Most Pseudo-copy Languages Are Not Context-Free

Hyunjoon Cheon[1], Joonghyuk Hahn[1], Yo-Sub Han[1](✉), and Sang-Ki Ko[2]

[1] Yonsei University, Seoul, Republic of Korea
{hyunjooncheon,greghahn,emmous}@yonsei.ac.kr
[2] Kangwon National University, Gangwon-Do, Republic of Korea
sangkiko@kangwon.ac.kr

Abstract. It is well known that the copy language $L = \{ww \mid w \in \Sigma^*\}$ is not context-free despite its simplicity. We study pseudo-copy languages that are defined to be sets of catenations of two similar strings, and prove non-context-freeness of these languages. We consider the Hamming distance and the edit-distance for the error measure of the two similar strings in pseudo-copy languages. When the error has an upper bound or a fixed value, we show that the pseudo-copy languages are not context-free. Similarly, if the error has a lower bound of at least four, then such languages are not context-free, either. Finally, we prove that all these pseudo-copy languages are context-sensitive.

Keywords: Context-freeness · Pseudo-copy languages · Hamming distance · Edit-distance

1 Introduction

For many years, people investigated the problems related to the repetition of strings from various perspectives such as bioinformatics [3,7,12], stringology [2,5,16,20] and formal language theory [1,14]. For example, it was already proved in the early 80's that one can decide whether or not a given string contains a *square*—a string of the form ww with w nonempty—in $O(n \log n)$ time when n is the length of an input string [2,5,16].

The problem of finding squares (also called *tandem repeat* or *contiguous repeat*) from biological sequences has been an intriguing topic in bioinformatics. Landau et al. [12] studied the problem of finding *approximate tandem repeats* from a given string, which can be described as xy, where $|x| = |y|$ and $d(x, y) \leq k$ for a given k under the Hamming distance and the edit-distance metrics. They showed that all approximate tandem repeats can be found in $O(nk \log(n/k) + s)$ time, where n is the length of the given string and s is the number of repeats found. Later, Kolpakov and Kucherov [10] slightly improved the bound to $O(nk \log k + s)$ only in the case of the Hamming distance.

H. Cheon and J. Hahn—The first two authors contributed equally to this work.

C.-Y. Chen et al. (Eds.): COCOON 2021, LNCS 13025, pp. 189–200, 2021.
https://doi.org/10.1007/978-3-030-89543-3_16

We focus on the language-theoretic property related to the repetitions of strings. A string is *square-free* if none of its substrings is a square. It is easily seen that there are only finitely many square-free strings over one or two letters. Over a ternary alphabet, the set of square-free strings is infinite and, moreover, not context-free [15]. People also considered the complement of square-free languages—a language contains strings with at least one square as a substring. The language is also proved to be not context-free [6,18], and Ogden et al. [17] established a simpler proof using the interchange lemma.

The set of all squares, often called the *copy language* (denoted by COPY), is not context-free but can be recognized by realtime nondeterministic queue automata (NQAs) [11,21]. The class rtNQA of languages recognized by realtime NQAs is a proper subclass of context-sensitive languages (CS), and is incomparable to the class of context-free languages (CF). Therefore, it is immediate that the following relationship holds: $\text{COPY} \in \overline{\text{CF}} \cap \text{rtNQA} \subset \text{CS}$.

An interesting fact is that the complement of the copy language is context-free unlike COPY [9,19]. Since $\text{COPY} = \{xy \mid d_H(x, y) < 1\}$ and its complement of even-length[1] strings $\overline{\text{COPY}} = \{xy \mid d_H(x, y) > 0, \text{ where } |x| = |y|\}$ can be defined using the Hamming distance d_H, one can consider the following question.

Problem 1. Consider the following language L:

$$L = \{xy \mid x, y \in \{0, 1\}^*,\ |x| = |y|,\ d_H(x, y) < k\},$$

where $d_H(x, y)$ is the Hamming distance between x and y.

Q. Is L context-free?

We can think of the language L in Problem 1 as a set of catenations of two similar strings—we call such L a *pseudo-copy language*. In other words, the pseudo-copy language is a language with a bounded Hamming distance k between two catenated strings.

Since one may consider different bound conditions such as threshold, inequality or equality relations, and error measures, a natural question that arises next is, whether or not such languages are context-free. In particular, many people conjecture that a complement of a pseudo-copy language with $k = 2$ would not be context-free, yet there is no formal proof and the problem is still open[2]. Even before, Bordihn [4] asked the following question, which has not been answered yet.

