K-Comma Codes and Their Generalizations [*]

Bo Cui
Department of Computer Science, University of Western Ontario,
London, Ontario, Canada, N6A 5B7
bcui2@csd.uwo.ca

Lila Kari
Department of Computer Science, University of Western Ontario,
London, Ontario, Canada, N6A 5B7
lila@csd.uwo.ca

Shinnosuke Seki
Department of Computer Science, University of Western Ontario,
London, Ontario, Canada, N6A 5B7
sseki@csd.uwo.ca

Abstract. In this paper, we introduce the notion of k-comma codes - a proper generalization of the notion of comma-free codes. For a given positive integer k, a k-comma code is a set L over an alphabet Σ with the property that $L\Sigma^k L \cap \Sigma^+ L \Sigma^+ = \emptyset$. Informally, in a k-comma code, no codeword can be a subword of the catenation of two other codewords separated by a “comma” of length k. A k-comma code is indeed a code, that is, any sequence of codewords is uniquely decipherable. We extend this notion to that of k-spacer codes, with commas of length less than or equal to a given k. We obtain several basic properties of k-comma codes and their generalizations, k-comma intercodes, and some relationships between the families of k-comma intercodes and other classical families of codes, such as infix codes and bifix codes. Moreover, we introduce the notion of n-k-comma intercodes, and obtain, for each $k \geq 0$, several hierarchical relationships among the families of n-k-comma intercodes, as well as a characterization of the family of 1-k-comma intercodes.

Address for correspondence: Bo Cui, Department of Computer Science, University of Western Ontario, London, Ontario, Canada, N6A 5B7

[*]This research was supported by The Natural Sciences and Engineering Council of Canada Discovery Grant R2824A01, and Canada Research Chair Award to L.K.

1. Introduction

The notion of codes is crucial in many areas such as information communication, data compression, and cryptography. In such systems, it is required that, if a message is encoded by using words from a code, then any arbitrary catenation of words should be uniquely decodable into codewords. Various codes with specific algebraic properties, such as prefix codes, infix codes, and comma-free codes [1, 4, 15, 18], have been motivated and defined for various purposes. For instance, the definition of comma-free codes [2, 5] followed the 1953 discovery of the double-helical structure of DNA, [17], as a proposed mathematical solution to a problem which arose in connection with protein synthesis. The problem was the following. There are 20 known types of aminoacids. The most plausible hypothesis at the time, that each aminoacid is encoded by one three-letter DNA sequence, i.e., a 3-letter sequence over the four-letter alphabet $\{A, C, G, T\}$ raised the following question: From the possible $4^3 = 64$ three-letter words over the DNA alphabet, which ones code for aminoacids and why? The hypothesis was advanced, for example, [2, 5, 17] that the triplets coding for aminoacids form a *comma-free code*, i.e., a set with the property that any sequence of codewords is uniquely decodable, as well as with the additional property that no codeword is a subword of the catenation of two codewords. This hypothesis seemed to be supported by the fact that the size of the maximal comma-free code over a four-letter alphabet, where all words have length three, was found to be exactly 20. We now know, [13], that some aminoacids are encoded by more than one triplet (codon), and that none of the sets consisting of choosing one codon per aminoacid is comma-free. As Hayes remarked, while this is less elegant than any of the theoretical codes proposed, it provides higher error-tolerance: "With Gamow's overlapping codes, any mutation could alter three adjacent amino acids at once, possibly disabling the protein. Comma-free codes are even more brittle in this respect, since a mutated codon is likely to become nonsense and terminate the translation" [7].

While in this case Nature proved that mathematical theories may be beautiful and still wrong, comma-free codes and their generalizations remain interesting and much studied concepts [8, 11, 16, 18, 19]. More recent developments in biology show that, although genetic information is encoded in DNA, genes (coding segments) are usually interrupted by noncoding segments, formerly known as "junk segments". A generalization of comma-free codes, wherein a comma (noncoding segment) is defined as a word of length k, and no codeword (gene, or coding segment) is a subword of two other codewords separated by a comma, may be of mathematical but also of biological interest.

In this paper, we generalize the notion of comma-free codes to k-comma codes, and further, to k-spacer codes, which allow "commas" (corresponding to noncoding segments) of lengths $k \geq 0$, respectively less than or equal to k, between two codewords. Since k-comma codes are proper generalizations of comma-free codes and comma codes [3] (which allow commas of length one), it is natural to investigate their properties and the properties of their generalizations, k-comma intercodes, which are defined analogously to intercodes (which generalize the comma-free codes). As consequences, some properties of k-spacer codes are obtained from those of k-comma codes and k-comma intercodes. For example, a k-spacer code is an infix code, and hence a code. Also, due to our result, for some $k \geq 0$, if the length of the shortest words of a language L is not longer than k, then L cannot be a k-spacer code.

The paper is organized as follows. In Section 2, we give the formal definitions of k-comma codes and k-spacer codes, and show that they are in the family of infix codes. In Section 3, we generalize k-comma codes to k-comma intercodes, and obtain a hierarchical relationship among the families of bifix codes, k-comma intercodes, and infix codes. Moreover, we obtain several closure properties and the synchronously decipherability of the families of k-comma intercodes and provide a polynomial time algorithm to decide whether a given regular language is a k-comma intercode. As consequences, several closure properties of families of k-spacer codes and a polynomial time algorithm that determines whether a regular language is a k-spacer code are obtained. In Section 4, we generalize k-comma intercodes into n-k-comma intercodes and obtain hierarchical relationships among them. Moreover, we obtain a characterization of the families of 1-k-comma intercodes, and describe the family of 1-k-comma intercodes by using the classic notions of bordered words, unbordered words, and primitive words.

We end this section by some preliminary definitions and notations used in the paper. An alphabet Σ is a nonempty finite set of letters. A word over Σ is a sequence of letters in Σ. The length of a word w, denoted by $|w|$, is the number of letters in this word. The empty word, denoted by λ, is the word of length 0. A unary word is a word of the form a^j, $j \geq 1$, $a \in \Sigma$. The set of all words over Σ is denoted by Σ^*, and $\Sigma^+ = \Sigma^* \setminus \{\lambda\}$ is the set of all nonempty words. A language is a subset of Σ^*. The catenation of two languages $L_1, L_2 \subseteq \Sigma^*$, denoted by L_1L_2, is defined as $L_1L_2 = \{uv \mid u \in L_1, v \in L_2\}$.

A word $x \in \Sigma^*$ is called an infix of a word $u \in \Sigma^+$ if $u = zxy$ for some words $y, z \in \Sigma^*$. In this definition, if z and y are nonempty, then x is called a *proper* infix of u. Similarly, a word $x \in \Sigma^*$ is called a prefix (suffix) of a word $u \in \Sigma^+$ if $u = xy$ (resp. $u = zx$) for some word $y \in \Sigma^*$ (resp. $z \in \Sigma^*$). In addition, if y (resp. z) is nonempty, then x is called a *proper* prefix (resp. suffix) of u. For a word $u \in \Sigma^*$, the set of its prefixes (suffixes) is denoted by $\mathrm{Pref}(u)$ (resp. $\mathrm{Suff}(u)$). For a word $u \in \Sigma^*$, we denote the prefix (suffix) of length $n \geq 0$ of u by $\mathrm{pref}_n(u)$ (resp. $\mathrm{suff}_n(u)$). These notations can be naturally extended to languages, e.g., $\mathrm{Pref}(L) = \cup_{u \in L}\mathrm{Pref}(u)$.

A nonempty word $u \in \Sigma^+$ is said to be *primitive*, also known as *non-periodic*, if $u = v^n$ implies $n = 1$ for any $v \in \Sigma^+$. Any nonempty word can be written as a power of a unique primitive word, which is called the *primitive root* of the word.

It is well known that, if nonempty words $x, y, z \in \Sigma^+$ satisfy $xy = yz$, then there exist $\alpha, \beta \in \Sigma^*$ such that $\alpha\beta$ is primitive, $x = (\alpha\beta)^i$, $y = (\alpha\beta)^j\alpha$, and $z = (\beta\alpha)^i$ for some $i \geq 1$ and $j \geq 0$.

