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

Panagiotis Charalampopoulos, Maxime Crochemore, Costas Iliopoulos, Tomasz Kociumaka, Solon Pissis, Jakub Radoszewski, Wojciech Rytter, Tomasz Waleń

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

Panagiotis Charalampopoulos[1], Maxime Crochemore[1,2], Costas S. Iliopoulos[1], Tomasz Kociumaka[3,⋆], Solon P. Pissis[1], Jakub Radoszewski[1,3,⋆⋆], Wojciech Rytter[3,⋆⋆⋆], and Tomasz Waleń[3,⋆⋆⋆]

[1] Department of Informatics, King's College London, London, UK
[panagiotis.charalampopoulos,maxime.crochemore,
costas.iliopoulos,solon.pissis]@kcl.ac.uk
[2] Université Paris-Est, France
[3] Faculty of Mathematics, Informatics and Mechanics,
University of Warsaw, Warsaw, Poland
[kociumaka,jrad,rytter,walen]@mimuw.edu.pl

Abstract. A partial word is a word with holes (also called don't cares: special symbols which match any symbol). A *p-square* is a partial word matching at least one standard square without holes (called a *full square*). Two p-squares are called *equivalent* if they match the same sets of full squares. Denote by *psquares*(T) the number of non-equivalent p-squares which are subwords of a partial word T. Let $\mathrm{PSQUARES}_k(n)$ be the maximum value of *psquares*(T) over all partial words of length n with k holes. We show asympthotically tight bounds:

$$c_1 \cdot \min(nk^2,\, n^2) \le \mathrm{PSQUARES}_k(n) \le c_2 \cdot \min(nk^2,\, n^2)$$

for some constants $c_1, c_2 > 0$. We also present an algorithm that computes *psquares*(T) in $\mathcal{O}(nk^3)$ time for a partial word T of length n with k holes. 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-squares.

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 UU, for some word U, is called a *square*. For a word W, 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 of length n of may contain $\Theta(n^2)$ square factors (e.g. a^n), however, it is known that such a word contains

⋆ Supported by Polish budget funds for science in 2013-2017 as a research project under the 'Diamond Grant' program.

⋆⋆ The author is a Newton International Fellow.

⋆⋆⋆ Supported by the Polish National Science Center, grant no 2014/13/B/ST6/00770.

only $\mathcal{O}(n)$ different square factors [16,20]; currently the best known upper bound is $\frac{11}{6}n$ [14]. Moreover, all different square factors of a word can be listed in $\mathcal{O}(n)$ time using two different approaches [17,11].

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 $U \approx V$. We define the *length* of a p-square as $|U|$. Same as in the context of words, a *p-square factor* of a partial word W is a factor being a p-square; see [2,9]. Alongside [2,8,9], we define a *full square* as a square of a word and *square subword* of a partial word W as a full square that matches some p-square factor of W.

We introduce the notion of *equivalence* of p-square factors in partial words. By $\textit{sq-val}(UV)$ let us denote the set of different full squares that match the partial word UV: $\textit{sq-val}(UV) = \{XX : X \in \Sigma^*, XX \approx UV\}$. Then p-squares UV and $U'V'$ are called *equivalent* if $\textit{sq-val}(UV) = \textit{sq-val}(U'V')$ (denoted as $UV \equiv U'V'$). E.g., the following two sets are equivalence classes of p-squares: $\mathcal{A} = \{aa\diamond\diamond,\ \diamond aa\diamond,\ \diamond\diamond aa,\ a\diamond\diamond a\}$ and $\mathcal{B} = \{a\diamond a\diamond\diamond\diamond,\ \diamond\diamond aa\diamond\diamond,\ \ldots\}$. Note that two p-square factors of a partial word W are equivalent in this sense iff they correspond to exactly the same sets of square subwords. Our work is devoted to enumeration of *non-equivalent p-square factors* of a partial word with a given number k of holes.

