Theoretical Computer Science 411 (2010) 4235–4241

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Distinct squares in run-length encoded strings[☆]

J.J. Liu [*]

Department of Information Management, Shih Hsin University, Taipei, Taiwan, ROC

ARTICLE INFO

Article history:
Received 3 January 2010
Received in revised form 15 June 2010
Accepted 30 August 2010
Communicated by M. Crochemore

Keywords:
Squares
Repetition
rc-occurrence
Run-length encoding

ABSTRACT

Squares are strings of the form ww where w is any nonempty string. Two squares ww and $w'w'$ are of different types if and only if $w \neq w'$. Fraenkel and Simpson [Avieri S. Fraenkel, Jamie Simpson, How many squares can a string contain? Journal of Combinatorial Theory, Series A 82 (1998) 112–120] proved that the number of square types contained in a string of length n is bounded by $O(n)$. The set of all different square types contained in a string is called the *vocabulary* of the string. If a square can be obtained by a series of successive right-rotations from another square, then we say the latter *covers* the former. A square is called a *c-square* if no square with a smaller index can cover it and it is not a *trivial square*. The set containing all c-squares is called the *covering set*. Note that every string has a unique covering set. Furthermore, the vocabulary of the covering set are called *c-vocabulary*. In this paper, we prove that the cardinality of c-vocabulary in a string is less than $\frac{14}{3}N$, where N is the number of runs in this string.

1. Introduction

Let $x = x_1x_2 \cdots x_n$ be a string of length n and $x_i \in \Sigma$ where Σ is the alphabet with constant number of symbols. A *square* ww is an immediately repeated nonempty string w, such as "*aa*", "*abab*", or "*xyzxyz*". An occurrence of square ww in string x can be indicated by $(p, 2|w|)$ where p and $2|w|$ are the starting position and the length, respectively, of this square. Thus, $(p, 2|w|)$ is called the *occurrence form* of the indicated square. For convenience, we use w^1 and w^2 to represent the first and the second occurrences of w in a square respectively. A straightforward implementation would take $O(n^3)$ time to find all squares within a string by inspecting all possible substrings. Adapting KMP's failure function [9], the time complexity can be reduced to $O(n^2)$ (for more details, please see [1,2]). Note that the total number of occurrences of squares in a^n is $\theta(n^2)$, which implies that an $O(n^2)$-time algorithm is optimal if one needs to enumerate all occurrences. Main and Lorentz observed that many squares in a string are related [10–12]. As a consequence, it is possible to report a family of squares in constant time. Accordingly, they extended their previous algorithm for searching for one square [10] to an algorithm for searching for all squares [11] in $O(n \log n)$ time. Based on their results, Liu et al. proposed an algorithm for finding the positions of all squares within a run-length encoded string in $O(N \log N)$ time where N is the number of runs in the string [8].

If ww is a square, we say w is its square type. Two squares ww and $w'w'$ are of different types if and only if $w \neq w'$. The set of all different square types contained in a string is called the *vocabulary* of the string. For example, the vocabularies of the string "$abaa\underset{5}{b}aade\underset{10}{d}edba\underset{15}{a}baab\underset{20}{d}edea\underset{25}{a}baab\underset{30}{a}$" are {"*abaaba*", "*baabaa*", "*aa*", "*dede*", "*eded*", "*aabaab*"}. Note that the set of all square occurrences in the string is {(1, 6), (2, 6), (3, 2), (6, 2), (8, 4), (9, 4), (13, 6), (14, 2), (14, 6), (17, 2), (20, 4), (24, 2), (24, 6), (25, 6), (27, 2)}. Fraenkel and Simpson proved that the number of vocabularies contained in a string of length n is bounded by $O(n)$ [4]. Ilie gave a very short proof of this bound [6] and improve this bound to $2n - \Theta(\log n)$ [7].

[☆] This work was supported in part by the National Science Council of the Republic of China under the contract NSC 97-2218-E-128-001-.

[*] Corresponding address: No. 1, Lane 17, Sec. 1, Mu–Cha Road, Taipei, Taiwan, ROC. Tel.: +886 2 22368225x3366.
E-mail address: jjliu@cc.shu.edu.tw.

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.08.031

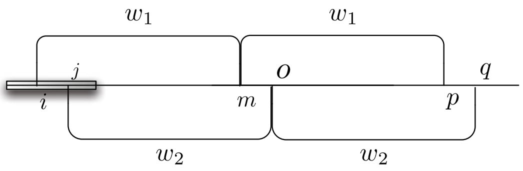


Fig. 1. Two squares of the same length starting within the same run.

In 2004, Gusfield and Stoye [5] presented an $O(n)$-time algorithm for finding all vocabularies and an $O(n+z)$-time algorithm for locating all square occurrences where z is the number of squares.

