minimal suffix and rotation of a substring in optimal time. In: R. Grossi, M. Lewenstein (eds.) *Combinatorial Pattern Matching, CPM 2016, LIPIcs*, vol. 54, pp. 28:1–28:12. Schloss Dagstuhl (2016). DOI 10.4230/LIPIcs.CPM.2016.28
20. Main, M.G., Lorentz, R.J.: An $O(n \log n)$ algorithm for finding all repetitions in a string. *Journal of Algorithms* **5**(3), 422 – 432 (1984). DOI [https://doi.org/10.1016/0196-6774\(84\)90021-X](https://doi.org/10.1016/0196-6774(84)90021-X)
21. Manea, F., Mercaş, R., Tiseanu, C.: An algorithmic toolbox for periodic partial words. *Discrete Applied Mathematics* **179**, 174–192 (2014). DOI 10.1016/j.dam.2014.07.017
22. Manea, F., Tiseanu, C.: Hard counting problems for partial words. In: A. Dediu, H. Fernau, C. Martín-Vide (eds.) *Language and Automata Theory and Applications, LATA 2010, Lecture Notes in Computer Science*, vol. 6031, pp. 426–438. Springer (2010). DOI 10.1007/978-3-642-13089-2_36