We say that WW is a *general form* of a square UV (see [8]), denoted as $\textit{repr}(UV)$, if $WW \approx UV$ and $\textit{sq-val}(WW) = \textit{sq-val}(UV)$ (in other words, W is the "most general" partial word that matches both U and V). Then $UV \equiv U'V'$ iff $\textit{repr}(UV) = \textit{repr}(U'V')$. E.g., $\textit{repr}(A) = (aa)^2$ for all $A \in \mathcal{A}$ and $\textit{repr}(B) = (a\diamond a)^2$ for all $B \in \mathcal{B}$.

Previous study of p-squares in partial words was mostly focused on their combinatorics. It started with the case of $k = 1$ hole [8], in which case different square subwords correspond to non-equivalent p-square factors. It was shown that a partial word with one hole contains at most $\frac{7}{2}n$ different square subwords [6] (later this bound was improved to $3n$ for binary partial words [19]). Also a generalization of the three squares lemma for words [12] was proposed for partial words [7].

As for a larger number of holes, the study was devoted mainly to counting the number of different square subwords of a partial word [8,2] or all occurrences of p-square factors [4,2]. Similarly, on the algorithmic side, [25] proved that the problem of counting different square subwords of a partial word is #P-complete and [15,24] and [9] 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. Other work includes avoidance of squares [18,3] or abelian squares [5] in partial words.

1.1 Our Results

We present the following combinatorial bounds and efficient algorithms related to enumeration of non-equivalent p-square factors of a partial word of length n with k of holes. A length p is called *ambiguous* if there are two holes at a distance p. Otherwise it is called *unambiguous*.

Combinatorial results. We prove that a partial word of length n with k holes contains $\mathcal{O}(nk^2)$ non-equivalent p-squares. We also show an example of a partial word of length n with k holes that contains $\Omega(nk^2)$ non-equivalent p-squares of ambiguous lengths and that contains $\Omega(nk)$ non-equivalent p-squares of unambiguous lengths. This work can be viewed as a generalization of the results on partial words with one hole [8,6,19] to $k > 1$ holes.

Algorithmic results. We present an algorithm that reports all non-equivalent p-squares in a partial word 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-squares. Our algorithm generalizes the approach of [11] and proposes, as an important tool, a non-obvious extension of the notion of runs to partial words (another definition of runs in partial words appeared in [9]).

2 Periodicity of Words and Partial Words

A word T is a sequence of letters over an alphabet Σ. By $|T| = n$ we denote the length of T, and by $T[i]$, for $i = 1, \ldots, n$, the ith letter of T. For $1 \le i \le j \le n$, $T[i..j]$ denotes the *factor* of T equal to $T[i] \ldots T[j]$. A positive integer q is called a *period* of T if $T[i] = T[i+q]$ for all $i = 1, \ldots, n-q$. In this case, $T[1..q]$ is called a *string period* of T. Two equal-length words S and T are called *cyclic shifts* if there exists an index i such that $S[i..|S|]S[1..i-1] = T$.

A *run* (also called a maximal repetition) in T is a triple (a, b, q) such that $T[a..b]$ has the shortest period q, $2q \le b - a + 1$, and the interval cannot be extended to the left nor to the right without violating the above property, that is, $T[a-1] \neq T[a+q-1]$ and $T[b-q+1] \neq T[b+1]$, provided that the respective letters exist. The *exponent* of a run is defined as $\frac{b-a+1}{q}$. A word of length n has at most n runs and they can all be computed in $\mathcal{O}(n)$ time [22,1].

From a run (a, b, q) we can produce all triples (a, b, kq) for integer k such that $2kq \le b - a + 1$; we call such triples *generalized runs*. In other words, the period specified in a generalized run need not be the shortest period of the fragment. The number of generalized runs is also $\mathcal{O}(n)$, as the sum of exponents of runs is $\mathcal{O}(n)$ [22,1].

A partial word is a sequence of symbols from $\Sigma' = \Sigma \cup \{\diamond\}$. For a partial word T we use the same notation as for words: $|T|$ for length, $T[i]$ for the ith letter, $T[i..j]$ for a factor. The relation of matching on Σ' is defined as $a \approx a$, $a \approx \diamond$ for all $a \in \Sigma'$. We define an operation $\wedge$ such that $a \wedge a = a$, $a \wedge \diamond = a$ for all $a \in \Sigma'$, and otherwise $a \wedge b$ is undefined. Two equal-length partial words T and S are said to *match* (denoted as $T \approx S$) if $T[i] \approx S[i]$ for all $i = 1, \dots, |T|$. In this case, by $S \wedge T$ we denote the partial word $S[1] \wedge T[1], \dots, S[|S|] \wedge T[|S|]$. Note that if UV is a p-square, then $repr(UV) = (U \wedge V)^2$.