Let $a \in \Sigma$ be a single letter and $w \in \Sigma^*$ a non-empty string. If two squares $(p, 2|w_1|)$ and $(p+1, 2|w_2|)$ have the same length, i.e., $|w_1| = |w_2|$, then $(p+1, 2|w_2|)$ is called the *right-rotation* of $(p, 2|w_1|)$. Note that, assuming $(p, 2|w_1|)$ is aw, thus $(p+1, 2|w_2|)$ is wa. If $(i, 2|w|)$, $(i+1, 2|w|)$, $(i+2, 2|w|)$, …, $(j, 2|w|)$ are all squares, then we say that $(i, 2|w|)$ *covers* $(j, 2|w|)$. A square $(p, 2|w|)$ is called a *c-square* if no square $(q, 2|w|)$ with $q < p$ can cover it and $(p, 2|w|)$ is not a *trivial square*. This means that $(p, 2|w|) \neq a^{2|w|}$. The set containing all c-squares is called the *covering set*. Note that every string has a unique covering set while a covering set may contain different occurrences having the same square type. Furthermore, the different square types in the covering set are called *c-vocabulary*. The set containing all c-squares of different square types is called the *rc-set*. In addition, if more than one c-square has the same square type, then only the last occurrence is contained in the rc-set. Accordingly, an occurrence in the rc-set is called an *rc-occurrence*. In this paper, we prove that the number of rc-occurrences in a string of N runs is less than $\frac{14}{3}N$. Furthermore, this result also implies that the cardinality of the c-vocabulary is less than $\frac{14}{3}N$. For instance, in the previous example, the covering set is $\{(1, 6), (8, 4), (13, 6), (20, 4), (24, 6)\}$ and its corresponding c-vocabularies are {"*abaaba*", "*baabaa*", "*dede*", "*aabaab*"}. Notice that square occurrences $(2, 6)$, $(9, 4)$, $(14, 6)$, and $(25, 6)$ can be covered by $(1, 6)$, $(8, 4)$, $(13, 6)$, and $(24, 6)$, respectively, and $(3, 2)$, $(6, 2)$, $(14, 2)$, $(17, 2)$, $(24, 2)$, and $(27, 2)$ are all trivial squares. Furthermore, squares $(8, 4)$ and $(20, 4)$ are c-squares and they form the same square type, i.e., *dede*. We use the last occurrence form $(20, 4)$ to represent the c-vocabulary *dede*. Hence, the rc-set is $\{(1, 6), (13, 6), (20, 4), (24, 6)\}$. Note that, an rc-occurrence is not necessarily the last occurrence of square type in x. For instance, squares $(1, 6)$ and $(25, 6)$ are the same square type, i.e., *abaaba*. However, $(1, 6)$ is a c-square and $(25, 6)$ is not since $(24, 6)$ covers $(25, 6)$. Thus, c-vocabulary *abaaba* is represented by $(1, 6)$.

The remaining part of this paper is organized as follows. In Section 2, we introduce some properties of rc-occurrences. Using these properties, the cardinality of c-vocabulary can be obtained in Section 3. Conclusions and open problems are given in Section 4.

2. Some Properties of *rc*-occurrences

For a string $x = x_1x_2 \dots x_n$ where x_i's are letters, we use $X = r_1^{\ell_1} r_2^{\ell_2} \dots r_N^{\ell_N}$ to denote its corresponding run-length encoded string where r_i are letters with $r_i \neq r_{i+1}$ and $r_i^{\ell_i}$ means a *run* of ℓ_i copies of the letter r_i. We also use $x[i..j]$ to represent $x_ix_{i+1} \dots x_j$, where $1 \leqslant i \leqslant j \leqslant n$. Similarly, $X_{[i,j]}$ represents $r_i^{\ell_i} r_{i+1}^{\ell_{i+1}} \dots r_j^{\ell_j}$ where $i \leq j \leq N$. Let $|X_{[i,j]}| = \ell_i + \ell_{i+1} + \cdots + \ell_j$. It is clear that $|X_{[1,N]}| = n$. For any substring S, $r(S)$ denotes the number of runs in S. Thus, $r(x) = N$.

Lemma 1. *If there exist two squares* $(i, 2|w_1|)$ *and* $(j, 2|w_2|)$ *starting within the same run, say* $r_u^{\ell_u}$*, where* $i < j$ *and* $|w_1| = |w_2|$*, then* $(j, 2|w_2|)$ *must be covered by* $(i, 2|w_1|)$*.*

Proof. Let $m = i + |w_1|$, $o = j + |w_2|$, $p = i + 2|w_1|$, and $q = j + 2|w_2|$ (see Fig. 1 for an illustration). Since $(i, 2|w_1|)$ and $(j, 2|w_2|)$ are squares, $x_{[i,j-1]} = x_{[m,o-1]} = x_{[p,q-1]}$ and $x_{[j,m-1]} = x_{[o,p-1]}$. Assume that $r_u = a$ and let $k = |x_{[i,j-1]}| = j - i$. Therefore, square $(i, 2|w_1|)$ is $a^k x_{[j,p-1]}$ and $(j, 2|w_2|)$ is $x_{[j,p-1]}a^k$, for $k > 0$. It is clear that $(j, 2|w_2|)$ can be covered by $(i, 2|w_1|)$. □

Lemma 2. *If two distinct rc-occurrences* $(i, 2|w_1|)$ *and* $(j, 2|w_2|)$ *with* $i < j$ *and* $|w_1| > |w_2|$ *start within the same run, say* $r_u^{\ell_u}$*, then* $j + 2|w_2| - 1 > |X_{[1,u]}| + |w_1|$*.*

Proof. Let r be the ending position of w_2^2, i.e., $r = j + 2|w_2| - 1$. Then, we have two cases to consider.

Case 1. $r \leq i + |w_1| - 1$ (see Fig. 2(a) and (b)).

There exists another square say $(d, 2|w_2|)$ for $d > j$, which appears within w_1^2 since $w_1^1 = w_1^2$. Thus, $(d, 2|w_2|)$ is not a c-square and can be covered by $(d-1, 2|w_2|)$ since $(j, 2|w_2|)$ is an rc-occurrence. However, this implies that $(j-1, 2|w_2|)$ covers $(j, 2|w_2|)$. This contradicts the assumption that $(j, 2|w_2|)$ is an rc-occurrence.

Case 2. $i + |w_1| \leq r \leq |X_{[1,u]}| + |w_1|$ (see Fig. 2(c) and (d)).