A nonempty word $u \in \Sigma^+$ is said to be *bordered* if there exists a nonempty word which is both proper prefix and proper suffix of u. A *bordered primitive word* is a primitive word which is bordered, and it can be written as xyx for some $x, y \in \Sigma^+$ [15].

2. K-comma codes

The classic notion of comma-free codes is defined as follows: *A language $L \subseteq \Sigma^+$ is called a comma-free code if $LL \cap \Sigma^+L\Sigma^+ = \emptyset$.* Recently, [3], the notion of comma codes was introduced for solving some language equations. A language $L \subseteq \Sigma^+$ is called a *comma code* if $L\Sigma L \cap \Sigma^+L\Sigma^+ = \emptyset$. It is clear that the following definition of k-comma codes is a natural generalization of these two notions, which can be interpreted as 0-comma codes and 1-comma codes, respectively.

Definition 2.1. For any $k \geq 0$, a set $L \subseteq \Sigma^+$ is called a k-comma code if $L\Sigma^kL \cap \Sigma^+L\Sigma^+ = \emptyset$.

In this section, we first show that a k-comma code is in fact a code (Corollary 2.1), and that, for any two integers $k_1, k_2 \geq 0$, the family of k_1-comma codes and the family of k_2-comma codes are not comparable (Proposition 2.1). Then, we extend the notion of k-comma codes to that of k-spacer codes, and show that the families of k-spacer codes form an infinite proper inclusion hierarchy (Proposition 2.2).

Intuitively, a k-comma code is a set L such that none of its words can be a proper infix of u_1vu_2 where u_1 and u_2 are words in L, and v is a "comma" of length k. It is clear that any codeword of a k-comma code must be longer than k. As examples, for any $k \geq 0$, $L = \{ab^ia \mid i > k\}$ is a k-comma code.

We first establish a relationship between comma-free codes and k-comma codes, for any $k \geq 0$.

Lemma 2.1. For a language $L \subseteq \Sigma^*$ and any $k \geq 0$, L is a k-comma code if and only if $L\Sigma^k$ is a comma-free code.

Proof:
We assume that $L\Sigma^k$ is a comma-free code, and suppose that L were not a k-comma code. Then there exist $w_1, w_2, w_3 \in L$, $v_1 \in \Sigma^k$, and $x, y \in \Sigma^+$ such that $w_1v_1w_2 = xw_3y$. By putting some $v_2 \in \Sigma^k$ at the ends of both sides, we can reach a contradiction with $L\Sigma^k$ being a comma-free code.

On the other hand, if $L\Sigma^k$ is not a comma-free code. Then we have $u_1v_1u_2v_2 = x'u_3v_3y'$ for some $u_1, u_2, u_3 \in L$, $v_1, v_2, v_3 \in \Sigma^k$, and $x', y' \in \Sigma^+$. Since y' is nonempty, we can cut the last k letters of both sides from this equation, and reach a contradiction that L is not a k-comma code. □

Recall that a nonempty set $L \subseteq \Sigma^+$ is an infix code if $L \cap (\Sigma^*L\Sigma^+ \cup \Sigma^+L\Sigma^*) = \emptyset$, and that a comma-free code is an infix code [18]. The following relationship leads us to the fact that k-comma codes are actually codes.

Lemma 2.2. For a language $L \subseteq \Sigma^*$, L is an infix code if and only if $L\Sigma^k$ is an infix code.

Proof:
The "only-if" direction is trivial because the family of infix codes is closed under concatenation. For the "if" direction, assume that $L\Sigma^k$ is an infix code, and suppose that L is not. Then there exist $u \in L$ and $x, y \in \Sigma^*$ such that $xuy \in L$ and $xy \neq \lambda$. Then for any $v_1 \in \Sigma^k$, $xuyv_1 \in L\Sigma^k$, which contains $uv_2 \in L\Sigma^k$ as its factor, where v_2 is the prefix of yv_1 of length k. Since $uv_2 \neq xuyv_1$, this is a contradiction. □

The following corollary is immediate.

Corollary 2.1. For any $k \geq 0$, a k-comma code is an infix code, and hence a code.

Lemma 2.1 implies that the families of k-comma codes are closely related to that of comma-free codes. However, the following result shows that any two of these families are incomparable, which means that, for any two integers n and m, $0 \leq n < m$, there exists an n-comma code which is not an m-comma code, and vice versa.

Proposition 2.1. Let $0 \leq n < m$. The family of n-comma codes and the family of m-comma codes are incomparable, but not disjoint.

Proof:
Let $L_1 = \{ab^{n+1}a\}$. We can easily verify that L_1 is an n-comma code but not an m-comma code. On the other hand, let us consider $L_2 = \{a^mba^{m+n}b\}$. This is an m-comma code but not an n-comma code. Moreover, there is a language which is both an n-comma code and an m-comma code. An example is $L_3 = \{ab^{m+1}a\}$. □

As a corollary, we cannot compare the classic family of comma-free codes with the other families of k-comma codes.

Corollary 2.2. For any $k \geq 1$, the family of k-comma codes and the family of comma-free codes are incomparable.

Now we loosen the restriction on the length of commas, and define k-spacer codes.

Definition 2.2. For any $k \geq 0$, a language L is called a *k-spacer code* if $L\Sigma^{\leq k}L \cap \Sigma^+ L \Sigma^+ = \emptyset$.

It is clear that, if a language is a k-spacer code, it is an i-comma code for all i, $0 \leq i \leq k$. Therefore, for any $k \geq 0$, a k-spacer code is a comma-free code and hence an infix code. Let S_k denote the family of k-spacer codes, and C_i denote the family of infix codes. Then we have the following relationship.

Proposition 2.2. $S_{k+1} \subset S_k \subset \cdots \subset S_0 \subset C_i$ holds.

Proof:
By definition, $S_{k+1} \subseteq S_k$ holds for any $k \geq 0$. To show that the inclusion is proper, note that $\{a^k b\}$ is in S_k but not in S_{k+1} for any $k \geq 0$. It is clear that S_0 is the family of 0-comma codes and $S_0 \subseteq C_i$ holds. Moreover, due to Proposition 2.1, there exists a 1-comma code that is an infix code but not a 0-comma code. Therefore, the inclusion $S_0 \subseteq C_i$ is proper. □

3. K-comma intercodes

Since a k-spacer code is an intersection of some k-comma codes, in this section, we obtain some closure properties (Proposition 3.7) and decidability results (Theorem 3.3) of the family of k-spacer codes, as consequences of those of k-comma codes. In coding theory, the notion of comma-free codes was extended to the more general one of intercodes [16].

Definition 3.1. For $m \geq 1$, a nonempty set $L \subseteq \Sigma^+$ is called an *intercode of index* m if $L^{m+1} \cap \Sigma^+ L^m \Sigma^+ = \emptyset$.

It is clear that an intercode of index 1 is a comma-free code.

Similarly, we introduce the notion of k-comma intercodes as a natural generalization of the notion of k-comma codes, and then obtain several basic properties of k-comma codes as consequences of those of k-comma intercodes. In particular, we first show that the k-comma intercodes are actually codes, and there exists an infinite inclusion hierarchy among the families of bifix codes, k-comma intercodes, and infix codes. Moreover, we obtain several results about k-comma intercodes, such as closure properties (Propositions 3.3, 3.4, and 3.5), synchronously decipherability (Proposition 3.8), and an efficient algorithm that determines whether a regular language is a k-comma intercode (Theorem 3.2).

The notion of k-comma intercodes is defined as follows.

Definition 3.2. For $k \geq 0$ and $m \geq 1$, a nonempty set $L \subseteq \Sigma^+$ is called a *k-comma intercode of index* m if $(L\Sigma^k)^m L \cap \Sigma^+ (L\Sigma^k)^{m-1} L \Sigma^+ = \emptyset$.