A *quantum period* of a partial word T is a positive integer q such that $T[i] \approx T[i+q]$ for all $i = 1, \dots, |T| - q$. T is called *quantum periodic* with a quantum period q if $2q \leq |T|$. Let T be a partial word of length n. We say that a triple (a, b, q) is a *quantum generalized run* (Q-run, for short) in T if $T[a..b]$ is quantum periodic with period q and none of the partial words $T[a-1..b]$ and $T[a..b+1]$ (if exists) has the quantum period q; see Example 1 and Fig. 2.

Example 1. The partial word $T = caa\diamond\diamond\diamond\diamond\diamond bbd$ contains one Q-run with period 2: $(2, 10, 2)$ that corresponds to the factor $aa\diamond\diamond\diamond\diamond\diamond bb$.

Generalized runs in standard words are strongly related to squares: (1) every square belongs to a generalized run and, moreover, (2) all factors of length $2q$ of a generalized run with period q are squares being each other's cyclic shifts. Unfortunately, Q-runs in partial words have only property (1). However, we can introduce a type of run in partial words for which both properties (1) and (2) hold. A *pseudorun* (P-run, in short) is a triple (a, b, q) such that:

(a) $T[a..b]$ is quantum periodic with period q
(b) $T[i] \wedge T[i+q] = T[i+q] \wedge T[i+2q]$ for all i such that $i, i+2q \in [a, b]$,
(c) none of the partial words $T[a-1..b]$ and $T[a..b+1]$ (if exists) satisfies the conditions (a) and (b).

We say that a p-square $T[c..d]$ is *induced* by the P-run if the length of the p-square is q and $[c, d] \subseteq [a, b]$.

Observation 2. *(1) Every p-square factor in T is induced by a P-run. (1) All factors of length $2q$ of a P-run with period q are p-squares and their representatives are each other's cyclic shifts.*

Proof. (1) Let $T[i..j]$ be a p-square factor of length $q = (j-i+1)/2$ in T. Initially we set $a = i$, $b = j$; then (a, b, q) satisfies conditions (a) and (b) of a pseudorun (the latter one trivially). Now we extend (a, b, q) until it becomes maximal under the two conditions, i.e., decrement a while $T[a-1] \wedge T[a+q-1] = T[a+q-1] \wedge T[a+2q-1]$, and same for b.

(2) Every factor of length $2q$ of a P-run is quantum periodic with period q, hence a p-square. Consider two such consecutive factors $X = T[i..i+2q-1]$ and $Y = T[i+1..i+2q]$. Then $repr(X) = T[i..i+q-1] \wedge T[i+q..i+2q-1]$ and

$$\begin{aligned} repr(Y) &= T[i+1..i+q] \wedge T[i+q+1..i+2q] \\ &= T[i+1..i+q-1] \wedge T[i+q+1..i+2q-1], T[i+q] \wedge T[i+2q] \\ &= T[i+1..i+q-1] \wedge T[i+q+1..i+2q-1], T[i] \wedge T[i+q] \end{aligned}$$

where the last equality is due to condition (b) of a P-run. Consequently, $repr(X)$ and $repr(Y)$ are cyclic shifts. □

Example 3. Let $T = caa\lozenge\lozenge\lozenge\lozenge\lozenge bbd$. Then T contains five P-runs with period 2: $(2,5,2)$, $(3,6,2)$, $(4,8,2)$, $(6,9,2)$ and $(7,10,2)$, that correspond to factors: $aa\lozenge\lozenge$, $a\lozenge\lozenge\lozenge$, $\lozenge\lozenge\lozenge\lozenge\lozenge$, $\lozenge\lozenge\lozenge b$, and $\lozenge\lozenge bb$, respectively. The squares induced by the respective P-runs are as follows: $aa\lozenge\lozenge$; $a\lozenge\lozenge\lozenge$; $\lozenge\lozenge\lozenge\lozenge$ (two times); $\lozenge\lozenge\lozenge b$; and $\lozenge\lozenge bb$.

