Computer Communication & Collaboration (Vol. 2, Issue 4, 2014)
ISSN 2292-1028(Print) 2292-1036(Online)
Submitted on July 28, 2014
DOIC: 2292-1036-2014-04-001-37

Homomorphism Preserving Some Generalizations of Comma-free Codes[1]

Chunhua Cao

School of Mathematics and Statistics, Yunnan University at Kunming,
Cuihu North Road 2#, Kunming, 650091, China
Email: chhcaoyd@gmail.com

Haiyan Liu(Correspondence author)

School of Mathematics and Statistics, Yunnan University at Kunming,
Cuihu North Road 2#, Kunming, 650091, China
Email:liuhaiyanstu@163.com

Di Yang(Correspondence author)

School of Information, Yunnan University of Finance and Economics,
Longquan Road 237#, Kunming, 650221, China
Email: yangdi65@yahoo.cn

Li Jia

Information Administration Center, Yunnan University of Finance and Economics,
Longquan Road 237#, Kunming, 650221, China

Abstract: In this paper, we mainly consider the homomorphism which preserves k-space codes, 1-comma-free codes, 2-comma-free code,1-k-comma codes, 2-k-comma codes, 2-k-comma intercodes, 1-intercodes and 2-intercodes.

Keywords: 1-comma-free code, 2-comma-free code, 1-k-comma code, 2-k-comma code, 2-k-comma intercode

[1] The research is supported by a National Natural Science Foundation of China \# 11261066 and a Foundation for the Forth Yunnan University Key Teacher \# XT412003.

1. Introduction

People use code in many areas such as information processing and communication, data processing, 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 proposed for different purpose, such as prefix code, infix code(see [7]), comma-free code and intercede. For instance, the definition of comma-free codes (see [8,9]) followed the 1953 discovery of the double-helical structure of DNA(see [10]), as a proposed mathematical solution to a problem which arose in connection with protein synthesis. Algebraic properties of comma-free code are investigated by many researches (see [3,5]). The family of comma-free codes is a proper subfamily of infix codes.

While Nature proved that mathematical theories may be beautiful and still wrong, comma-free codes and their generalizations remain interesting and much studied concepts (see [4,6,8]).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". So, in [1],the authors generalized the notion of comma-free codes to k-comma codes, and further, to k-spacer codes($k \geq 0$), gave some related algebraic properties , pointed that the family of k-comma codes(spacer codes) is not closed under non-erasing homomorphism. Moreover, the family of k-comma intercodes is not closed under this operation.

The motivation for property-preserving homomorphism comes from much older research. Concerning the specific property, property-preserving homomorphism are applied to generate words or languages with such a property. By applying iterated homomorphism on certain words, one can generate infinite words. Some studies on problems concerning the behavior of homomorphism with respect to prefix code, infix code, comma-free code and intercode have been done in [2,3,7]. The paper consider the non-erasing homomorphism and inverse non-erasing homomorphism which preserve some generalizations of comma-free codes— k-space codes, 1-comma-free codes, 2-comma-free code,1- k-comma codes, 2- k-comma codes, 2- k-comma intercodes, 1-intercodes and 2-intercodes).

2. Definition and preliminary

Let X be a non-empty finite set of letters, which is called an alphabet and let $|X| \geq 2$ be the number of letters in X . Any finite string over X is called a *word* . The word that contains no letter is called the empty word, denoted by 1. The set of all words over X is denoted by X^*. Let $X^+ = X^* \setminus 1$. Any non-empty subset of X^+ is called a language. We denote the cardinality of a language A by $|A|$.The catenation of two languages $L_1, L_2 \subseteq X^+$, denoted by L_1L_2, is defined as $L_1L_2 = \{uv \mid u \in L_1, v \in L_2\}$. Let $L^1 = L$ and $L^k = L^{k-1}L$ for $k \geq 2$. For any word $w \in X^+$, let $|w|$ be the number of letters that occur in w and |1| = 0. For any $x, y \in X$, if $y = xz$ for some $z \in X^*$, then x is called a prefix of y, denoted by $x \leq_p y$. If $z \in X^+$, x is called a proper prefix and denoted by $x <_p y$. A language L is a prefix code if $L \cap LX^+ = \varnothing$. Dually, we can define $x \leq_s y$ if x is a suffix of y and $x <_s y$ if x is a proper suffix of y. A language L is a suffix code if $L \cap X^+L = \varnothing$. That is, if a language L is a prefix code(suffix code), then no code word can be the proper prefix(suffix) of another code word in L. We call x is an infix of the word y, if $y = uxv$ for some $u, v \in X^*$. A word x is called a proper infix of y if $uv \neq 1$. Let $E(y)$ be the set of all proper infix of y. That is $E(y) = \{x \mid y = uxv, uv \neq 1\}$. A language L is an infix code if for all $x, y, u \in X^*$, $u \in L$ and $xuy \in L$ together imply $xy = 1$. Thus, if L is an infix code, then no code- word can be the proper infix of another codeword in L. We first give the formal definition of the homomorphism which preserve families of languages and some necessary definitions.