It is immediate that a k-comma intercode of index 1 is a k-comma code, and that a 0-comma intercode is an intercode. For any $k \geq 0$, a language L is called a *k-comma intercode* if there exists an integer $m \geq 1$ such that L is a k-comma intercode of index m. The family of k-comma intercodes is denoted by I_k.

We will prove that, for any $k \geq 0$, a k-comma intercode is actually a code. Recall that a nonempty set $L \subseteq \Sigma^+$ is a *bifix code* if $L \cap L\Sigma^+ = \emptyset$ (prefix code) and $L \cap \Sigma^+ L = \emptyset$ (suffix code).

Proposition 3.1. For any $k \geq 0$, a k-comma intercode is a bifix code.

Proof:
Let L be a k-comma intercode of index m for some $k \geq 0$ and $m \geq 1$. Suppose that L were not a prefix code. Then we have $u, w \in L$ such that $w = uv$ for some $v \in \Sigma^+$. This implies that for some $x_1, \ldots, x_m \in \Sigma^k$, $wx_1wx_2 \cdots x_m w = wx_1(wx_2 \cdots x_m u)v \in \Sigma^+(L\Sigma^k)^{m-1}L\Sigma^+$, which contradicts that L is a k-comma intercode of index m. In the same way, we can prove that L is a suffix code. Thus, L is a bifix code. □

Similar to Lemma 2.1, we establish a relationship between intercodes and k-comma intercodes.

Lemma 3.1. For a language $L \subseteq \Sigma^*$ and any integers $k \geq 0$ and $m \geq 1$, L is a k-comma intercode of index m if and only if $L\Sigma^k$ is an intercode of index m.

The families of intercodes of different indexes form an infinite proper inclusion hierarchy within the family of bifix codes, i.e., the family of intercodes of index m is a proper subset of the family of intercodes of index $m+1$, for any $m \geq 1$. Moreover, the family of all the intercodes of any index is a proper subset of the family of bifix codes [15]. In the following, we prove that such an infinite proper inclusion hierarchy exists among the families of k-comma intercodes of different indexes for any $k \geq 0$. We first prove the following lemma.

Lemma 3.2. Let L be a k-comma intercode for some $k \geq 0$. Then any codeword in L must be longer than k.

Proof:
Suppose u were a codeword in L of length at most k. Then, we can find words $x, y \in \Sigma^k$ with $ux = yu$. For any $m \geq 1$, $(ux)^m u = (yu)^m u$. This contradicts L being a k-comma intercode. □

Let $I_{k,m}$ denote the family of k-comma intercodes of index m, for any $k \geq 0$ and $m \geq 1$. We have the following hierarchies.

Theorem 3.1. $I_{k,1} \subset I_{k,2} \subset \cdots \subset I_{k,m} \subset \cdots \subset C_b$ holds for any $k \geq 0$.

Proof:
We first prove that, for any $k \geq 0$ and $m \geq 1$, every k-comma intercode of index m is a k-comma intercode of index $m+1$. Let L be a k-comma intercode of index m. By definition, we have $(L\Sigma^k)^m L \cap \Sigma^+(L\Sigma^k)^{m-1}L\Sigma^+ = \emptyset$. Suppose that L were not a k-comma code of index $m+1$. Then $(L\Sigma^k)^{m+1}L \cap \Sigma^+(L\Sigma^k)^m L\Sigma^+ \neq \emptyset$. That is, there exist $u_1, \ldots, u_{m+2} \in L$, $v_1, \ldots, v_{m+1} \in L$, $x_1, \ldots, x_{m+1}$, $y_1, \ldots, y_m \in \Sigma^k$, and $z_1, z_2 \in \Sigma^+$ such that $u_1x_1 \cdots x_{m+1}u_{m+2} = z_1v_1y_1 \cdots y_m v_{m+1}z_2$.

We claim that $|z_1| < |u_1|$ and $|z_2| < |u_{m+2}|$ must hold. Suppose $z_1 = u_1 z'$ for some $z' \in \Sigma^*$, then $x_1 \cdots x_{m+1} u_{m+2} = z' v_1 y_1 \cdots y_m v_{m+1} z_2$. Since v_1 is in L, we have $|v_1| > |x_1|$. Then, we can easily check that $v_2 y_2 \cdots v_{m+1}$ is an proper infix of $u_2 x_2 \cdots u_{m+2}$, a contradiction. Similarly, we can prove that $|z_2| < |u_{m+2}|$.

However, even if $|z_1| < |u_1|$ and $|z_2| < |u_{m+2}|$, we still have $v_1 y_1 \cdots y_m v_{m+1}$ in $\Sigma^+ u_2 x_2 \cdots x_m u_{m+1} \Sigma^+$, and hence $(L\Sigma^k)^m L \cap \Sigma^+ (L\Sigma^k)^{m-1} L\Sigma^+ \neq \emptyset$. This is a contradiction. Thus, $I_{k,m} \subseteq I_{k,m+1}$.

We then prove that this inclusion is proper by giving examples of languages $L \in I_{k,m+1} \setminus I_{k,m}$. Let $\Sigma = \{a, b\}$ and $u_i = ab^{i+k}a$ for some $i \geq 1$. Then, for some $x_1, \ldots, x_{m+1} \in \Sigma^k$, $L = \{u_1 x_1 \cdots u_{m+1} x_{m+1} u_{m+2}, u_2, u_3, \ldots, u_{m+1}\}$ satisfies the condition $(L\Sigma^k)^{m+1} L \cap \Sigma^+ (L\Sigma^k)^m L\Sigma^+ = \emptyset$, and hence $L \in I_{k,m+1}$. On the other hand, $L \notin I_{k,m}$, since $u_2 x_2 \cdots u_{m+1}$ is a proper infix of word $u_1 x_1 \cdots u_{m+1} x_{m+1} u_{m+2}$, and hence $(L\Sigma^k)^m L \cap \Sigma^+ (L\Sigma^k)^{m-1} L\Sigma^+ \neq \emptyset$.

Lastly, we can verify that $L' = \{aa, aba\}$ is a bifix code but not a k-comma intercode of index m for any $k \geq 0$ and $m \geq 1$. It is clear that L' cannot be a k-comma intercode of any index for $k \geq 2$. Then, for either $k = 0$ or $k = 1$, we have $aba(a^{k+2})^{m-1} a^k (aba) \in (L'\Sigma^k)^m L' \cap \Sigma^+ (L'\Sigma^k)^{m-1} L'\Sigma^+$ for any $m \geq 1$. Therefore, $I_{k,m} \subset C_b$. □

Although an intercode of index $m+1$ is not always an intercode of index m, we show in the following that, it is true for specific languages of the form $u\Sigma^k$.

Lemma 3.3. For a word $u \in \Sigma^*$ and an integer $m \geq 1$, $u\Sigma^k$ is an intercode of index m if and only if $u\Sigma^k$ is an intercode of index $m + 1$.

Proof:

It is well known that an intercode of index m is an intercode of index $m + 1$, but its converse implication is not always true. We prove that it is true for specific languages of the form $u\Sigma^k$. Under the assumption that $u\Sigma^k$ is an intercode of index $m + 1$, suppose that $u\Sigma^k$ were not an intercode of index m. Due to the assumption, Lemma 3.1 gives us that u is a k-comma intercode and hence $|u| > k$. There exists $x_1, \cdots, x_{m+1}, x'_1, \cdots, x'_m \in \Sigma^k$ and $y, z \in \Sigma^+$ such that

$$u x_1 u x_2 \cdots u x_m u x_{m+1} = y u x'_1 u x'_2 \cdots u x'_m z. \tag{1}$$

Note that $|u| + k = |y| + |z|$. We consider two cases depending on the length of y. If $|y| < |u|$ (i.e., $|z| > k$), let $z = u_s x_{m+1}$ with $u = u_p u_s$ for some $u_p, u_s \in \Sigma^+$. Then $u x_1 u x_2 \cdots u x_{m-1} u_p = y u x'_1 u x'_2 \cdots u x'_{m-1}$ and $u_s x_m u_p = u x'_m$. With these, we have

$$\begin{aligned} u x_1 u x_2 \cdots u x_{m-1} (u x_m)^2 u x_{m+1} &= u x_1 \cdots u x_{m-1} u_p (u_s x_m u_p)^2 u_s x_{m+1} \\ &= y u x'_1 u x'_2 \cdots u x'_{m-1} (u x'_m)^2 z. \end{aligned}$$

Thus, $u\Sigma^k$ would not be an intercode of index $m + 1$, a contradiction.