3 Combinatorial Bounds

Theorem 4. *There exists a partial word of length n with k holes that contains $\Omega(nk^2)$ non-equivalent p-squares, for $k = \mathcal{O}(\sqrt{n})$.*

Proof. Assume that $2 \mid k$. Consider the following partial word:

$$T_{m,k} = a^m \lozenge^{k/2} a^m (a^{k/2-1}\lozenge)^{k/2} a^{3m}$$

of length $n = 4m + (\frac{k}{2})^2 + \frac{k}{2}$ over $\Sigma = \{a, b\}$; here $m = \Theta(n)$. Point (a) follows from the next claim.

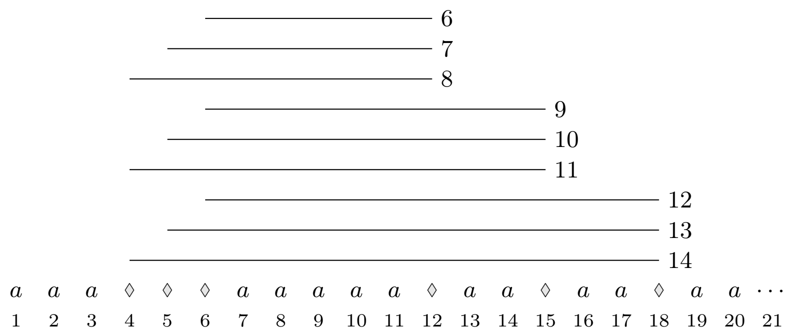

Fig. 1. Ambiguous lengths in $T_{3,6}$.

Claim. For some $k = \mathcal{O}(\sqrt{n})$, $T_{m,k}$ contains $\Theta(mk^2)$ non-equivalent p-squares of ambiguous lengths.

Proof. First, note that that there are $(\frac{k}{2})^2$ different ambiguous lengths and that for each such length ℓ, there is exactly one pair of holes at this distance in $T_{m,k}$. Indeed, consider the i-th hole in the $\lozenge^{k/2}$ part and the j-th hole in the $(a^{k/2-1}\lozenge)^{k/2}$ part $(1 \le i, j \le \frac{k}{2})$. Then the distance between them is $\frac{k}{2} - i + m + \frac{k}{2} \cdot j = \frac{k}{2} \cdot j - i + (\frac{k}{2} + m)$. Finally, note that for i and j in the considered range, values of the form $\frac{k}{2} \cdot j - i$ are all distinct (this is a known example of a differenece cover); see Fig. 1.

Let us choose k so that $2(m + (\frac{k}{2})^2 + \frac{k}{2}) \le \frac{n}{2}$. Then for each ambiguous length ℓ, we have exactly ℓ p-square factors in $T_{m,k}$ of the general form $(a^i \lozenge a^{\ell-i-1})$. As $\ell \ge m = \Theta(n)$, this concludes the proof. $\square$

(b) ... $\square$

Now we proceed to the upper bound proof. Let us fix a partial word T of length n with k holes. Let $holes(a,b)$ denote the number of holes in $T[a..b]$.

We say that a square subword has a *solid occurrence* if... By the following fact, there are $\mathcal{O}(n)$ square subwords of T with solid occurrences.

Fact 5 ([16,20,14]). *≤ 2 rightmost occurrences.*

Our upper bound for partial words is based on the following key lemma that generalizes Fact 5. We call an interval $[\ell, r) \subseteq [1, n]$ is *short* if $r \leq \frac{3k+3}{3k+2}\ell$.

Lemma 6. *For a short interval $[\ell, r)$, there are $\mathcal{O}(k')$, where $k' = holes(i, i + 2r - 1)$, p-squares of unambiguous lengths without solid occurrences in $[\ell, r]$ that have their rightmost occurrence in T at position i.*

Proof. Let us start with the following claim.