Problem 2. Consider the following language L:

$$L = \{xy \mid x, y \in \{0, 1\}^*,\ |x| = |y|,\ |x| - d_H(x, y) \geq 2\}.$$

Q. Is L context-free?

We consider several variants of pseudo-copy languages and their complements depending on the bound conditions, and demonstrate that most pseudo-copy languages and their complements are not context-free.

[1] We only consider even-length strings for the Hamming distance between two halves.

[2] https://cs.stackexchange.com/q/11585.

2 Preliminaries

Let Σ denote a finite alphabet of symbols. Then a string w is a finite sequence of symbols from Σ and the length $|w|$ of w is the number of symbols in w. The character λ denotes an empty string.

For every string w and every natural number n, we define the n-th power of the string w, denoted by w^n, by $w^0 = \lambda$ and $w^k = w^{k-1}w$ for $k = 1, 2, \ldots, n$. For a string w of even length, we call two substrings α and β of the same length, where $w = \alpha\beta$, *halves* of w.

A *context-free grammar* (CFG) G is a tuple $G = (V, \Sigma, R, S)$, where V is a set of nonterminals, Σ is a set of terminals, $R \subseteq V \times (V \cup \Sigma)^*$ is a finite set of productions and $S \in V$ is the start symbol. Let $\alpha A \beta$ be a string over $V \cup \Sigma$, where $A \in V$ and $A \to \gamma \in R$. Then, we say that A can be rewritten as γ and the corresponding *derivation step* is denoted $\alpha A \beta \Rightarrow \alpha\gamma\beta$. A production $A \to t \in R$ is a *terminating production* if $t \in \Sigma^*$. The reflexive, transitive closure of $\Rightarrow$ is denoted by $\overset{*}{\Rightarrow}$ and the context-free language generated by G is $L(G) = \{w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w\}$ [19].

The *Hamming distance* $d_H(x, y)$ measures the error between two strings x and y of the same length by counting the number of different symbols on the same position of each [8]. In other words, $d_H(x, y) = \sum_i d(x_i, y_i)$, where $d(a, b) = 0$ if $a = b$ and one otherwise. For example, $d_H(abca, acab) = 3$ since there are three positions with different symbols. $d_S(x, y) = |x| - d_H(x, y)$, on the other hand, can be seen as the *similarity* between x and y, denoting the number of identical symbols at the same position of them.

An *alignment* of two strings x and y in Σ^* is a sequence of n pairs (x_1, y_1), (x_2, y_2), …, (x_n, y_n) where $x_i, y_i \in \Sigma \cup \{\lambda\}$, $x_1 x_2 \cdots x_n = x$ and $y_1 y_2 \cdots y_n = y$. The *edit-distance* $d_E(x, y)$ of two strings x and y is the minimum number of pairs with different symbols in alignments of x and y [13]. For instance, strings $abca$ and $acab$ have two alignments (a, a), (b, c), (c, a), (a, b) and (a, a), (b, λ), (c, c), (a, a), (λ, b). Although the first alignment is shorter, the number of different pairs is smaller for the second. Thus, the edit-distance of the two strings is two with the second alignment. Note that (λ, λ), (a, a), (b, c), (c, a), (a, b) is also a valid alignment for the strings.

We generalize the pseudo-copy language in Problem 1 by allowing different conditions between the two catenated strings. First is to consider different error measures. While Problem 1 defines a language with the Hamming distance d_H for the error measure. In Problem 2, we not only consider the conditions on the number for mismatches between two catenated strings but also matches by introducing the similarity measure d_S as follows. For two equal-length strings x, y, we define $d_S(x, y) = |x| - d_H(x, y)$. Another measure is the edit-distance d_E of x and y, which does not require the two strings to be the same length. The edit-distance allows more operations than the Hamming distance. From the perspective of error correction, a symbol is not only tripped but added or removed in transmission, which resembles the edit operations: substitution, insertion and deletion, respectively.

Second is to consider the relations for error values. Similar to $\overline{\mathrm{COPY}}$ where the Hamming distance is nonzero, we examine languages with different error bounds. Especially, these variants specify that the error (or similarity) of the two catenated strings should be bounded. For instance, one can think of a language with more than k different symbol positions in its halves ($d_H > k$). Note that the languages with a lower bound is a natural extension of $\overline{\mathrm{COPY}}$.

Problem 3. Given an integer $k \geq 0$ and an alphabet Σ, let $L = \{\alpha\beta \mid \alpha, \beta \in \Sigma^*,\ d(\alpha, \beta) \circ k\}$, where $d \in \{d_H, d_E, d_S\}$ and $\circ \in \{\leq, =, \geq\}$.

Q. Is such L context-free?

Let $L_{X \circ k}$ denote the language under $d = d_X$. For example, $L_{H=k}$ is the language under d_H and $\circ$ as $=$. For $d = d_E$, L is the language with its minimum edit-distance considered. The languages with the same error measure define a class with bounded errors.