Now we consider the second case when $|y| \geq |u|$. Recall that $|u| > k$. Hence we can let $x_1 = y_s u_p$ and $u y_s = y$, where $u_p \in \mathrm{Pref}(u)$ and $y_s \in \mathrm{Suff}(y)$. We can see in Eq. 1 that x_2 also has the suffix u_p as $x_2 = w u_p$ for some $w \in \Sigma^*$. Then $u x'_1 = u_p u w$ and $u_p u x_3 \cdots u x_{m+1} = u x'_2 \cdots u x'_m z$, and we have

$$\begin{aligned} u x_1 (u x_2)^2 u x_3 \cdots u x_{m+1} &= u y_s u_p (u w u_p)^2 u x_3 \cdots u x_{m+1} \\ &= u y_s (u_p u w)^2 u_p u x_3 \cdots u x_{m+1} \\ &= y (u x'_1)^2 u x'_2 \cdots u x'_m z. \end{aligned}$$

Even in this case, we reached the same contradiction. □

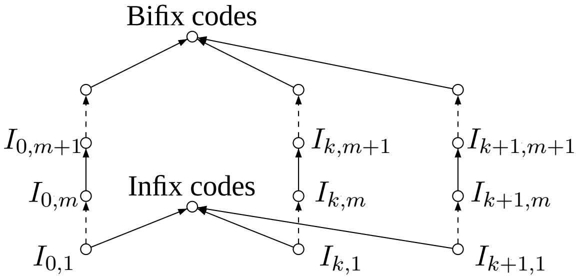

Figure 1. The inclusion hierarchy of the families of bifix codes, k-comma intercodes, and infix codes, where arrows indicate proper inclusion

Due to Theorem 3.1, the language L_1 considered in the proof of Proposition 2.1 is an n-comma intercode of any index. Moreover, we can verify that it is not an m-comma intercode for any index where $m > n$. On the other hand, the language L_2 in the same proof is an m-comma intercode of any index but not an n-comma intercode for any index where $n < m$. Hence, the following is clear.

Proposition 3.2. For any $k_1, k_2 \geq 0$ and $m_1, m_2 \geq 1$, the family of k_1-comma intercodes of index m_1 and the family of k_2-comma intercodes of index m_2 are incomparable unless $k_1 = k_2$.

Furthermore, due to Corollary 2.1 and Proposition 2.1, we know that, for any $k \geq 0$, there exists an infix code that is not a k-comma code. Therefore, the family of k-comma codes is a proper subset of the family of infix codes for any $k \geq 0$. Thus, we can draw the proper inclusion hierarchy of the families of bifix codes, k-comma intercodes, and infix codes as shown in Figure 1.

Next, we consider closure properties of the families of k-comma intercodes of index m for any $k \geq 0$ and $m \geq 1$ and the families of k-comma intercodes. Recall that a function $f : \Sigma_1^* \to \Sigma_2^*$ is called a *homomorphism* (on Σ_1^*) if $h(xy) = h(x)h(y)$ for all $x, y \in \Sigma_1^*$. The homomorphism f is *non-erasing* if $f(w) = \lambda$ implies $w = \lambda$. Then the *inverse non-erasing homomorphism* $f^{-1} : \Sigma_2^* \to 2^{\Sigma_1^*}$ is defined as: for $u \in \Sigma_2^*$, $f^{-1}(u) = \{v \in \Sigma_1^* \mid f(v) = u\}$, where f is non-erasing.

Proposition 3.3. For any $k \geq 0$ and $m \geq 1$, the families of k-comma intercodes of index m are not closed under union, catenation, +, complement, or non-erasing homomorphism. The families of k-comma intercodes are not closed under these operations either. In contrast, they are closed under reversal and intersection with an arbitrary set.

Proof:
Due to Theorem 3.1, we just need to show for each operation that the resulting languages of some k-comma codes under the operation may not be a bifix code, or not a k-comma intercode of index m for any $m \geq 1$. The union of two k-comma codes $\{ab^{1+k}a\}$ and $\{ab^{1+k}ab^{1+k}a\}$ is not a bifix code. We can easily verify that the catenation of AB of k-comma codes $A = \{aab^{1+k}a\}$ and $B = \{ab^{1+k}aab\}$ is not a k-comma intercode of index m for any $m \geq 1$. For any $L \subseteq \Sigma^+$, L^+ is not a bifix code. The complement of a k-comma code $\{ab^{1+k}a\}$ is not a bifix code. Consider alphabets $\Sigma_1 = \{a, b\}$ and $\Sigma_2 = \{a\}$, and let $f : \Sigma_1^* \to \Sigma_2^*$ be a non-erasing homomorphism defined as $f(a) = f(b) = a$. Then f maps a k-comma code $\{ab^{1+k}a, ab^{2+k}a\}$ onto $\{a^{3+k}, a^{4+k}\}$, which is not a bifix code.

By definition, it is clear that the families of k-comma intercodes of index m and the families of k-comma intercodes are closed under reversal or intersection with an arbitrary set. □

The closure properties of the family of intercodes and the families of k-comma intercodes for $k \geq 1$ under inverse non-erasing homomorphism are different.

Proposition 3.4. For any $m \geq 1$, the family of intercodes (0-comma intercodes) of index m is closed under inverse non-erasing homomorphism, and therefore the family of intercodes is closed under this operation.

Proof:
Let L be an intercode of index m over Σ_1. Suppose the family of intercodes of index m were not closed under inverse non-erasing homomorphism. Then, there exists a non-erasing homomorphism $f : \Sigma_2^* \to \Sigma_1^*$ such that $f^{-1}(L)$ is not an intercode of index m. This implies that there exist $u_1, \cdots, u_{m+1}, v_1, \cdots, v_m \in f^{-1}(L)$ such that $u_1 \cdots u_{m+1} \in \Sigma_2^+ v_1 \cdots v_m \Sigma_2^+$. Since f is non-erasing, $f(u_1) \cdots f(u_{m+1}) \in \Sigma_1^+ f(v_1) \cdots f(v_m) \Sigma_1^+$, a contradiction. □

For any positive integer k, the family of k-comma intercodes is not closed under non-erasing homomorphism.

Proposition 3.5. For any $k \geq 1$ and $m \geq 1$, the family of k-comma intercodes of index m is not closed under non-erasing homomorphism. Moreover, the family of k-comma intercodes is not closed under this operation.

Proof:
Consider alphabets $\Sigma_1 = \{a\}$ and $\Sigma_2 = \{a, b\}$, and let $f : \Sigma_1^* \to \Sigma_2^*$ be a homomorphism defined as $f(a) = ab^k$. We can verify that $L = \{ab^k ab^k\}$ is a k-comma code but $f^{-1}(L) = \{aa\}$ is not a k-comma intercode of index m for any $m \geq 1$. □

Proposition 3.3 says that the catenation of two k-comma codes is not always a k-comma intercode. So we investigate a condition under which the catenation of two languages A and B becomes a k-comma intercode under the assumption that $A \cup B$ is an infix code. Under this assumption, an element of AB could be a proper infix of an element of $AB\Sigma^k AB$ only in two ways as shown in Figure 2. The following results offer additional conditions on A and B, which make AB a k-comma code, and therefore k-comma intercode for any index, by preventing both cases in Figure 2 from occurring.

Proposition 3.6. For two languages $A, B \subseteq \Sigma^*$, if $A \cup B$ is a k-comma code, then AB is a k-comma intercode of any index.