Claim 7. *There exists an interval $I = [a, b] \subseteq [i, i+\ell-1]$ such that $holes(I) = 0$, $holes(J) = 0$ where $J = [a + \ell, b + r)$, and $I \geq 2(r - \ell)$.*

Proof. A position i cannot belong to I if $T[i]$ is a hole or $T[i + \ell..i + r]$ contains a hole. In total, a hole in $T[i..i + \ell - 1]$ excludes one positions in $[i, i + \ell - 1]$ and a hole in $T[i + \ell..i + 2r - 1]$ excludes at most $r - \ell + 1$ consecutive positions in $[i, i + \ell - 1]$. By the pigeonhole principle, there exists a non-excluded fragment in $T[i..i + \ell - 1]$ of consecutive positions of length at least

$$\frac{\ell - k(r - l)}{k + 1} \geq \frac{(3k + 2)(r - \ell) - k(r - l)}{k + 1} = 2(r - \ell).$$

$\square$

Let X be the subword of T at positions in I. If there are at least two p-square factors of the considered type starting at position i, then X is periodic (Galil...). Let p be its shortest period.

Let UV be a p-square factor of unambiguous length of the considered type and let $WW = repr(UV)$ There are three cases:

(a) WW has period p
(b) W has period p
(c) W does not have period p.

There is at most one p-square factor UV corresponding to case (a). Assume to the contrary that there are two such p-squares, UV and $U'V'$, of lengths $d < d'$. We have that $p \mid d, d'$ and the generalized forms of the two p-squares have the same string period (as they share a subword U). Hence, $repr(UV)$ is a border of $repr(U'V')$, so UV occurs in T at position $i + d' - d$.

We will show that there are at most $k' + 1$ p-square subwords of type (b). Assume to the contrary that at least $k' + 2$ of them, of lengths $d_1 < \ldots < d_{k'+2}$. We see that $d_j \bmod k$ is the same, as the occurrences of X in the right half of the p-square in T differ by multiples of p. Consider the shortest UV and the longest $U'V'$ these p-squares, with generalized forms WW and $W'W'$, and the subword $T[i+d_1..i+d_{k'+2}-1]$. It matches a prefix P of length $d_{k'+2}-d_1$ of W and a suffix S of the same length of W'. Both P and S have period P; however, their string periods are not equal. Consequently, in every occurrence of the period in $T[i+d_1..i+d_{k'+2}-1]$ there mus be a hole; this makes $(d_{k'+2}-d_1)/p \geq k'+1$ holes in total, a contradiction.

(c) Consider the occurrence of X in W. The periodicity of X does not cover the whole W, so there is a position j in W where the periodicity breaks. Assume w.l.o.g. that j is to the right of X. Consider the positions j_1 and j_2 that correspond to j in the subwords U, V of T. If any of $T[j_1]$ and $T[j_2]$ is not a hole, then it is determined uniquely as the first position where the deterministic period p breaks, starting from the corresponding occurrence of X. Hence, if both of them are not holes, then $|W| = j_2 - j_1$ is uniquely determined. Otherwise, if $T[j_1]$ or $T[j_2]$ is a hole (they cannot be both holes, as the length is unambiguous), then one of j_1, j_2 is uniquely determined and there are at most k' choices for the other. Consequently, there are at most $4k'+2$ such p-squares. □

Theorem 8. *A partial word T of length n with k holes contains $\mathcal{O}(nk^2)$ non-equivalent p-squares.*

Proof. Obviously, in T there are at most k^2 ambiguous lengths. Consequently, there are $\mathcal{O}(nk^2)$ non-equivalent p-squares of such lengths. Let us consider p-squares of unambiguous lengths. By Fact 5, among them there are $\mathcal{O}(n)$ different p-squares with a solid occurrence. From now on we consider only p-squares without a solid occurrence.

Let $[\ell, r]$ be a short interval. By Lemma 6, the total number of different p-squares of unambiguous lengths in $[\ell, r]$ in T is:

$$\mathcal{O}\left(\sum_{i=1}^{n} k_{i,i+2r-1}\right) = \mathcal{O}(rk). \tag{1}$$

The equality is based on the fact that each of the k holes in T is counted in at most $2r$ terms $k_{i,i+2r-1}$.