3 Pseudo-copy Languages

The first problem is for $L_{H=k}$, whose halves have exactly k different symbols. Let us establish Lemma 4 for counting the Hamming distance on the specific form of strings for the problem.

Lemma 4. *For every string* $\alpha\beta = 0^a 1^b 0^c 1^d$*, where* $|\alpha| = |\beta|$*,* $d_H(\alpha, \beta) = \min(a + c, b + d, \max(|a - c|, |b - d|))$.

Proof (Sketch). If a 0-sequence occupies at least a half of $\alpha\beta$, then $d_H(\alpha, \beta)$ is the length $b + d$ of two 1-sequences. Otherwise, there is no sequence occupying a half. Without loss of generality, let us assume that a 0-sequence entirely aligns with the other 0-sequence. Then, $d_H(\alpha, \beta)$ is $|a - c|$, the number of 1's aligning with 0's. □

Based on the result of Lemma 4, we next show that $L_{H=k} = \{\alpha\beta \mid \alpha, \beta \in \Sigma^*,\ |\alpha| = |\beta|,\ d_H(\alpha, \beta) = k\}$ for every non-negative integer k is not context-free.

Theorem 5. *For all* $k \geq 0$*,* $L_{H=k}$ *is not context-free.*

Proof (Proof by contradiction). Suppose that $L_{H=k}$ is context-free. Then $L' = L_{H=k} \cap \{0^a 1^b 0^c 1^d \mid a, b, c, d \geq k\}$ must be context-free and satisfies the pumping lemma. For an arbitrary pumping constant p, let $z = 0^l 1^{l+k} 0^{l+k} 1^l \in L'$ where $l = \max(p!, k)$. Then z must have a decomposition of $uvwxy$ such that $|vx| > 0, |vwx| \leq p$ and $uv^n wx^n y \in L'$ for all $n \geq 0$. Note that vx can only be a part of at most two consecutive sequences, each sequence of which consists of only 0's or only 1's. By pumping v and x, we show that d_H exceeds k, which contradicts the pumping lemma.

1. When vx consists of only 0's or only 1's ($|vx|_0 = 0$ or $|vx|_1 = 0$)
Without loss of generality, assume that vx is in a sequence of 0's. We can pump v and x until the sequence of 0's that vx is in occupies over half of the string. Let $z' = \alpha'\beta' = uv^{|z|}wx^{|z|}y$ and $|\alpha'| = |\beta'|$. Then, since the sequence containing vx dominates z', $d_H(\alpha', \beta') = 2l + k > k$. The same procedure can be applied when vx is in the sequence of 1's.
2. When vx consists of both 0's and 1's ($|vx|_0 \neq 0$ and $|vx|_1 \neq 0$)
vx is in consecutive sequences in forms such as 0^a1^b or 1^b0^a. Apparently, when either v or x contains both 0 and 1, by pumping up v and x, we obtain strings that are not in L' which contradicts the pumping lemma. In the following, we assume that each of v and x contains only 0's or 1's. Without the loss of generality, let $|vx|_0 = a$ and $|vx|_1 = b$. Regarding which consecutive sequences vx is placed in, one of the following holds:
   - $d_H(\alpha', \beta') = \min(2l + k + \min(a, b)i, \max(|ai - k|, |bi + k|))$,
   - $d_H(\alpha', \beta') = \min(2l + k + \min(a, b)i, \max(|ai + k|, |bi + k|))$ or
   - $d_H(\alpha', \beta') = \min(2l + k + \min(a, b)i, \max(|ai + k|, |bi - k|))$

   where $\alpha'\beta' = uv^{i+1}wx^{i+1}y$. For example, when vx is in the first two sequences, applying Lemma 4 yields the first condition. Similarly, the other conditions can be computed from the remaining cases. All three cases show $d_H(\alpha', \beta') > k$ when $i = 2k + 2$, contradicting the pumping lemma. Note that $l \geq k$ and the first part cannot be the minimum.

By the above, L' is not context-free and, thus $L_{H=k}$ is not context-free. □

For different error bounds, we examine a language $L_{S=k} = \{\alpha\beta \mid \alpha, \beta \in \Sigma^*, |\alpha| = |\beta|, d_S(\alpha, \beta) = k\}$ that consists of strings whose halves have k identical symbols.