This means that $x_{j-1} = x_{j+|w_2|-1} = x_r$. Thus, square $(j, 2|w_2|)$ can be covered by $(j-1, 2|w_2|)$ after a right-rotation. This establishes the lemma. □

Lemma 3. *If two distinct rc-occurrences* $(i, 2|w_1|)$ *and* $(j, 2|w_2|)$ *with* $i < j$ *and* $|w_1| > |w_2|$ *start from the same run, say* $r_u^{\ell_u}$*, then* $|X_{[1,u]}| + |w_2| + 1 < i + |w_1| < |X_{[1,u]}| + |w_1| < j + 2|w_2| - 1$*.*

Proof. Let p and r be the starting and ending positions, respectively, of w_2^2, i.e. $p = j + |w_2|$ and $r = j + 2|w_2| - 1$, and let $q = |X_{[1,u]}| + |w_2| + 1$. By Lemma 2, $r > |X_{[1,u]}| + |w_1|$. Thus, the inequality for the last two terms holds. It is obvious

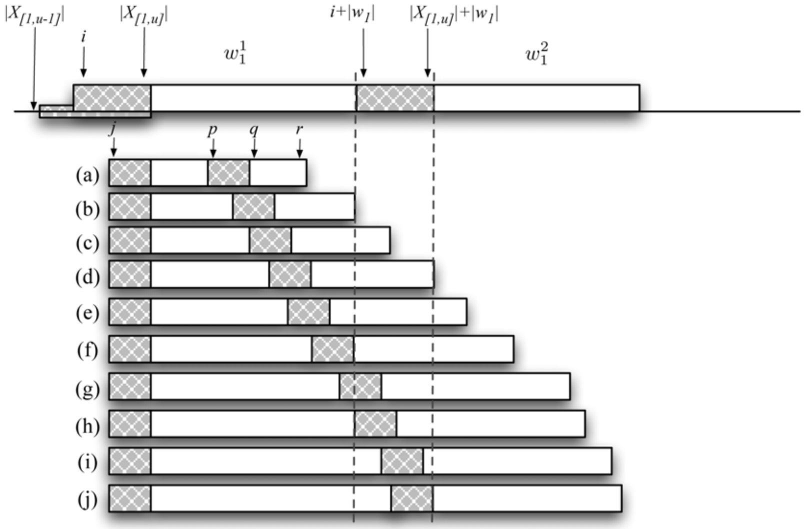


Fig. 2. Two squares starting at the same run with different lengths.

that the inequality holds for the second and the third terms. Therefore, we only need to consider the possible positions of the first term, namely q, in the inequality. If $q \geq |X_{[1,u]}| + |w_1| + 1$, we would have, by the definition of q that $|w_2| > |w_1|$ contradicting the assumptions of the lemma, so we have $q < |X_{[1,u]}| + |w_1| + 1$. Hence, $i + |w_1| \leq q \leq |X_{[1,u]}| + |w_1|$ (see Fig. 2(f)–(i)). In this case, x_q and x_{q-1} are located in the same run. This implies that $x_{|X_{[1,u]}|} = x_{|X_{[1,u]}|+1}$, a contradiction.

Therefore, if there exist two *rc*-occurrences $(i, 2|w_1|)$ and $(j, 2|w_2|)$ starting at the same run with $i < j$ and $|w_1| > |w_2|$, then $q < i + |w_1| < |X_{[1,u]}| + |w_1| < r$ (see Fig. 2(e)). □

We know that a run may have more than two *rc*-occurrences. In the following lemma, we describe a property between two rc-occurrences in the same run.

Lemma 4. *For any rc-occurrence $(i, 2|w_1|)$, there is at most one rc-occurrence $(j, 2|w_2|)$ starting within the same run, say $r_u^{\ell_u}$, with $i < j$ and $|w_1| > |w_2|$.*

Proof. Assume on the contrary that there exist another two *rc*-occurrences, say $(j, 2|w_2|)$ and $(k, 2|w_3|)$, such that $|w_3| < |w_1|$, $|w_2| < |w_1|$, and $|X_{[1,u-1]}| < i < j < k \leq |X_{[1,u]}|$. Note that the above inequalities and Lemma 3 imply that both w_2^1 and w_3^1 are substrings of w_1^1. We consider the following three cases.

Case 1. $|w_2| = |w_3|$.

By Lemma 1, $(k, 2|w_3|)$ must be covered by $(j, 2|w_2|)$. Thus, this case is impossible.

Case 2. $|w_2| > |w_3|$.

Since w_3^1 is a substring of w_2^1 which is a substring of w_1^1 by Lemma 3, w_3 is also a substring of both w_2^2 and w_1^2. We use w_{3B} and w_{3C} to denote the instances of w_3 appearing in w_2^2 and w_1^2, respectively, and w_3^2 is denoted by w_{3A} (see Fig. 3(a) for an illustration).

Let $\alpha = x[i..j-1]$, $\beta = x[j..k-1]$, and $\gamma = x[k..|X_{[1,u]}|]$, where $|\alpha|$, $|\beta|$, and $|\gamma|$ are all larger than 0. We must have $|w_1| < 2|w_3| + \alpha + \beta$ as otherwise square $(k, 2|w_3|)$ would appear later, contradicting the assumption that $(k, 2|w_3|)$ is the last occurrence. Put $w_2 = \beta w_3 \delta$ and assume that $r_u = a$. We consider the following two subcases.

Subcase 2.1. $|\delta| \leq |\gamma|$.

In this case, $\delta\beta$ is a prefix of w_3. The overlap between w_{3A} and w_{3B} gives that $\delta\beta$ repeatedly appears as a prefix of w_3 until the last one only contains a prefix of $\delta\beta$. Therefore, w_3 can be represented as $(\delta\beta)^q(\delta\beta)'$ where y' denotes some prefix of y. Since $r_u = a$, $w_3 = a^h$, for some integer h. However, $x_{|X_{[1,u]}|} \neq x_{|X_{[1,u]}|+1}$ and the statement "$w_3 = a^h$" is a contradiction.