Let us consider a family of endpoints $r_0, r_1, \ldots$:

$$r_j = \left\lfloor \frac{n}{(1+\frac{1}{3k+2})^j} \right\rfloor = \left\lfloor n\left(\tfrac{3k+2}{3k+3}\right)^j \right\rfloor.$$

We divide the p-square lengths into short intervals $[r_{j+1}+1, r_j]$. By (1), the total number of p-squares in T is:

$$\mathcal{O}\left(\sum_{j=0}^{\infty} r_j k\right) = \mathcal{O}\left(nk\sum_{j=0}^{\infty}\left(\tfrac{3k+2}{3k+3}\right)^j\right) = \mathcal{O}\left(\frac{nk}{1-\frac{3k+2}{3k+3}}\right) = \mathcal{O}(nk^2).$$

□

4 Main Algorithm

Let T be a partial word of length n with k holes.

4.1 Computing Q-runs

We divide Q-runs into *solid Q-runs* that do not contain a hole and the remaining *non-solid Q-runs*. A solid Q-run is a generalized run in a maximal solid factor of T. Thus all solid Q-runs can be computed in $\mathcal{O}(n)$ time via a linear-time algorithm for computing runs in words [22,1].

Non-solid Q-runs are computed with a modification of Main-Lorentz algorithm [23]. Let us first reformulate the algorithm in the language of computing generalized runs in words. For a word S of length n, it finds a representation of all p-squares that contain the position $i = \lfloor n/2 \rfloor$ and then makes recursive computations in $S[1..i]$ and $S[i..n]$. It first computes all the p-squares with first half containing the position i and then computes the remaining p-squares containing the position i symmetrically. For a pair of positions i, j, we define $lcp(i, j)$ as the length of the longest common prefix of $S[i..n]$ and $S[j..n]$ and $lcs(i, j)$ as the length of the longest common suffix of $S[1..i]$ and $S[1..j]$. The algorithm for each position $j > i$ computes $lcp(i, j) + lcs(i, j)$ and, if this value is at least $j - i$, reports a generalized run $(i - lcs(i, j)..j + lcp(i, j), j - i)$. To avoid duplicates, the generalized run can be omitted if it reaches the end of the word in the recursive call which does not coincide with the position 1 or n in the original word. The algorithm's running time is $\mathcal{O}(n \log n)$ as lcp-queries and lcs-queries can be answered in $\mathcal{O}(1)$-time after preprocessing.

For a partial word T, the longest common compatible prefix of two positions i, j, denoted $lccp(i, j)$, is defined as the largest ℓ such that $T[i..i+\ell-1] \approx T[j..j+\ell-1]$. Symmetrically, we can define $lccs$ as the length of the longest common compatible suffix. In [10] it was shown that after $\mathcal{O}(nk)$-time preprocessing, $lccp$-queries (hence, $lccs$-queries) can be answered in $\mathcal{O}(1)$ time.

Thus we could directly apply the Main-Lorentz scheme for partial words; the result would be exactly the set of Q-runs in T. However, this would yield $\mathcal{O}(n \log n)$-time computation. We aim only at computing non-solid Q-runs, which lets us easily reduce the complexity to $\mathcal{O}(n \log k)$. To this end, we only make recursive calls in positions that contain holes, with the recursive call taking place at the position of the $\lceil \frac{k}{2} \rceil$-th hole in T. Together with $\mathcal{O}(n)$-time solid Q-runs computation we arrive at the following.

Lemma 9. *A partial word of length n with k holes contains $\mathcal{O}(n \log k)$ Q-runs and they can all be computed in $\mathcal{O}(n \log k)$ time.*

4.2 Computing Pseudoruns

Observation 10. *If (a, b, p) is a P-run, then there exists a Q-run (a', b', p) such that $a' \leq a \leq b \leq b'$.*

From the set of m Q-runs of period p we can produce the set of all P-runs with period p. The computation time is $\mathcal{O}(m+k)$.