Proof:
Suppose that AB were not a k-comma code. Then there exist $u_1, u_2, u_3 \in A$, $v_1, v_2, v_3 \in B$, and $w \in \Sigma^k$ such that $u_1 v_1 w u_2 v_2 = r u_3 v_3 s$ for some $r, s \in \Sigma^+$. Since k-comma codes are infix codes, $A \cup B$ is an infix code. Thus, we have the two cases shown in Figure 2. Nevertheless, they cause a contradiction with $A \cup B$ being a k-comma. Thus, AB is a k-comma code, and therefore a k-comma intercode for any index. □

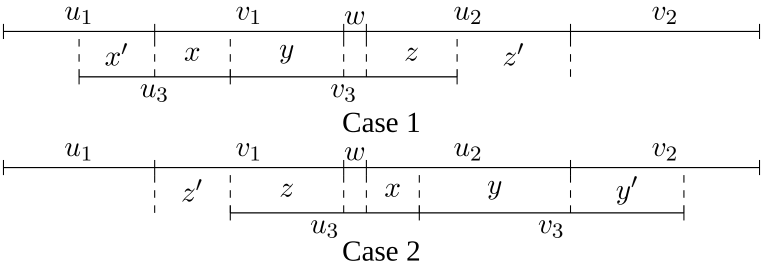

Figure 2. For $u_1, u_2, u_3 \in A$ and $v_1, v_2, v_3 \in B$, if $A \cup B$ is an infix code, u_3v_3 can be a proper infix of $u_1v_1wu_2v_2$ only in these two ways, where $w \in \Sigma^k$. Note that x' and y in Case 1 can be empty at the same time, and x and y' in Case 2 can be empty at the same time.

Now, we consider closure properties of the families of k-spacer codes. Since the family of 0-spacer codes is the family of comma-free codes, we only consider the cases when $k \geq 1$. By noticing the languages in the proofs of Propositions 3.3 and 3.5 are also k-spacer codes, the following result is immediate.

Proposition 3.7. For any $k \geq 1$, the family of k-spacer codes is not closed under union, catenation, +, complement, non-erasing homomorphism, or inverse non-erasing homomorphism. In contrast, it is closed under reversal and intersection with an arbitrary set.

Although the definitions and previous properties of k-comma intercodes are obtained for any $k \geq 0$, we show in the following that intercodes ($k = 0$) and their generalizations ($k \geq 1$) are different in terms of synchronous decoding delay. A code L is *synchronously decipherable* if there is a non-negative integer n such that for all $u, v \in \Sigma^*$ and $x \in L^n$, $uxv \in L^*$ implies $u, v \in L^*$. If a code L is synchronously decipherable, then the smallest such n is called the *synchronous decoding delay* of L. It is known that, for a code $L \subseteq \Sigma^+$, L is an intercode of index n if and only if L is synchronously decipherable with delay less than or equal to n [18]. In contrast, for any $k \geq 1$, k-comma intercodes do not have such a property.

Proposition 3.8. Let $L \subseteq \Sigma^+$ be a k-comma intercode of index n, for some $k \geq 1$ and $n \geq 1$. Then L is not necessarily synchronously decipherable with delay less than or equal to n.

Proof:
Consider $L = \{a^{k+2}b^k, ab^kab^k\}$, which is a k-comma intercode of index 1, and hence a k-comma intercode of any index. For any $n \geq 1$, we have $a^{k+2}b^k(ab^kab^k)^n = a^{k+1}(ab^kab^k)^n ab^k \in L^{n+1}$ and $(ab^kab^k)^n \in L^n$, but a^{k+1} and ab^k are not in L. Therefore, L is not with delay n. □

Since a k-spacer code is a comma-free code, it is synchronously decipherable with delay 1.

From the definition of k-comma intercodes, we can easily decide if a given regular language is a k-comma intercode of index m, for a given m, by using the closure properties of regular languages. A natural question is whether there exists a method that solves the problem efficiently. In the following, we show that there exists a polynomial time algorithm to do so.

Note that Han, Salomaa, and Wood [6] introduced an algorithm that decides if a given finite automaton (FA) accepts an intercode of a given index m in $m^2O(|Q|^2 + |\delta|^2)$ worst-case time (Lemma 3.2

in [6]). Furthermore, without the specification of m, their algorithm can determine whether the regular language given by an FA is an intercode for some index $m \geq 1$, and if the answer is positive, then it can find the smallest index m such that the language is an intercode of index m. The time complexity of this algorithm is $O(\log|Q|(|Q|^4 + |Q|^2|\delta|^2))$ in worst-case (Theorem 3.2 in [6]).

Due to Lemma 3.1, for a regular language given as a finite automaton and a given integer k, $k \geq 0$, we can determine whether L is a k-comma intercode of a given index m in $m^2 O(|Q|^2 + |\delta|^2)$ worst-case time. Due to Lemma 3.2, we first check if the shortest word of L is longer than k. If not, L can not be a k-comma intercode of any index. If the answer is yes, then we give an answer to the question by checking if $L\Sigma^k$ is an intercode of index m. Thus, we obtain the following result.

Lemma 3.4. Given an FA A and an index $m \geq 1$, we can determine whether $L(A)$ is a k-comma intercode of index m in $m^2 O(|Q|^2 + |\delta|^2)$ worst-case time.

Similarly, for some given $k \geq 0$, and without the specification of m, we can determine if a language given by an FA is a k-comma intercode of some index $m \geq 1$ such that the language is a k-comma intercode of index m but not of index $m - 1$.

Lemma 3.5. Given an FA A and some $k \geq 0$, in $O(\log|Q|(|Q|^4 + |Q|^2|\delta|^2))$ worst-case time, we can determine whether $L(A)$ is a k-comma intercode for some index $m \geq 1$, and if the answer is positive we can find the smallest index m such that $L(A)$ is a k-comma intercode of index m.

Furthermore, without the specification of k and m, we can find all k such that a language given by FA is a k-comma intercode of some index $m \geq 1$ such that the language is a k-comma intercode of index m but not of index $m - 1$. Since k must be shorter than the shortest words in the language, we just need to check all possible k and k is bounded by the size of the FA.

Theorem 3.2. Given an FA A, in $O(\log|Q|(|Q|^5 + |Q|^3|\delta|^2))$ worst-case time, we can determine whether $L(A)$ is a k-comma intercode for all $k \geq 0$ and index $m \geq 1$, and if the answer is positive we can find the smallest index m such that $L(A)$ is a k-comma intercode of index m.

We know that a language L cannot be a k-spacer code if its shortest words are not longer than k. Thus, given an FA A, to determine if $L(A)$ is a k-spacer code for some $k \geq 0$, we just need to find the length l of the shortest words of $L(A)$, and then, check if L is an i-comma code (i-comma intercode of index 1) for all i, $0 \leq i \leq k$, for some $k < l$. Since k-spacer codes form a proper inclusion hierarchy with respect to their index (Proposition 2.2), we can apply a binary search to find the largest k (if any) in the range from 0 to $l - 1$, and therefore L is a k'-spacer code for all $0 \leq k' \leq k$. Based on the analysis, we establish the following result.

Theorem 3.3. Given an FA A, in $O(\log|Q|(|Q|^3 + |Q||\delta|^2)$ worst-case time, we can determine whether $L(A)$ is a k-spacer code for any $k \geq 0$, and if the answer is positive we can find the largest such k.

4. N-k-comma intercodes

A language L is an n-*code* if every nonempty subset of L of size at most n is a code. The authors of [9] obtained several properties about the combinatorial structure of n-codes and showed that these codes form an infinite proper inclusion hierarchy, i.e., for any integer $n \geq 1$, the family of $(n+1)$-codes is a proper subset of the family of n-codes. Later, they applied similar constructions to prefix and suffix codes, and obtained n-ps-codes [10]. However, unlike the hierarchy of n-codes, the hierarchy of n-ps-codes collapses after only three steps, and turned out to be finite. In [12], the authors generalized the notions of intercodes to those of n-intercodes, established relationships among these codes, and obtained an infinite inclusion hierarchy including both intercodes and n-intercodes.