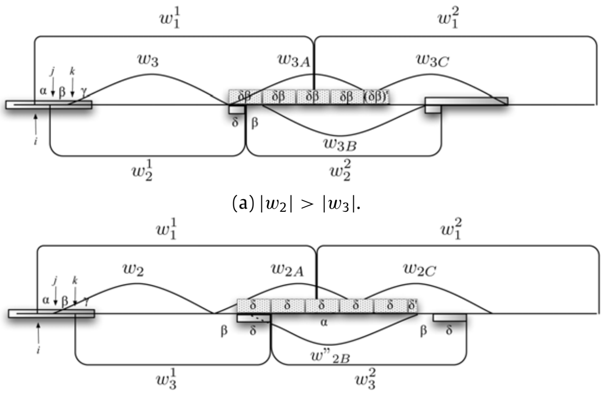
Subcase 2.2. $|\delta| > |\gamma|$.

Let $\delta = \gamma\tau$, where the first letter of τ is not equal to a. The overlap between w_{3A} and w_{3B} gives that $w_3 = (\gamma\tau\beta)^q(\gamma\tau\beta)'$. Since $w_3 = w_{3A}$, $w_{3A} = (\gamma\tau\beta)^q(\gamma\tau\beta)'$. The overlap between w_1^2 and w_{3A} starts at $\alpha\beta\gamma\tau$. It implies that $|\tau| > |\alpha|$, $\tau = \tau'\alpha$, and the last $|\alpha|$ letters of τ are $a^{|\alpha|}$. With square $(j, 2|w_2|) = \beta w_3 \gamma\tau'\alpha\beta w_3\gamma\tau'\alpha$ which can be found through $(i, 2|w_2|)$ after $|\alpha|$ right-rotations, a contradiction occurs.

Case 3. $|w_2| < |w_3|$.

Let $w_2 = \beta w_2''$, where y'' denotes some suffix of y. Since w_2'' is a substring of w_3^1 and w_2 is a substring of w_1^1, we use w_{2B}'' and w_{2C} to denote the instances of w_2'' and w_2 appearing in w_3^2 and w_1^2, respectively, and w_2^2 also expressed as w_{2A} (see Fig. 3(b) for an illustration).

Let $\alpha = x[i..j-1]$, $\beta = x[j..k-1]$, and $\gamma = x[k..|X_{[1,u]}|]$, where $|\alpha|$, $|\beta|$, and $|\gamma|$ are all larger than 0. We must have $|w_1| < 2|w_2| + \alpha$ as otherwise square $(j, 2|w_2|)$ would appear later. It contradicts that $(j, 2|w_2|)$ is the last occurrence. Put $w_3 = w_2''\beta\delta$. We consider the following two subcases.



(a) $|w_2| > |w_3|$.

(b) $|w_2| < |w_3|$.

Fig. 3. Three squares start at the same run.

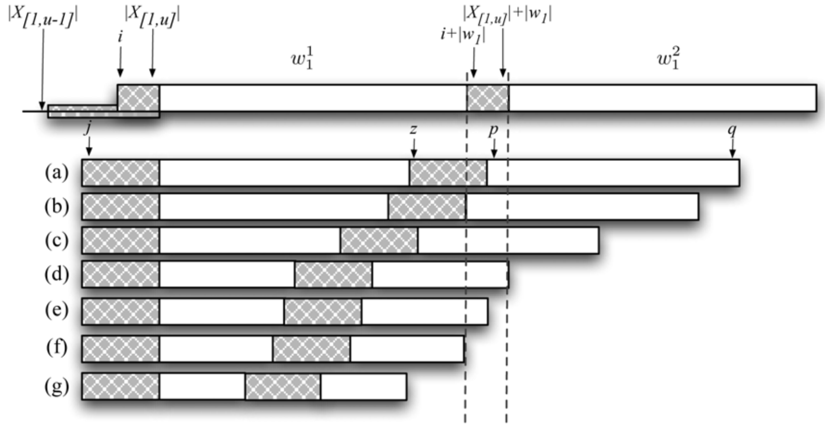


Fig. 4. Two squares start at the same run with different lengths.

Subcase 3.1. $|\delta| \le |\gamma|$.

In this case, $\beta\delta$ is a prefix of w_2. The overlap between w_{2A} and w''_{2B} gives that δ repeatedly appears in w_3 until the last one only contains a prefix of δ. Therefore, w_3 can be represented as $\beta(\delta)^q(\delta)'$. This implies that $w_3 = a^h$ for some integer h. Since $x_{|X_{[1,u]}|} \neq x_{|X_{[1,u]}|+1}$, the statement "$w_3 = a^h$" is a contradiction.

Subcase 3.2. $|\delta| > |\gamma|$.