Lemma 11. *A partial word of length n with k holes contains $\mathcal{O}(nk)$ P-runs and they can all be computed in $\mathcal{O}(nk)$ time.*

4.3 Reporting Squares

Assume that the alphabet Σ is ordered and that $\diamond$ is smaller than all the letters from Σ. We use Observation 2. If we group the P-runs by the minimal cyclic shift of the induced p-squares, then we can apply the approach of [11] to count the p-squares in time linear in the number of P-runs and n.

First, note that the minimal cyclic shift value for a P-run (a, b, p) can be computed in $\mathcal{O}(k^2)$ time using Theorem 23 from [21]. Indeed, we can represent the representative of the p-square $T[a..a+2p-1]$ as a concatenation of at most k factors of T and at most k single letters that appear elsewhere in T.

...

Theorem 12. *All non-equivalent p-squares in a partial word of length n with k holes can be reported (as factors of the partial word) in $\mathcal{O}(nk^3)$ time.*

References

1. Bannai, H., I, T., Inenaga, S., Nakashima, Y., Takeda, M., Tsuruta, K.: A new characterization of maximal repetitions by Lyndon trees. CoRR abs/1406.0263 (2014), http://arxiv.org/abs/1406.0263
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), http://dx.doi.org/10.1016/j.dam.2014.12.003
3. Blanchet-Sadri, F., Choi, I., Mercas, R.: Avoiding large squares in partial words. Theor. Comput. Sci. 412(29), 3752–3758 (2011), http://dx.doi.org/10.1016/j.tcs.2011.04.009
4. Blanchet-Sadri, F., Jiao, Y., Machacek, J.M., Quigley, J.D., Zhang, X.: Squares in partial words. Theor. Comput. Sci. 530, 42–57 (2014), http://dx.doi.org/10.1016/j.tcs.2014.02.023
5. Blanchet-Sadri, F., Kim, J.I., Mercas, R., Severa, W., Simmons, S.: Abelian square-free partial words. In: Dediu et al. [13], pp. 94–105, http://dx.doi.org/10.1007/978-3-642-13089-2_8
6. Blanchet-Sadri, F., Mercas, R.: A note on the number of squares in a partial word with one hole. ITA 43(4), 767–774 (2009), http://dx.doi.org/10.1051/ita/2009019
7. Blanchet-Sadri, F., Mercas, R.: The three-squares lemma for partial words with one hole. Theor. Comput. Sci. 428, 1–9 (2012), http://dx.doi.org/10.1016/j.tcs.2012.01.012
8. Blanchet-Sadri, F., Mercas, R., Scott, G.: Counting distinct squares in partial words. Acta Cybern. 19(2), 465–477 (2009), http://www.inf.u-szeged.hu/actacybernetica/edb/vol19n2/BlanchetSadri_2009_ActaCybernetica.xml