Theorem 6. *For all $k \geq 0$, $L_{S=k}$ is not context-free.*

Proof (Proof by contradiction). Suppose that $L_{S=k}$ is context-free and let $L' = L_{S=k} \cap L(0^*1^*0^*1^*0^*1^*)$. Then L' must satisfy the pumping lemma. For an arbitrary pumping constant p, choose $z = 0^P1^{P+k}0^P1^{P+k}0^P1^P$, where $P = 2(k+2)p$. Then z must have a decomposition of $uvwxy$ that satisfies the pumping lemma. Let $t = |vx|/2$, and α and β denote the first and the latter half of z. $z_i = uv^iwx^iy$ denotes the string after pumping v and x up $i - 1$ times, whose halves are α' and β', respectively. Note that $|vx|$ must be even—otherwise $z_0 \notin L'$. The following case-by-case proof shows that $z_{k+3} \notin L'$.

1. vx is in α,
When vx is in the first half α of z, pumping sends latter part of α to β. This results in having identical substring in the head of α and β. By pumping v and x up $k + 2$ times, the last $0^{t(k+2)}$ portion of α is pushed to the front of the latter half, thus $z_{k+3} = \alpha'0^{t(k+2)}1^{P+k}0^P1^P$, as illustrated in Fig. 1. Then $d_S(\alpha', \beta') \geq t(k + 2) > k$.

2. vx is in β,
   Similar to the case when vx is in α, we pump vx to obtain identical substring in the tail of α and β. By pumping up v and x by $k+2$ times, the first $1^{t(k+2)}$ portion of β is pushed to the first half, thus $z_{k+3} = 0^P 1^{P+k} 0^P 1^{t(k+2)} \beta'$. Then $d_S(\alpha', \beta') \geq t(k+2) > k$.
3. vx is in both α and β, $vx = 0^a 1^b$.
   Contrary to the above, pumping vx does not always result in having identical substring in the head or tail of α and β. We, therefore, examine the inner part of α and β, specifically, 1^{P+k}.
   (a) When $a \leq b$, $\alpha' = \alpha 0^{a(k+2)} 1^{(b-a)(k+2)/2}$ and $\beta' = 1^{(a+b)(k+2)/2} \beta$. Since β is pushed by $\frac{a+b}{2}(k+2)$ while α is not, the overlap in 1^{P+k} strictly increases. Thus $d_S(\alpha', \beta') \geq \frac{a+b}{2}(k+2) + k > k$.
   (b) When $a > b$, $\alpha' = \alpha 0^{(a+b)(k+2)/2}$ and $\beta' = 0^{(a-b)(k+2)/2} 1^{b(k+2)} \beta$. Thus $d_S(\alpha', \beta') \geq \frac{a+b}{2}(k+2) + k > k$.

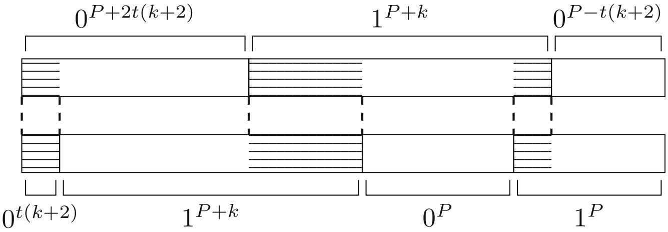

Fig. 1. Illustration of z_{k+3} after pumping the first 0-sequence. The slanted lines denote the alignment pairs with the same symbols. Note that the second and the third overlaps already have k symbols aligned.

Since every case contradicts the pumping lemma, L' is not context-free, which leads to the fact that $L_{S=k}$ is not context-free. □

For the edit-distance case, we show that the Hamming distance and the edit-distance between the two catenated strings of a pseudo-copy language are the same. For a string $w = \alpha\beta$, we denote $\widehat{d_H}(w) = d_H(\alpha, \beta)$ and $\widehat{d_E}(w) = \min_{w=\alpha'\beta'} d_E(\alpha', \beta')$—the smallest edit-distance among all possible α', β' for w.

Lemma 7. *Let $w \in L(0^*1^*0^*1^*)$ be a string of even length. Then, $\widehat{d_E}(w) = \widehat{d_H}(w)$.*

Proof (Proof by induction). When $|w| = 0$, $\widehat{d_E}(w) = \widehat{d_H}(w) = 0$. Assume the claim holds for $|w| \leq n$. For $|w| = n+2$, suppose that the claim does not hold. Then, since $\widehat{d_E}(w) < \widehat{d_H}(w)$ is the case, there must be an optimal alignment with two symbols u, v that matches to λ. Let w' be the string without u and v, then $\widehat{d_E}(w) = \widehat{d_E}(w') + 2 = \widehat{d_H}(w') + 2 < \widehat{d_H}(w)$. This cannot hold by case analysis on Lemma 4, contradicting the claim. □