Let $\delta = \gamma\tau$, where the first letter of τ is not equal to a. The overlap between w_{2A} and w''_{2B} gives that $w_2 = \beta(\gamma\tau)^q(\gamma\tau)'$. Since $w_2 = w_{2A}$, $w_{2A} = \beta(\gamma\tau)^q(\gamma\tau)'$. The overlap between w_1^2 and w_{2A} starts at $\alpha\beta\gamma\tau$. This implies that $|\tau| > |\alpha| + |\beta|$, $\tau = \tau'\alpha\beta$, and the last $|\alpha| + |\beta|$ letters of τ are $a^{|\alpha|+|\beta|}$. With square $(k, 2|w_3|) = w''_2\beta\gamma\tau'\alpha\beta w''_2\beta\gamma\tau'\alpha\beta$ which can be found through $(i, 2|w_3|)$ after $|\alpha| + |\beta|$ right-rotations, a contradiction occurs. Therefore, the lemma follows. □

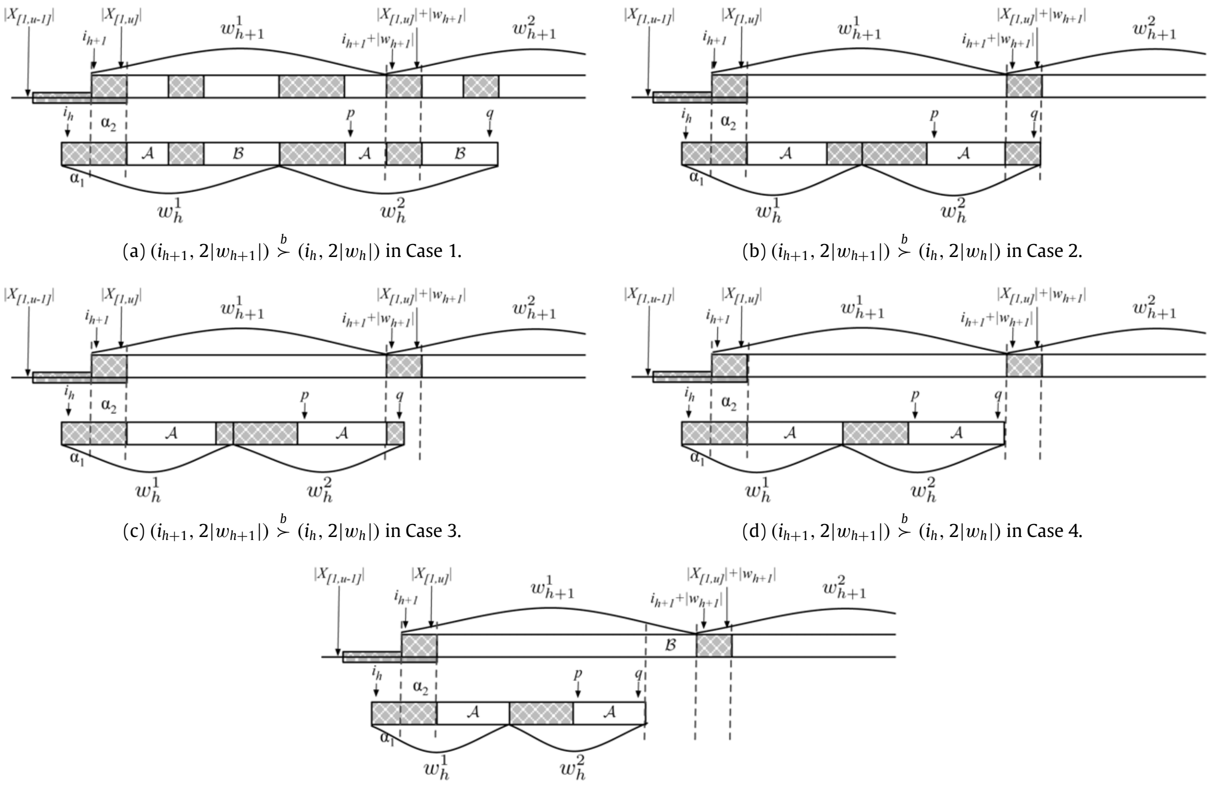
Now, we discuss the properties of two *rc*-occurrences $(i, 2|w_1|)$ and $(j, 2|w_2|)$ starting within the same run, for $j < i$ and $|w_1| > |w_2|$.

Lemma 5. *If there exist two rc-occurrences $(i, 2|w_1|)$ and $(j, 2|w_2|)$ starting within the same run, say $r_u^{\ell_u}$, where $j < i$ and $|w_1| > |w_2|$, then $|X_{[1,u]}| + |w_2| < i + |w_1| - 1$.*

Proof. Let $r_u = a$. Clearly, if $|X_{[1,u]}| + |w_2| \ge i + |w_1| - 1$ (see Fig. 4(a) and (b)), then the last $i - j$ letters of w_1 are a^{i-j}. Thus, square $(i, 2|w_1|)$ can be found through $(j, 2|w_1|)$ after $i - j$ right-rotations, a contradiction. □

For convenience, if there exist two *rc*-occurrences $(i, 2|w_1|)$ and $(j, 2|w_2|)$ starting within the same run, say $r_u^{\ell_u}$, where $j < i$ and $|w_1| > |w_2|$, then we say that $(i, 2|w_1|)$ *backwardly dominates* $(j, 2|w_2|)$, denoted as $(i, 2|w_1|) \overset{b}{\succ} (j, 2|w_2|)$. Now, we prove the following auxiliary lemma.

Lemma 6. *Let Γ be the substring $x[|X_{[1,u-1]}| + 1..i_m + 2|w_m| - 1]$ and $\gamma = r(\Gamma)$. If there exist m rc-occurrences in Γ such that $(i_m, 2|w_m|) \overset{b}{\succ} (i_{m-1}, 2|w_{m-1}|) \cdots \overset{b}{\succ} (i_1, 2|w_1|)$, where $i_1, i_2, \ldots,$ and i_m are all within the same run $r_u^{\ell_u}$ and $m \ge 1$, then $\gamma \ge 3m + 1$.*



(a) $(i_{h+1}, 2|w_{h+1}|) \overset{b}{\succ} (i_h, 2|w_h|)$ in Case 1.

(b) $(i_{h+1}, 2|w_{h+1}|) \overset{b}{\succ} (i_h, 2|w_h|)$ in Case 2.