9. Blanchet-Sadri, F., Nikkel, J., Quigley, J.D., Zhang, X.: Computing primitively-rooted squares and runs in partial words. In: Kratochvíl, J., Miller, M., Froncek, D. (eds.) Combinatorial Algorithms - 25th International Workshop, IWOCA 2014, Duluth, MN, USA, October 15-17, 2014, Revised Selected Papers. Lecture Notes in Computer Science, vol. 8986, pp. 86–97. Springer (2014), http://dx.doi.org/10.1007/978-3-319-19315-1_8
10. Crochemore, M., Iliopoulos, C.S., Kociumaka, T., Kubica, M., Langiu, A., Radoszewski, J., Rytter, W., Szreder, B., Waleń, T.: A note on the longest common compatible prefix problem for partial words. J. Discrete Algorithms 34, 49–53 (2015), http://dx.doi.org/10.1016/j.jda.2015.05.003
11. Crochemore, M., Iliopoulos, C.S., Kubica, M., Radoszewski, J., Rytter, W., Waleń, T.: Extracting powers and periods in a word from its runs structure. Theor. Comput. Sci. 521, 29–41 (2014), http://dx.doi.org/10.1016/j.tcs.2013.11.018
12. Crochemore, M., Rytter, W.: Sqares, cubes, and time-space efficient string searching. Algorithmica 13(5), 405–425 (1995), http://dx.doi.org/10.1007/BF01190846
13. Dediu, A., Fernau, H., Martín-Vide, C. (eds.): Language and Automata Theory and Applications, 4th International Conference, LATA 2010, Trier, Germany, May 24-28, 2010. Proceedings, Lecture Notes in Computer Science, vol. 6031. Springer (2010), http://dx.doi.org/10.1007/978-3-642-13089-2
14. Deza, A., Franek, F., Thierry, A.: How many double squares can a string contain? Discrete Applied Mathematics 180, 52–69 (2015), http://dx.doi.org/10.1016/j.dam.2014.08.016
15. Diaconu, A., Manea, F., Tiseanu, C.: Combinatorial queries and updates on partial words. In: Kutylowski, M., Charatonik, W., Gebala, M. (eds.) Fundamentals of Computation Theory, 17th International Symposium, FCT 2009, Wroclaw, Poland, September 2-4, 2009. Proceedings. Lecture Notes in Computer Science, vol. 5699, pp. 96–108. Springer (2009), http://dx.doi.org/10.1007/978-3-642-03409-1_10
16. Fraenkel, A.S., Simpson, J.: How many squares can a string contain? J. Comb. Theory, Ser. A 82(1), 112–120 (1998), http://dx.doi.org/10.1006/jcta.1997.2843
17. Gusfield, D., Stoye, J.: Linear time algorithms for finding and representing all the tandem repeats in a string. J. Comput. Syst. Sci. 69(4), 525–546 (2004)
18. Halava, V., Harju, T., Kärki, T.: Square-free partial words. Inf. Process. Lett. 108(5), 290–292 (2008), http://dx.doi.org/10.1016/j.ipl.2008.06.001
19. Halava, V., Harju, T., Kärki, T.: On the number of squares in partial words. RAIRO - Theor. Inf. and Applic. 44(1), 125–138 (2010), http://dx.doi.org/10.1051/ita/2010008
20. Ilie, L.: A simple proof that a word of length n has at most $2n$ distinct squares. J. Comb. Theory, Ser. A 112(1), 163–164 (2005), http://dx.doi.org/10.1016/j.jcta.2005.01.006
21. Kociumaka, T.: Minimal suffix and rotation of a substring in optimal time. In: Grossi, R., Lewenstein, M. (eds.) 27th Annual Symposium on Combinatorial Pattern Matching, CPM 2016, June 27-29, 2016, Tel Aviv, Israel. LIPIcs, vol. 54, pp. 28:1–28:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2016), http://dx.doi.org/10.4230/LIPIcs.CPM.2016.28
22. Kolpakov, R.M., Kucherov, G.: Finding maximal repetitions in a word in linear time. In: 40th Annual Symposium on Foundations of Computer Science. pp. 596–604. FOCS'99, IEEE Computer Society (1999)

23. Main, M.G., Lorentz, R.J.: An O(n log n) algorithm for finding all repetitions in a string. J. Algorithms 5(3), 422–432 (1984), http://dx.doi.org/10.1016/0196-6774(84)90021-X
24. Manea, F., Mercas, R., Tiseanu, C.: An algorithmic toolbox for periodic partial words. Discrete Applied Mathematics 179, 174–192 (2014), http://dx.doi.org/10.1016/j.dam.2014.07.017
25. Manea, F., Tiseanu, C.: Hard counting problems for partial words. In: Dediu et al. [13], pp. 426–438, http://dx.doi.org/10.1007/978-3-642-13089-2_36

A Auxiliary Figures

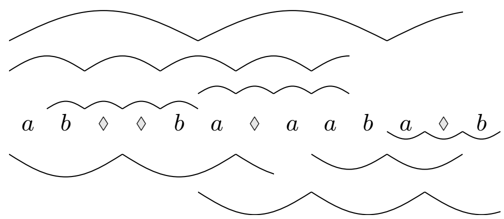

Fig. 2. A partial word together with all its Q-runs. This partial word contains 5 p-squares of length 3: $aaba\diamond b$, $\diamond aaba\diamond$, $ab\diamond\diamond ba$, $a\diamond aaba$, $b\diamond\diamond ba\diamond$. The following pair is equivalent: $ab\diamond\diamond ba \equiv a\diamond aaba$.