Theorem 8. *For all $k \geq 0$, $L_{E=k}$ is not context-free.*

Proof (Proof by contradiction). Let $L' = L_{E=k} \cap \{0^a1^b0^c1^d \mid a,b,c,d \geq k \text{ and } (a+b+c+d) \bmod 2 = 0\}$ and suppose $L_{E=k}$ is context-free. Then, L' must be context-free and satisfies the pumping lemma. For an arbitrary pumping constant p, let $z = 0^l1^{l+k}0^{l+k}1^l \in L'$, where $l = \max(p!, k)$. Then z must have a factorization of $uvwxy$ such that $|vx| > 0$, $|vwx| \leq p$, and $uv^nwx^ny \in L'$ for all $n \geq 0$. By pumping v and x, we show that d_E exceeds k, which contradicts the pumping lemma. Referring to Lemma 7, $\widehat{d_E}(w) = \widehat{d_H}(w)$ for $w \in L'$. Instead of handling d_E, we can show that d_H exceeds k and this is already proven in Theorem 5. Therefore L' is not context-free. By the above, $L_{E=k}$ is not context-free. □

One can define a hierarchy of pseudo-copy languages over exact error with these results. Theorem 5, 6 and 8 show that the class of languages with exact Hamming distance (exact similarities, edit-distance, resp.) is different from that of context-free languages.

From the proofs for the exact cases in Theorems 5, 6 and 8, one can observe that the error value of the chosen string strictly increases after pumping. These strings also apply to showing that the pseudo-copy languages are not context-free.

Corollary 9. *For all $k \geq 0$, $L_{H\leq k}$, $L_{S\leq k}$ and $L_{\leq k}$ are not context-free.*

Proof. In Theorem 5, we prove that $L_{H=k}$ is not context-free by showing the strings in $L_{H=k}$ have a larger error value when pumped, following Theorem 5. We can apply the exactly same procedure here. Instead of applying the pumping lemma directly to $L_{H\leq k}$, define $L' = L_{H=k} \cap \{0^a1^b0^c1^d \mid a,b,c,d \geq k\}$. We know that L' is not context-free as the pumped string has an error value larger than k. This is, in other words, the string which has an upper-bounded error value of k can be pumped until the error value exceeds k. Therefore, the same string for $L_{H=k}$ contradicts the pumping lemma for $L_{H\leq k}$. Respectively on $L_{S\leq k}$ and $L_{E\leq k}$, we can use the proof procedure in each case similarly. □

4 Complements of Pseudo-copy Languages

The complements of pseudo-copy languages under the error measure $d_X \in \{d_H, d_S, d_E\}$ are defined as follows:

$$\overline{L_{X\leq k}} = L_{X\geq k+1}.$$

Therefore, only even-length strings exist in $\overline{L_{d_H\leq k}}$ and $\overline{L_{d_S\leq k}}$ for the Hamming distance and similarity, respectively. On the other hand, the complements of pseudo-copy languages under the edit-distance can have both odd-length and even-length strings.

Theorem 10. *For all $k \geq 4$, $L_{H\geq k}$ is not context-free.*

Proof (Proof by contradiction). The intuition is choosing a string of which a symbol, say 0, occupies the largest portion. Then we make an alignment of the sparse symbols, say 1's, to reduce the Hamming distance between its halves at least by one. For $n \geq 2$, suppose that $L_{H \geq 2n}$ is context-free and let $L' = L_{H \geq 2n} \cap \{w \mid |w|_1 = 2n+1\}$. Then, by the pumping lemma, there must be a pumping constant p for L'.

Choose $z \in L'$ so that the position indices of 1's, $i_j \geq 1$ $(1 \leq j \leq 2n+1)$ are

$$i_j = \begin{cases} 2jP, & j \leq n, \\ (2j+1)P - 1, & j > n, \end{cases}$$

where $P = p!$. In other words, we place 1's in z so that when we divide the string into halves, 1's from the first half alternate with 1's in the second half by $l = 2(n+1)P - 1$.

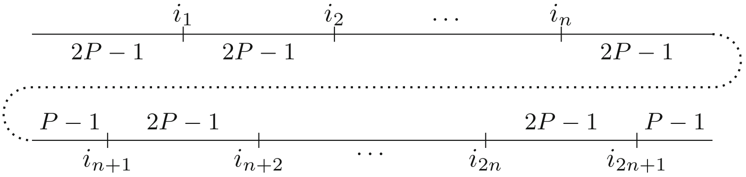

Fig. 2. An illustration of the chosen z

Let i'_j denote the indices of 1's after pumping v and x. It contradicts the pumping lemma if there exist s and t such that $i'_s \leq l + T < i'_t$ and $i'_t - i'_s = l + T$— two 1's in each half are aligned in the Hamming distance computation— where $2T$ is the length of the entire pumped string, and therefore, v and x duplicate $2T/|vx|$ times. The Hamming distance d_H is at most $2n - 1$ in this case since two 1's do not contribute on the Hamming distance computation.