(c) $(i_{h+1}, 2|w_{h+1}|) \overset{b}{\succ} (i_h, 2|w_h|)$ in Case 3.

(d) $(i_{h+1}, 2|w_{h+1}|) \overset{b}{\succ} (i_h, 2|w_h|)$ in Case 4.

(e) $(i_{h+1}, 2|w_{h+1}|) \overset{b}{\succ} (i_h, 2|w_h|)$ in Case 5.

Fig. 5. Two consecutive *rc*-occurrences $(i_{h+1}, 2|w_{h+1}|) \overset{b}{\succ} (i_h, 2|w_h|)$.

Proof. Let w_j^1 (respectively, w_j^2) denote the first (respectively, second) occurrence of w_j in $(i_j, 2|w_j|)$, for $j = 1, 2, \ldots, m$. Let γ_j stand for the smallest possible value of γ, i.e., $\gamma \geq \gamma_j$, while $m = j$. If this statement $\gamma_m \geq 3m + 1$ is true, then this lemma follows. We prove this statement by induction on m. If $m = 1$ then Γ contains a run of the form r_u^k, at least one run in $x[X_{[1,u-1]} + 1..i_1 + w_1]$, a copy of the suffix of $r_u^{\ell_u}$ in $x[i_1 + w_1 + 1..X_{[1,u]}]$ and at least one more run in the rest of Γ. Thus $\gamma_1 = 4 \geq 3m + 1$. For $m > 1$, assume that $\gamma_h \geq 3h + 1$. Thus, we consider two consecutive *rc*-occurrences $(i_{h+1}, 2|w_{h+1}|) \overset{b}{\succ} (i_h, 2|w_h|)$, where $1 \leq h < m$. Let $p = |X_{[1,u]}| + |w_h| + 1$, $q = i_h + 2|w_h| - 1$, and $r_u = a$. In addition, $S[1]$ denotes the first letter in string S. Thus, all the possible situations are formalized as follows :

Case 1: $p < i_{h+1} + |w_{h+1}| < |X_{[1,u]}| + |w_{h+1}| < q$ (see Fig. 5(a) for illustration)

Let $\alpha_1 = x[i_h..i_{h+1} - 1]$ and $\alpha_2 = x[i_{h+1}..|X_{[1,u]}|]$. Then, in this case, the *rc*-occurrence $(i_{h+1}, 2|w_{h+1}|)$ is of the form $(\alpha_2\mathcal{A}\alpha_2\mathcal{B}\alpha_1\alpha_2\mathcal{A})^2$ and $(i_h, 2|w_h|)$ is $(\alpha_1\alpha_2\mathcal{A}\alpha_2\mathcal{B})^2$, where $\mathcal{A} \in \Sigma^*$, $\mathcal{B} = (\mathcal{A}\alpha_2)^h\mathcal{C}$, $h \geq 0$, $\mathcal{C}$ is a prefix of $\mathcal{A}\alpha_2$, and $\mathcal{A}[1] \neq a$. Thus, $\gamma_{h+1} \geq \gamma_h + 3 \geq 3(h + 1) + 1$.

Case 2: $p < i_{h+1} + |w_{h+1}| < |X_{[1,u]}| + |w_{h+1}| = q$ (see Fig. 5(b) for illustration)

In this case, the *rc*-occurrence $(i_{h+1}, 2|w_{h+1}|)$ is of the form $(\alpha_2\mathcal{A}\alpha_2\alpha_1\alpha_2\mathcal{A})^2$ and $(i_h, 2|w_h|)$ is $(\alpha_1\alpha_2\mathcal{A}\alpha_2)^2$, where $\mathcal{A} \in \Sigma^*$ and $\mathcal{A}[1] \neq a$. Thus, $\gamma_{h+1} \geq \gamma_h + 3 \geq 3(h + 1) + 1$.

Case 3: $p < i_{h+1} + |w_{h+1}| < q < |X_{[1,u]}| + |w_{h+1}|$ (see Fig. 5(c) for illustration)

In this case, the *rc*-occurrence $(i_{h+1}, 2|w_{h+1}|)$ is of the form $(\alpha_2\mathcal{A}\alpha_3\alpha_1\alpha_2\mathcal{A})^2$ and $(i_h, 2|w_h|)$ is $(\alpha_1\alpha_2\mathcal{A}\alpha_3)^2$, where $\mathcal{A} \in \Sigma^*$, $\mathcal{A}[1] \neq a$, and $|\alpha_3| < |\alpha_2|$. Thus, $\gamma_{h+1} \geq \gamma_h + 3 \geq 3(h + 1) + 1$.

Case 4: $q = i_{h+1} + |w_{h+1}| - 1$ (see Fig. 5(d) for illustration)

In this case, the *rc*-occurrence $(i_{h+1}, 2|w_{h+1}|)$ is of the form $(\alpha_2\mathcal{A}\alpha_1\alpha_2\mathcal{A})^2$ and $(i_h, 2|w_h|)$ is $(\alpha_1\alpha_2\mathcal{A})^2$, where $\mathcal{A} \in \Sigma^*$ and $\mathcal{A}[1] \neq a$. Thus, $\gamma_{h+1} \geq \gamma_h + 3 \geq 3(h + 1) + 1$.

Case 5: $q < i_{h+1} + |w_{h+1}| - 1$ (see Fig. 5(e) for illustration)

In this case, the *rc*-occurrence $(i_{h+1}, 2|w_{h+1}|)$ is of the form $(\alpha_2\mathcal{A}\alpha_1\alpha_2\mathcal{A}\mathcal{B})^2$ and $(i_h, 2|w_h|)$ is $(\alpha_1\alpha_2\mathcal{A})^2$, where $\mathcal{A} \in \Sigma^*$ and $\mathcal{A}[1] \neq a$. Thus, $\gamma_{h+1} \geq \gamma_h + 3 \geq 3(h + 1) + 1$.