In this section, we consider n-k-comma intercodes. We show that, for any $k \geq 0$, there exists an infinite inclusion hierarchy of the families of n-k-comma intercodes and k-comma intercodes within the family of bifix codes (Theorem 4.1). Moreover, we give a characterization of the family of 1-k-comma intercodes for any $k \geq 0$ (Proposition 4.2). Lastly, we describe the family of 1-1-comma intercodes in terms of bordered words, unbordered words, and primitive words (Proposition 4.3).

An *n-k-comma intercode of index m* is a nonempty language $L \subseteq \Sigma^+$ such that every nonempty subset of L of cardinality at most n is a k-comma intercode of index m. For any $n \geq 1$ and $k \geq 0$, a language L is called an *n-k-comma intercode* if there exists an integer $m \geq 1$ such that L is an n-k-comma intercode of index m. Let $I_{n,k,m}$ denote the family of n-k-comma intercodes of index m over Σ and let $I_{n,k,\infty} = \bigcup_{m\geq 1} I_{n,k,m}$ denote the family of n-k-comma intercodes. We have that

$$I_{k,m} = \bigcap_{n\geq 1} I_{n,k,m} \text{ and } I_k = \bigcap_{n\geq 1} I_{n,k,\infty}.$$

Moreover, the following two lemmas are clear from the definition of n-k-comma intercode of index m.

Lemma 4.1. For any integers $n, m \geq 1$, and $k \geq 0$, $I_{n+1,k,m} \subseteq I_{n,k,m}$.

Lemma 4.2. For any integers $n, m \geq 1$, and $k \geq 0$, $I_{k,m} \subseteq I_{n,k,m}$.

In the following, for each $k \geq 0$, we obtain several hierarchical relationships among k-comma intercodes, n-k-comma intercodes, and bifix codes.

Theorem 4.1. For any $k \geq 0$ and every $n, m \geq 1$, the following statements hold true:

1. $I_{1,k,\infty}$ and the family of bifix codes are incomparable.
2. Every n-k-comma intercode with $n \geq 2$ is a bifix code.
3. $I_{k,m} = \cdots = I_{2m+2,k,m} = I_{2m+1,k,m} \subset \cdots \subset I_{2,k,m} \subset I_{1,k,m}$.
4. $I_{k,m} \subset I_{k,m+1}$.
5. $I_{n,k,1} \subseteq I_{n,k,2} \subseteq \cdots \subseteq I_{n,k,m} \subseteq \cdots$.
6. If $n \geq 2m+1$, $I_{n,k,m} \subset I_{n,k,m+1}$.
7. If $n \geq 2$ and $n \leq 2m+1$, $I_{n,k,m} \subset I_{n,k,m+1}$.
8. $I_{n+1,k,\infty} \subset I_{n,k,\infty}$.
9. If $n \geq 2$, then $I_{k,m} \subseteq I_{n,k,m} \subset I_{n,k,\infty} \subset C_b$ and $I_{k,m} \subset I_k \subset I_{n,k,\infty}$.

10. $I_{2,k,\infty} \subset I_{1,k,\infty} \cap C_b$.

Proof:
For (1), let us consider the two languages $L_1 = \{aa, aba\}$ and $L_2 = \{ab^{k+1}a, ab^{k+1}ab^{k+1}a\}$. We can verify that L_1 is a bifix code but not in $I_{1,k,\infty}$, while L_2 is in $I_{1,k,\infty}$ but not a bifix code.

For (2), assume that L is an n-k-comma intercode of index m with $n \geq 2$ for some $m \geq 1$. Suppose that L is not a bifix code. Then, there exists two words $u, v \in L$ such that $u = vz$ for some $z \in \Sigma^+$. Let L' be a subset of L of size n such that $v, u \in L'$. For some $x \in \Sigma^k$, we have that $(ux)^m u \in (L'\Sigma^k)^m L' \cap \Sigma^+(L'\Sigma^k)^{m-1}L'\Sigma^+$, a contradiction. This implies that $I_{n,k,m} \in C_b$, and hence $I_{n,k,\infty} \in C_b$.

For (3), due to Lemmas 4.1 and 4.2, it suffices to prove that (i) $I_{2m+1,k,m} \subseteq I_{k,m}$ and (ii) for any $1 \leq n \leq 2m+1$, $I_{n,k,m} \setminus I_{n+1,k,m} \neq \emptyset$.

We first prove (i). For $L \notin I_{k,m}$, then there exist $u_1, u_2, \cdots, u_m, u_{m+1} \in L$, $v_1, v_2, \cdots, v_m \in L$, $x_1, x_2, \cdots, x_m, y_1, y_2, \cdots, y_{m-1} \in \Sigma^k$, and $z, z' \in \Sigma^+$ such that

$$u_1x_1u_2x_2 \cdots u_mx_mu_{m+1} = zv_1y_1v_2y_2 \cdots v_{m-1}y_{m-1}v_mz',$$

which implies that $L \notin I_{2m+1,k,m}$. Hence, $I_{2m+1,k,m} \subseteq I_{k,m}$.

Then, we prove (ii). We give a construction for some languages $L_n \in I_{n,k,m} \setminus I_{n+1,k,m}$. Let $\Sigma = \{a, b\}$ and $u_i = ab^{k+i}a$ for $i \geq 1$. For some words $x_1, \ldots, x_{n+1} \in \Sigma^k$, define L_n in the following ways:
if $n \leq m$, then, as

$$\{u_2, u_3, \ldots, u_{n+1}, u_1x_1(u_2x_2)^{m-n+1}u_3 \cdots u_{n+2}\},$$

if $m < n < 2m$ and n is odd, then, as

$$\{u_jx_ju_{j+1} \mid j = 1, \ldots, n-1\} \cup \{u_{n+1}, u_nx_n(u_{n+1}x_{n+1})^{m-(n-1)/2}u_{n+2}\},$$

if $m < n < 2m$ and n is even, then, as

$$\{u_jx_ju_{j+1} \mid j = 1, \ldots, n-2\} \cup \{u_n, u_{n+1}, u_{n-1}x_{n-1}u_nx_n(u_{n+1}x_{n+1})^{m-n/2}u_{n+2}\},$$

if $n = 2m$, then, as

$$\{u_jx_ju_{j+1} \mid j = 1, \ldots, n+1\}.$$

We can easily verify that $L_n \in I_{n,k,m} \setminus I_{n+1,k,m}$.

Statement (4) is proven in Theorem 3.1.

For (5), if $L \in I_{n,k,m}$, then for any subset L' of L with $|L'| \leq n$, $L' \in I_{k,m}$. Statement (4) implies that $L' \in I_{k,m+1}$. Thus, $L \in I_{n,k,m+1}$.

For (6), statement (3) implies that $I_{n,k,m} = I_{k,m}$ since $n \geq 2m+1$. With statement (4) and Lemma 4.2, we have $I_{n,k,m} = I_{k,m} \subset I_{k,m+1} \subseteq I_{n,k,m+1}$.

To show (7), due to statement (5), we just need to show the inclusion is proper. We use the construction of languages L_n in (3), and we can verify that $L_{n-1} \in I_{n,k,m+1} \setminus I_{n,k,m}$.

For (8), $I_{n+1,k,\infty} \subseteq I_{n,k,\infty}$ is an immediate consequence of the definition. To prove the inequality, we give examples of languages $M_n \in I_{n,k,\infty} \setminus I_{n+1,k,\infty}$. We still use the same words u_i defined previously. For some words $x_1, \cdots, x_{n+1} \in \Sigma^k$, define M_n as

$$\{u_jx_ju_{j+1} \mid j = 1, \ldots, n\} \cup \{u_{n+1}x_{n+1}u_1\}.$$

We can verify that $M_n \in I_{n,k,\infty} \setminus I_{n+1,k,\infty}$ for any $n \geq 1$.