1. $|vx|$ is odd or $|vx|_1 = 1$, i.e., the pumping part contains 1.
   Since $uv^0xw^0y \notin L'$, it contradicts the pumping lemma.
2. $|vwx|_1 = 0$, i.e., the pumping occurs in a single 0-sequence. See Fig. 3 for an example.
   If vwx is in the h-th 0-sequence, the indices i'_j of 1's after pumping up v and x $2T/|vx|$ times is

   $$i'_j = \begin{cases} i_j, & j < h, \\ i_j + 2T, & j \geq h, \end{cases}$$

   assuming $2T/|vx|$ is an integer.
   (a) If $h \leq n$, let $s = h$ and $t = h + n + 1$. $i'_t - i'_s = (2n+3)P - 1$. Figure 3(a) depicts how two 1's align. Note that the right-hand side of h-th 1 in the first half shortens by T while that of $(h+n+1)$-th in the second half does not. These two 1's eventually meet after the pumping, when $l + T = i'_t - i'_s$, i.e., when $T = P$.

(b) If $n < h < 2n+1$, let $s = h - n$ and $t = h$. $i'_t - i'_s = (2n+1)P + 2T - 1$. Figure 3(b) depicts how two 1's align.
(c) If $h \in \{2n+1, 2n+2\}$, let $s = 1$ and $t = n+2$. $i'_t - i'_s = (2n+3)P - 1$.
Since, for all of three cases, $i'_t - i'_s = l + T$ holds if $T = P$, we pump up v and x $2P/|vx|$ times to contradict the pumping lemma for any positive integer p. Note that $P/|vx| = p!/|vx|$ is an integer.

3. $|vwx|_1 = 1, |vx|_1 = 0$, i.e., the pumping occurs in two 0-sequences.
For h such that the h-th 1 is in w, the indices i'_j after pumping up v and x $2T/|vx|$ is

$$i'_j = \begin{cases} i_j, & j < h, \\ i_j + a, & j = h, \\ i_j + 2T, & j > h, \end{cases}$$

where $a = |v| \cdot 2T/|vx|$.
(a) $h = 1$: Let $s = 2$ and $t = n+3$. $i'_t - i'_s = (2n+3)P - 1$.
(b) $2 \le h \le n$: Let $s = 1$ and $t = n+1$. $i'_t - i'_s = (2n+1)P + 2T - 1$.
(c) $n+1 \le h \le 2n-1$: Let $s = n$ and $t = 2n$. $i'_t - i'_s = (2n+1)P + 2T - 1$.
(d) $h \in \{2n, 2n+1\}$: Let $s = 1$ and $t = n+2$. $i'_t - i'_s = (2n+3)P - 1$.
For all cases, $i'_t - i'_s = l + T$ holds if $P = T$ and it contradicts the pumping lemma for CFLs.

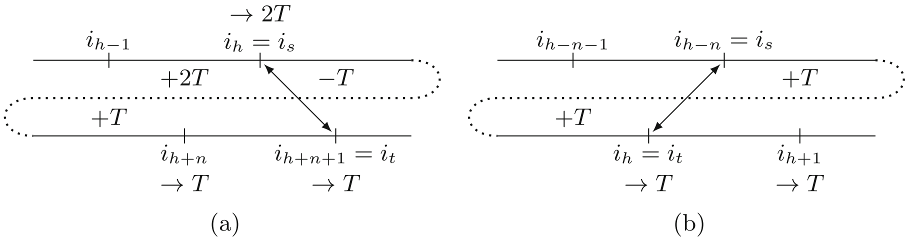

Fig. 3. Pumping a 0-block in (a): the first half and (b): the latter half. On each half, $\pm X$ denotes that the length of the sequence increases by X and $\to X$ denotes that the specific point is pushed by X.

Because every case contradicts the pumping lemma, L' is not context-free and neither is $L_{H \ge 2n}$. The case for $L_{H \ge 2n+1}$ is similar to the proof above. □

We have investigated languages with lower-bounded Hamming error values and in most cases, they are not context-free. However, it is still unknown whether or not $L_{H \ge 2}$ and $L_{H \ge 3}$ are context-free or non-context-free.