According to the above results, we can obtain that $\gamma_{h+1} \geq \gamma_h + 3 \geq 3(h + 1) + 1$ for $h = 1, 2, \ldots, m - 1$, and this lemma holds. □

3. The number of *rc*-occurrences

In this section, we prove that the number of *rc*-occurrences in a string of N runs is less than $\frac{14}{3}N$. Let S be a substring of $x[i..j]$. We use $IR(S)$ to represent the set of the *rc*-occurrences in S and use $|IR(S)|$ to denote the number of *rc*-occurrences in S. Given an *rc*-occurrence $t = (u, 2|w|)$, $IR(t)$ can be separated into "cross *rc*-occurrences" and "right *rc*-occurrences". The *right rc-occurrence* $(c, 2|d|)$ means the set of *rc*-occurrences in the second occurrence of w, i.e., $(c, 2|d|) \in IR(w^2)$. An *rc*-occurrence $(c, 2|d|)$ is called a *cross rc-occurrence* of t if $u < c < u + |w| < c + 2|d| < u + 2|w|$. Let $cross(t)$ denote the set of cross *rc*-occurrences $(c, 2|d|)$ of t. We also use $|cross(t)|$ to denote the number of cross *rc*-occurrences in t. Hence, $|IR(t)| = |cross(t)| + |IR(w^2)|$.

Theorem 1 ([8]). *Suppose that a string x of length n is partitioned into two nonempty strings u of length p and v of length q, where $p + q = n$ and $u_p \neq v_1$. The number of c-squares in x such that each one starts in u and ends in v is less than or equal to $r(x)$.*

Proof. We say that a square starting in u and ending in v is a *cross square* in x. In [8], cross squares in x can be grouped so that each group can be uniquely specified by two parameters (I, ℓ), i.e., and interval I of starting positions of squares together with the common length ℓ of every square. For instance, let u = "*daabc*" and v = "*bcbaabcbcba*". Squares $\{(2, 7), (3, 7), (4, 2), (5, 2)\}$ are cross squares in x and those can be grouped by $([15, 16], 7)$ and $([7, 8], 2)$. Since the number of groups with respect to u and v is bounded by $r(x)$ and each group can be obtained in constant time, all groups with respect to u and v can be calculated by $O(r(x))$. However, each group has just one c-square, i.e., the first square in this group. Thus, one sees easily that the number of c-squares in x such that each one starts in u and ends in v is less than or equal to $r(x)$. □

Let $t = w^1w^2$ be a nontrivial square, $a_1^{\ell_1}a_2^{\ell_2} \dots a_M^{\ell_M}$ be its corresponding run-length encoded string, and $|w_1| = |w_2| = m$. From Theorem 1, we know that if $w_m^1 \neq w_1^2$, then the number of c-squares in t such that each one starts in w^1 and ends in w^2 is less than or equal to $r(t)$. On the other hand, if $w_m^1 = w_1^2$, then it implies that w_m^1 and w_1^2 are within the same run, say r_u. Let ρ (respectively, ρ_1 and ρ_2) denote the set of c-squares in t such that each one starts in w^1 (respectively, $T_{[1,u-1]}$ and $T_{[1,u]}$) and ends in w^2 (respectively, $T_{[u,M]}$ and $T_{[u+1,M]}$). Since each c-square of t is either in ρ_1 or in ρ_2, $\rho \subseteq (\rho_1 \cup \rho_2)$. Since the number of c-squares in ρ_1 (respectively, ρ_2) is less than or equal to $r(t)$, the number of c-squares in ρ is less than or equal to $2r(t)$. By definition, *rc*-set is a subset of c-vocabulary. Therefore, we have proven the following lemma.

Lemma 7. *Let $t = w^1w^2$ be a nontrivial square. The number of c-squares that starts in w_1 and finishes in w_2 is at most $2r(t)$.*

Now, we are at a position to mention the number of *rc*-occurrences in a string.

Theorem 2. *For any string x of N runs, $|IR(x)| < \frac{14}{3}N$.*

Proof. Let λ_t denote a string of t runs, for $1 \leq t \leq N$. We prove this theorem by induction on t. If $t = 1$, string λ_t is trivial, i.e., $\lambda_t = a^k$ for some integer k where $a \in \Sigma$. Thus, $|IR(\lambda_t)| = 0$. For $t \geq 2$, assume that $|IR(\lambda_t)| \leq \frac{14}{3}t$ is true. Now, we consider the number of *rc*-occurrences in λ_{t+1}.

Let $\lambda_{t+1} = r_u \cdot \lambda_t$, where r_u is a newly added run (see Fig. 6). If there is no *rc*-occurrence starting within r_u, then $|IR(\lambda_{t+1})| = |IR(\lambda_t)| \leq \frac{14}{3}t$. On the other hand, if there are m *rc*-occurrences starting within r_u, then let us denote the longest one by $T_1 = (i_1, 2|w_1|)$. Let $r(T_1) = \gamma$, $\gamma \leq t + 1$. The overlap between w_1^1 and r_u is denoted by α, thus, $w_1^1 = \alpha w'$. In addition, λ_t can be partitioned into w' and $\lambda_t - w'$. Thus, $r(\lambda_t - w') \leq t - \frac{1}{2}\gamma$. By the property of Lemma 4, there is at most one *rc*-occurrence $(j, 2|w|)$ starting within the same run r_u, with $i_1 < j$ and $|w_1| > |w|$. Then, assume that there exist m *rc*-occurrences begin at $j \leq i_1$ (included T_1). So, by Lemma 6, $\gamma \geq 3m + 1$ and so $m \leq (\gamma - 1)/3$. Therefore,