For (9), by definitions, the inclusions $I_{k,m} \subseteq I_{n,k,m} \subseteq I_{n,k,\infty}$ and $I_{k,m} \subseteq I_k \subseteq I_{n,k,\infty}$ are immediate. The inclusion $I_{n,k,\infty} \subseteq C_b$ follows from (2). The inequalities $I_{n,k,m} \neq I_{n,k,\infty}, I_{k,m} \neq I_k$, and $I_k \neq I_{n,k,\infty}$ follow from (7), (4), and (8), respectively. The inequality $I_{n,k,\infty} \neq C_b$ follows from (10).

For (10), we have $I_{2,k,\infty} \subseteq I_{1,k,\infty} \cap C_b$ by (2) and (8). For the inequality, as an example, M_1 constructed in (8) is a language in $I_{1,k,\infty} \cap C_b$, but not in $I_{2,k,\infty}$. □

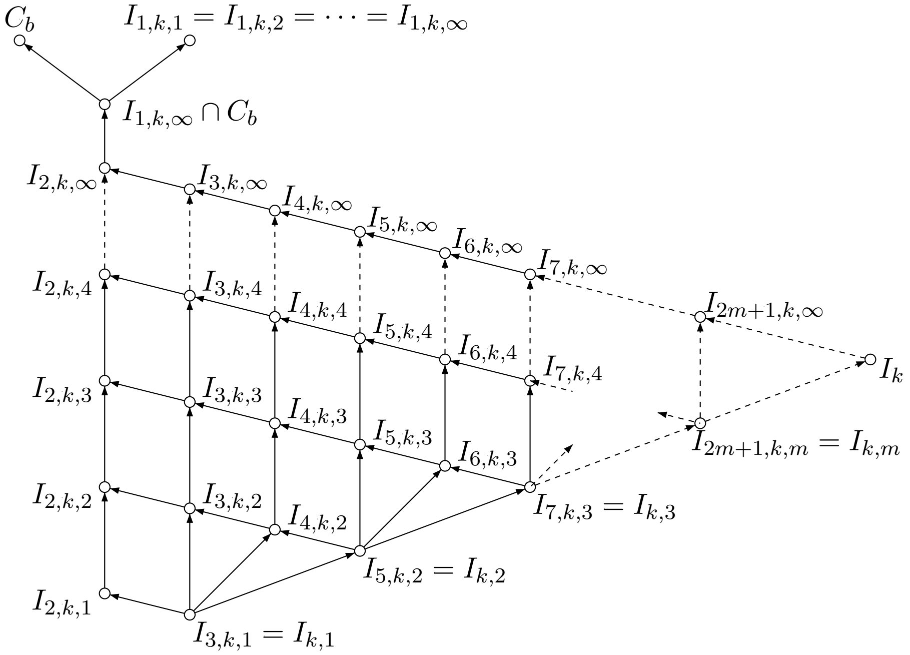

Figure 3. The inclusion hierarchy of k-comma intercodes, n-k-comma intercodes, and bifix codes, where arrows indicate proper inclusion.

From statements 5, 6, and 7 in the previous theorem, we obtain the following corollary.

Corollary 4.1. For any integers $n \geq 2$ and $k \geq 0$, the following strict set inclusion hierarchy exists

$$I_{n,k,1} \subset I_{n,k,2} \subset \cdots I_{n,k,m} \subset \cdots .$$

This hierarchy does not exist among the families of 1-k-comma intercodes as proven below.

Proposition 4.1. $I_{1,k,1} = I_{1,k,2} = \cdots = I_{1,k,m} = \cdots$.

Proof:
Due to statement 5 in Theorem 4.1, it suffices to prove $I_{1,k,m+1} \subseteq I_{1,k,m}$. Let $L \in I_{1,k,m+1}$. Then for any $u \in L$, $\{u\}$ is a k-comma intercode of index $m + 1$. Lemma 3.1 implies that $u\Sigma^k$ is an intercode of index $m + 1$, and this language is an intercode of index m due to Lemma 3.3. We apply Lemma 3.1 once again to obtain $\{u\}$ is a k-comma intercode of index m. Therefore, $L \in I_{1,k,m}$. □

We notice that the resulting languages in the proof of Propositions 3.3 are neither a bifix code nor a 1-k-comma intercode of any index. Therefore, for any $n \geq 1$, $k \geq 0$, and $m \geq 1$, the family of n-k-comma intercode of index m is not closed under union, catenation, +, complement, or non-erasing homomorphism. Similar to the proofs of Propositions 3.4 and 3.5, we can show that the family of n-intercodes (n-0-comma intercodes) of any index is closed under inverse non-erasing homomorphism, while, for any $k \geq 1$, the family of n-k-comma intercodes of any index is not closed under the operation.

Let Q be the set of all primitive words. It is known that the set of 1-intercodes of index m is equal to the $2^Q \setminus \emptyset$ for any $m \geq 1$ [12]. In the next proposition, we show a stronger result. For any $k \geq 0$, the family of 1-k-comma intercodes is equal to $2^{X_k} \setminus \emptyset$, where X_k is defined as:

$$X_k = \{u \in \Sigma^+ \mid uvu \cap \Sigma^+ u \Sigma^+ = \emptyset \text{ where } v \in \Sigma^k\}.$$

Note that, $X_0 = Q$.

Proposition 4.2. For any $k \geq 0$, a language L is a 1-k-comma intercode if and only if $L \in 2^{X_k} \setminus \emptyset$.

Proof:
Due to Proposition 4.1, we just need to show that L is a 1-k-comma intercode of index 1 if and only if $L \in 2^{X_k} \setminus \emptyset$.

If L is a 1-k-comma intercode of index 1, then, for every $u \in L$, $\{u\}$ is a k-comma intercode of index 1. Suppose that $L \notin 2^{X_k} \setminus \emptyset$. Then, there exists a word $w \in L$ such that $w \notin X_k$. Thus, $wvw \cap \Sigma^+ w \Sigma^+ \neq \emptyset$ for some $v \in \Sigma^k$, a contradiction to $\{w\}$ being a k-comma intercode of index 1.

For the converse implication, let L be a non-empty subset of X_k. Suppose there were a word $u \in L$ such that $\{u\}$ is not a k-comma intercode of index 1. Then, $uvu \in \Sigma^+ u \Sigma^+$ for some $v \in \Sigma^k$, which implies that $u \notin X_k$, a contradiction. □

In the following, we give a characterization of X_1 in terms of bordered words, unbordered words, and primitive words. It is clear that, no unary word can be in X_1, and the set of all unbordered words of length at least 2, denoted by $U^{>1}$, is a subset of X_1. Let $N_{(>1)}$ denote the set of all non-primitive words whose primitive root is of length at least 2. The next result shows that no word u in $N_{(>1)}$ can be a proper infix of uau, for any $a \in \Sigma$.

Lemma 4.3. $N_{(>1)} \subseteq X_1$.

Proof:
Suppose that there were $u \in N_{(>1)}$ such that $u \notin X_1$. Let $u = g^i$ for some primitive word g of length at least 2 and $i > 1$. Also we can let $u = u_s a u_p$ for some $u_s \in \mathrm{Suff}(u)$, $a \in \Sigma$, and $u_p \in \mathrm{Pref}(u)$. The equation $g^i = u_s a u_p$ implies that this a is inside one and only one of these g's. Since g^2 cannot overlap with g in any nontrivial way, either u_s or u_p is a power of g. We only consider the case when $u_s = g^j$ for some $j \geq 1$; the other can be proved in a similar way. Then $au_p = g^{i-j}$. Since $u_p \in \mathrm{Pref}(g^i)$, this means g is a power of a, a contradiction with the primitivity of g. □

Let Q_B be the set of all bordered primitive words. Any word in Q_B can be written as $w = (\alpha\beta)^k\alpha$ for some primitive word $\alpha\beta$, and $k \geq 1$. We partition Q_B into two sets. The first one, $Q_B^{(=1)}$, denotes the set of all bordered primitive words w that can be written as $(\alpha\beta)^k\alpha$ with $|\beta| = 1$. The second one is simply the complement, $Q_B^{(>1)} = Q_B \setminus Q_B^{(=1)}$. For example, $aaabaa, abbabba \in Q_B^{(>1)}$ while $aabaabaa \in Q_B^{(=1)}$. This is because even though we can regard $aabaabaa$ as $\alpha\beta\alpha$ with $\alpha = a$ and $\beta = abaaba$, we can also consider it as $(\alpha'\beta')^2\alpha'$, where $\alpha' = aa$ and $\beta' = b$.