The case of $L_{S \ge k}$ starts with an obvious observation that $L_{S \ge 1}$ is context-free. $L_{S \ge 1}$ can be generated by the following CFG $G = (V, \Sigma, R, S)$:

$$\begin{aligned} S &\to AA \mid BB, \\ A &\to 0A0 \mid 0A1 \mid 1A0 \mid 1A1 \mid 0, \\ B &\to 0B0 \mid 0B1 \mid 1B0 \mid 1B1 \mid 1. \end{aligned}$$

Regarding $L_{S\geq k}$ with $k \geq 2$, we establish that the language over ternary alphabet is not context-free. Refer to the appendix for the full proof.

Theorem 11. *For all $k \geq 2$, $L_{S\geq k}$ over a ternary alphabet is not context-free.*

For binary case, some languages of the same type are not context-free.

Theorem 12. *For all $k \geq 5$, $L_{S\geq k}$ over a binary alphabet is not context-free.*

Proof (Sketch). This proof idea is similar to that of Theorem 10. Assume that k is even and $L_{S\geq k}$ is context-free, then the following L' is also context-free and should satisfy the pumping lemma.

$$L' = L_{S\geq k} \cap L([(01)^*00]^{k/2}[(01)^*11]^{k/2}(01)^*). \qquad (*)$$

For the illustration purpose, let $k = 6$. For a pumping constant p, let $P = (\max\{p, k\})!$ and choose

$$z = uvwxy = (01)^P 00(01)^{2P}00(01)^{2P}00(01)^{4P+1}11(01)^{2P}11(01)^{2P}11(01)^P.$$

The 00's and 11's alternate like the 1's in Theorem 10. We can observe that v and x must be in $L((01)^* + (10)^*)$, otherwise, $uwy \notin L'$. It is also worth noting that Fig. 4 is the target alignment of 00 and 11 in each half, which reduces the similarity by two. Our goal is to show that similarity reduces for all possible cases, contradicting the pumping lemma.

$$\begin{gathered} \ldots 0101\underline{00}01010 \ldots \\ \ldots 10101\underline{11}0101 \ldots \end{gathered}$$

Fig. 4. The target alignment for z. The symbols not from $(01)^*$ are underlined. The 11 shifts to the right by one symbol.

We make $z' = uv^iwx^iy$ to show such alignment by pumping up v and x sufficiently large. For example, when both v and x are in the first (01)-block, by shifting all 00's and 11's, z' has similarity of $0 < k$ with $(k/2) = 3$ target alignments. The following is z' after pumping up v and x sufficiently so that $(i-1) \cdot |vx| = 4P$, where i is the number of duplications.

$$\begin{aligned} z' =&(01)^{P\underline{+2P}}\underline{00}(01)^{2P}\underline{00}(01)^{2P}\underline{00}(01)^{2P-P}0 \\ &1(01)^{2P+P}\underline{11}(01)^{2P}\underline{11}(01)^{2P}\underline{11}(01)^P \end{aligned}$$

One can make at least one target alignment for every factorization of $uvwxy$ and it reduces the similarity at least by two. Thus, z' has similarity of at most $k-2$, which contradicts the pumping lemma—$L_{S\geq k}$ is not context-free. This argument also holds for odd k, but with $k+1$ instead of k for choosing a regular language to intersect with $L_{S\geq k}$ in (∗). □

We then provide Lemma 13 as a simple conversion scheme from a language with the edit-distance to a corresponding language with the Hamming distance.

Lemma 13. *Let $\Gamma = \{0, 1, \#\}$ be an alphabet and $h : \Gamma \to \Sigma$ be a homomorphism such that $h(0) = 0$, $h(1) = 1$ and $h(\#) = \lambda$. Then, $h^{-1}(L_{E \geq k}) \cap L((\Sigma^2)^*)$ is the language with $d_H \geq k$ over Γ.*

Proof. For $\alpha\beta \in L_{E \geq k}$, every alignment of α and β has at least k different pairs. Then, h^{-1} replaces λ in such alignment pairs in $L_{E \geq k}$ or inserts $(\#, \#)$ pairs. Thus, the strings with even length represent alignments of the strings in $L_{E \geq k}$, with at least k differences.

On the other hand, let $L = L_{H \geq k}$ over Γ. Then, on its alignment of two halves, one can derive an alignment for strings with at least k different pairs by replacing $\#$ with λ. □

Since context-free languages are closed under inverse homomorphism [9], if a language with the edit-distance is context-free, then the resulting language, which is one with the Hamming distance, must be context-free. We now show that such language is not context-free due to Theorem 10 and Lemma 13.