$$\begin{aligned}
|IR(\lambda_{t+1})| &\leq m + 1 + |cross(T_1)| + |IR(\lambda_t - w')| \\
&\leq \frac{1}{3}(\gamma - 1) + 1 + |cross(T_1)| + |IR(\lambda_t - w')| \quad \text{(by Lemma 6)} \\
&\leq \frac{1}{3}(\gamma + 2) + 2\gamma + |IR(\lambda_t - w')| \quad \text{(by Lemma 7)} \\
&< \frac{1}{3}(\gamma + 2) + 2\gamma + \frac{14}{3}\left(t - \frac{1}{2}\gamma\right) \\
&= \frac{1}{6}(2\gamma + 4 + 12\gamma + 28t - 14\gamma) \\
&= \frac{1}{6}(28t + 4) \\
&< \frac{14}{3}(t + 1).
\end{aligned}$$

By induction, since $|IR(\lambda_{t+1})| < \frac{14}{3}(t + 1)$, this theorem follows. □

Theorem 3. *For any string x of N runs, the cardinality of c-vocabulary is less than $\frac{14}{3}N$.*

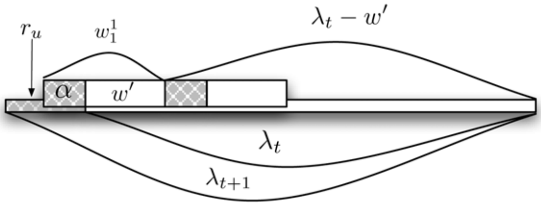


Fig. 6. An illustration for x_{t+1}.

Proof. By the definition of c-vocabulary, the cardinality of c-vocabulary is equal to the number of rc-occurrences. According to Theorem 2, the number of rc-occurrences in a string of N runs is less than $\frac{14}{3}N$. Thus, this completes the proof. □

4. Concluding remarks

Fraenkel and Simpson proved that the number of vocabularies contained in a string of length n is bounded by $O(n)$ [4]. Ilie gave a very short proof of this bound [6]. Furthermore, Gusfield and Stoye [5] presented an $O(n)$-time algorithm for finding out all vocabularies and an $O(n+z)$-time algorithm for locating all square occurrences where z is the number of squares.

A period of a string x is an integer p, $0 < p \leq |x|$, such that $x_i = x_{i+p}$ for all $i \in \{1, 2, \ldots, |x| - p\}$. Let $p(x)$ denote the size of the smaller period of x. We say that a string x is *periodic* if and only if $p(x) \leq \frac{|x|}{2}$. In [2], the authors defined that a *run* in a string x is an interval $[i..j]$ such that $x[i..j]$ is a periodic and this period is not extendable to the left or the right of $[i..j]$. Comparing with the definition of c-square, a c-square $(u, 2|p|)$ is in a run $[u..v]$, where $u + 2|p| \leq v$. The number of runs in a string x is equal to the number of c-squares in a string x except a run $[i..j]$ with $x[i..j] = a^k$ for some integer k. In [3], the upper bound on the maximum number of runs in a string is $1.6n$. Main [12] gave a linear-time algorithm for finding all leftmost occurrences of runs.

In this paper, we find that all squares can be represented by c-vocabulary, and prove that the cardinality of c-vocabulary is less than $\frac{14}{3}N$. Thus, it ensures the possibility for identifying all c-vocabulary in $O(N)$ time. Accordingly, all squares in a run-length encoded string (respectively, all leftmost occurrences of runs) can be reported in $O(N+z)$, where z is the number of squares (respectively, z is the number of runs). This is our future work.

References

[1] M. Crochemore, W. Rytter, Squares, cubes, and time-space efficient string searching, Algorithmica 13 (1995) 405–425.
[2] Maxime Crochemore, W. Rytter, Jewels of Stringology, World Scientific, Singapore, 2002.
[3] Maxime Crochemore, Lucian Ilie, Maximal repetitions in strings, Journal of Computer and System Sciences 74 (5) (2008) 796–807.
[4] Avieri S. Fraenkel, Jamie Simpson, How many squares can a string contain? Journal of Combinatorial Theory, Series A 82 (1998) 112–120.
[5] D. Gusfield, J. Stoye, Linear time algorithm for finding and representing all the tandem repeats in a string, Journal of Computer and System Sciences 69 (4) (2004) 525–546.
[6] Lucian Ilie, A simple proof that a word of length n has at most $2n$ distinct squares, Journal of Combinatorial Theory, Series A 112 (2005) 163–164.
[7] Lucian Ilie, A note on the number of squares in a word, Theoretical Computer Science 380 (2007) 373–376.
[8] J.J. Liu, G.S. Huang, Y.L. Wang, A fast algorithm for finding the positions of all squares in a run-length encoded string, Theoretical Computer Science 410 (2009) 3942–3948.
[9] D.E. Knuth, J.H. Morris, V.R. Pratt, Fast pattern-matching in strings, SIAM Journal on Computing 6 (2) (1977) 323–350.
[10] M.G. Main, R.J. Lorentz, An $O(n \log n)$ algorithm for recognizing repetitions, Technical Report CS-79-056, Washington State University, 1979.
[11] M.G. Main, R.J. Lorentz, An $O(n \log n)$ algorithm for finding all repetitions in a string, Journal of Algorithms 5 (3) (1984) 422–432.
[12] M.G. Main, Detecting Leftmost Maximal periodicities, Discrete Applied Mathematics 25 (1989) 145–153.