The next result shows that every bordered primitive word w that can only be written as $(\alpha\beta)^k\alpha$ such that $\alpha\beta$ is primitive, $k \geq 1$, and $|\beta|$ cannot be 1, cannot be a proper infix of waw for any $a \in \Sigma$. Formally, we have

Lemma 4.4. $Q_B^{(>1)} \subseteq X_1$.

Proof:
Suppose that there exists $u \in Q_B^{(>1)}$ but $u \notin X_1$. This means that $u = u_s a u_p$ for some $u_s \in \mathrm{Suff}(u)$ and $u_p \in \mathrm{Pref}(u)$ and $a, b \in \Sigma$ such that $u = u_p b u_s$. The Parikh vector [14] of a word contains the occurrences of each letter in Σ. Since the Parikh vectors of u_p and u_s together contain the same number of occurrences of each letter in $u_s a u_p$ and $u_p b u_s$, we can obtain $a = b$ and hence $u = u_p a u_s$. Due to a well known result mentioned in Section 1, there exist $\alpha, \beta \in \Sigma^*$ such that $u_s a = (\alpha\beta)^i$ and $u_p = \alpha(\beta\alpha)^j$ for some $i \geq 1$ and $j \geq 0$ and $\beta\alpha$ is primitive. Then $ua = u_p a u_s a = u_p a(\alpha\beta)^i = \alpha(\beta\alpha)^{i+j}a$, and hence the suffix of length $|\alpha\beta| + 1$ of ua is $b\alpha\beta = \beta\alpha a$. Again, based on the Parikh vector of this suffix, $b = a$, i.e., $a\alpha\beta = \beta\alpha a$. Note that $|\beta| \geq 2$ because $u \in Q_B^{(>1)}$ and hence a is a proper suffix of β. Therefore, this equation means that $\beta\alpha$ overlaps with its square in a nontrivial way, a contradiction with its primitivity. □

The next result states that any word w that is either a unary word or a bordered primitive word that can be written as $(\alpha\beta)^k\alpha$ with $\alpha\beta$ being primitive, $k \geq 1$, and $|\beta| = 1$, can be a proper infix of waw for some $a \in \Sigma$.

Lemma 4.5. $\left(Q_B^{(=1)} \cup \{a^i \mid a \in \Sigma, i \geq 1\}\right) \cap X_1 = \emptyset$.

Proof:
As mentioned above, any unary word cannot be in X_1. Let $w \in Q_B^{(=1)}$. By definition, there exist $\alpha \in \Sigma^+$ and $b \in \Sigma$ such that αb is primitive and $w = (\alpha b)^k\alpha$ for some $k \geq 1$. Then w is a proper infix of wbw, and hence $w \notin X_1$. □

Note that

$$\Sigma^+ = \underbrace{N_{(>1)} \cup \{aa^+ \mid a \in \Sigma\}}_{\text{non-primitive}} \cup \underbrace{\Sigma \cup U^{>1} \cup Q_B^{(=1)} \cup Q_B^{(>1)}}_{\text{primitive}}.$$

As a consequence of Lemmas 4.3, 4.4, and 4.5, we have the following proposition.

Proposition 4.3. $X_1 = U^{>1} \cup Q_B^{(>1)} \cup N_{(>1)}$.

This proposition, by using several classic notions, characterizes the set of all words u that cannot be a proper infix of uau for any $a \in \Sigma$, as being either unbordered words of length greater than 1, or bordered primitive words of the form $(\alpha\beta)^k\alpha$ such that $\alpha\beta$ is primitive, $k \geq 1$, and $|\beta|$ cannot be 1, or non-primitive words whose primitive root has length longer than 1.

5. Conclusion

In this paper, we introduced the notion of k-comma codes, a generalization of comma-free codes, as well as the notion of k-spacer codes, and k-comma intercodes.

We established some relationships among families of k-comma codes, k-comma intercodes, infix codes, and bifix codes. Also, we obtained several closure properties of families of k-comma intercodes, and showed that we can determine efficiently whether a regular language given by a finite automaton is a k-comma intercode of index m for any $k \geq 0$ and $m \geq 1$, or a k-spacer code for any $k \geq 0$.

Lastly, we introduced the notion of n-k-comma intercodes and obtained several hierarchical relationships among families of n-k-comma intercodes. Moreover, we gave a characterization of the family of 1-k-comma intercodes for any $k \geq 0$, and describe the family of 1-1-comma intercodes in terms of several classic notions.

Future work includes experimental testing of, e.g., whether or not the language of genes of a certain organism is indeed a k-spacer code for some value k.

References

[1] Berstel, J., Perrin, D.: *Theory of Codes*. Academic Press. Inc., Orlando, Toronto. (1985)

[2] Crick, H.C., Griffith, J.S., Orgel, L.E.: Codes without commas. *Proc. Nat. Acad. Sci.* **43** (1957) 416-421

[3] Cui, B., Kari, L., Seki, S.: On the reversibility of parallel insertion, and its relation to comma codes. *Proc. of CAI 2009*. LNCS **5725**, 204-219

[4] Eastman, W. L.: On the construction of comma-free codes. *IEEE Trans. Inform. Theory*. **11** (1965) 263-267

[5] Golomb, S.W., Gordon, B., Welch, L.R.: Comma-free codes. *Canadian Journal of Mathematics*. **10** (1958) 202-209

[6] Han, Y.-S., Salomaa, K., Wood, D.: Intercode regular languages. *Fundamenta Informaticae*. **76** (2007) 113-128

[7] Hayes, B.: The invention of the genetic code. *American Scientist*. 86:8-14, 1998

[8] Hsieh, C. Y., Hsu, S. C., Shyr, H. J.: Some algebraic properties of comma-free codes. *RIMS Kenkyuroku*. **697** Japan (1989) 57-66

[9] Ito, M., Jürgensen, H., Shyr, H. J., Thierrin, G.: Anti-commutative languages and n-codes. *Discrete Applied Math*. **24** (1989) 187-196

[10] Ito, M., Jürgensen, H., Shyr, H. J., Thierrin, G.: N-prefix-suffix languages. *Intern. J. Computer Math*. **30** (1989) 37-56

[11] Jürgensen, H., Konstantinidis, S.: Codes. In Rozenberg, G., Salomaa, A. (eds): *Handbook of Formal Languages*, vol. **I**, 511-607. Springer-Verlag, Berlin, 1997.

[12] Jürgensen, H., Yu, S. S.: Relations on free monoids, their independent sets, and codes. *Intern. J. Computer Math*. **40** (1991) 17-46

[13] Lewin, B.: *Genes IX*. Jones and Bartlett Publishers, 2007

[14] Parikh, R.J.: On context-free languages. *Journal of the Association for Computing Machinery*. **13** (1966) 570- 581

[15] Shyr, H. J.: Free Monoids and Languages. *Lecture Notes*, Institute of Applied Mathematics, National Chung-Hsing University, Taichung, Taiwan. (2001)

[16] Shyr, H. J., Yu, S. S.: Intercodes and some related properties. *Soochow J. Math*. **16** No.1 (1990) 95-107

[17] Watson, J.: *Genes, Girls and Gamow: After the Double Helix*, Oxford University Press, 2001

[18] Yu, S. S.: Languages and Codes. *Lecture Notes*, Department of Computer Science, National Chung-Hsing University, Taichung, Taiwan 402. (2005)

[19] Yu, S. S.: A characterization of intercodes. *Intern. J. Computer Math*. **36** (1990) 39-48