Theorem 14. *For $k \geq 4$, $L_{E \geq k}$ is not context-free.*

Proof (Proof by contradiction). Suppose that $L_{E \geq k}$ is context-free. Consider the alphabet Γ and the homomorphism h in Lemma 13. Since context-free languages are closed under these operations, $h^{-1}(L_{E \geq k}) \cap L((\Sigma^2)^*)$ must be context-free. However, this language is $L_{H \geq k}$ over Γ, which is proven to be non-context-free in Theorem 10 for $k \geq 4$. □

Finally, we can easily show that the pseudo-copy languages are strictly included in the class of context-sensitive languages by constructing realtime NQAs. Refer to the appendix for full proofs.

5 Conclusions

We have examined the problems of determining non-context-freeness of pseudo-copy languages and their complements defined under error measures such as the Hamming distance and the edit-distance. Unlike $\overline{\text{COPY}}$, the languages are proved to be non-context-free. Especially, our results show that most pseudo-copy languages as well as their complements are not context-free. It is interesting as the complements are not significantly different from $\overline{\text{COPY}}$ which is context-free.

There are, however, remaining problems that need further investigation to determine their context-freeness. Even though our results show that the answer for Problem 1 is not context-free, it still remains open for the complements of extended pseudo-copy languages. $L_{H \geq k}$, $L_{E \geq k}$ and $L_{S \geq k}$ regarding errors of small lower-bounds are to be examined in further study. We hope that our findings are helpful for answering these questions.

Acknowledgments. We wish to thank the referees for valuable suggestions that improve the presentation of the paper.

This research was supported by the NRF grant funded by MIST (NRF-2020R1A4A3079947, NRF-2018R1D1A1A09084107).

References

1. Anderson, T., Rampersad, N., Santean, N., Shallit, J.: Finite automata, palindromes, powers, and patterns. In: Proceedings of the 2nd International Conference on Language and Automata Theory and Applications, pp. 52–63 (2008)
2. Apostolico, A., Preparata, F.: Optimal off-line detection of repetitions in a string. Theor. Comput. Sci. **22**(3), 297–315 (1983)
3. Bordihn, H., Mitrana, V., Păun, A., Păun, M.: Hairpin completions and reductions: semilinearity properties. Nat. Comput. **20**(2), 193–203 (2020). https://doi.org/10.1007/s11047-020-09797-0
4. Bordihn, H., Shallit, J.: Personal communication
5. Crochemore, M.: Recherche linéaire d'un carré dans un mot. Comptes rendus de l'Académie des sciences. Série I, Mathématique **296**(18), 781–784 (1983)
6. Ehrenfeucht, A., Rozenberg, G.: On the separating power of EOL systems. RAIRO Theor. Inform. Appl. **17**(1), 13–22 (1983)
7. Gusfield, D.: Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, USA (1997)
8. Hamming, R.W.: Error detecting and error correcting codes. Bell Syst. Tech. J. **29**(2), 147–160 (1950)
9. Hopcroft, J.E., Motwani, R., Ullman, J.D.: Introduction to Automata Theory, Languages, and Computation, 3rd edn. Pearson Education Inc, Boston, MA, USA (2006)
10. Kolpakov, R., Kucherov, G.: Finding approximate repetitions under Hamming distance. Theor. Comput. Sci. **303**(1), 135–156 (2003)
11. Kutrib, M., Malcher, A., Wendlandt, M.: Queue automata: foundations and developments. In: Adamatzky, A. (ed.) Reversibility and Universality. ECC, vol. 30, pp. 385–431. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-73216-9_19
12. Landau, G.M., Schmidt, J.P., Sokol, D.: An algorithm for approximate tandem repeats. J. Comput. Biol. **8**(1), 1–18 (2001)
13. Levenshtein, V.I.: Binary codes capable of correcting deletions, insertions, and reversals. Sov. Phys. Doklady **10**(8), 707–710 (1966)
14. Lischke, G.: Squares of regular languages. Math. Log. Q. **51**(3), 299–304 (2005)
15. Lothaire, M.: Combinatorics on Words, 2nd ed. Cambridge University Press, Cambridge (1997)
16. 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)
17. Ogden, W.F., Ross, R.J., Winklmann, K.: An "interchange lemma" for context-free languages. SIAM J. Comput. **14**(2), 410–415 (1985)
18. Ross, R.J., Winklmann, K.: Repetitive strings are not context-free. RAIRO Theor. Inform. Appl. **16**(3), 191–199 (1982)
19. Sipser, M.: Introduction to the Theory of Computation, 3rd edn. Cengage Learning, Boston, MA, USA (2013)
20. Stoye, J., Gusfield, D.: Simple and flexible detection of contiguous repeats using a suffix tree. Theor. Comput. Sci. **270**(1), 843–856 (2002)
21. Vollmar, R.: Über einen automaten mit pufferspeicherung. Computing **5**(1), 57–70 (1970)