Definition 2.1. A mapping h be a monomorphism from X^* into X^* is said to be a homomorphism or simply an endomorphism, if

$$h(xy) = h(x)h(y) \quad \forall x, y \in X^*.$$

If $h(x) = h(y)$ implies $x = y$ for all $x, y \in X^*$, then h is called a momomorphism. Let $\Im$ be a family of languages over X and $h(A) \in \Im$ for any $A \in \Im$, then we say that h preserves $\Im$.

Definition 2.2. A language L is a comma-free code if

$$L^2 \cap X^+LX^+ = \varnothing.$$

Thus, if a language L is a comma-free code, then no codeword is a subword of the catenation of two codewords.

Definition 2.3. A language L is an intercode if

$$L^m \cap X^+ L^{m-1} X^+ = \varnothing \text{ where } m \ge 1.$$

The integer m is called the index of L. Intercedes of index one is just comma-free codes. So it is a generalization of comma-free code. Every intercede of index m is an intercede of index $m+1$.

Definition 2.4. For a given positive integer k, a language L is a k-comma code if

$$LX^k L \cap X^+ L X^+ = \varnothing.$$

If $k=0$, a k-comma code is a comma-free code. Intuitively, a k-comma code is a language L such that none of its words can be a proper infix of $u_1 v u_2$ 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.

Definition 2.5. For any $k \ge 0$, a language L is a k-space code if

$$LX^{\le k} L \cap X^+ L X^+ = \varnothing.$$

From the definition, we note that a k-spacer code means that it is an i-comma code for all $0 \le i \le k$. Therefore, for any $k \ge 0$, a k-spacer code is a comma-free code and hence an infix code. A k-spacer code is an intersection of some k-comma codes.

Definition 2.6. A language L is a k-comma intercode of index m if

$$(LX^k)^m L \cap X^+ (LX^k)^{m-1} LX^+ = \varnothing.$$

The integer m is called the index of L. A k-comma intercede of index 1 is a k-comma code.

For a fixed $n \ge 1$, an n-comma-free code (intercode, k-comma code, k-spacer, k-comma intercode) is a language L such that every subset of L with n elements is a comma-free code (intercode, k-comma code, k-comma intercode).

Lemma 2.7[1] Every k-comma code is an infix code, hence a prefix code.

3. Main results

Theorem 3.1. Let $h: X^* \to X^*$ be a monomorphism. If $h(X)$ is a k-space code for any $k \geq 0$, then h preserves k-spacer codes.

Proof . Suppose L is a k-spacer code, but $h(L)$ isn't a k-spacer code. Then there exist $u, v, w \in L$, $z \in X^{\leq k}$ and $x, y \in X^+$ such that $h(u)zh(v) = xh(w)y$. Let $u = u_1 \ldots u_n$, $v = v_1 \ldots v_m$, $w = w_1 \ldots w_l$ where $u_1, \ldots, u_n$, $v_1, \ldots, v_m$, $w_1, \ldots, w_l \in X$. Hence $h(u_1) \ldots h(u_n) zh(v_1) \ldots h(v_m) = xh(w_1) \ldots h(w_l) y$. Then one of the following cases holds: (1) $x = h(u_1) \ldots h(u_i) x_1$, where $x_1 \leq_p h(u_{i+1})$, $0 \leq i \leq n$; (2) $x = h(u_1) \ldots h(u_n) z_1$ where $z_1 <_p z$; (3) $x = h(u_1) \ldots h(u_n) zh(v_1) \ldots h(v_j) y_1$ where $y_1 \leq_p h(v_{j+1}), 0 \leq j \leq m$.

For the case (1), we divide the proof into the following two cases:

(1-1) If $x_1 = 1$ or $x_1 = h(u_{i+1})$, without loss of generality, we assume $x_1 = 1$.

(1-1-1) If $i = n$, then $zh(v_1) \ldots h(v_m) = h(w_1) \ldots h(w_l) y$. we divide the proof into the following two cases:

Case 1 If $|h(w_1)| < |zh(v_1)|$, then there exists $f_1 \in X^+$ such that $zh(v_1) = h(w_1) f_1$. Hence we have $h(w_1) zh(v_1) = h(w_1) h(w_1) f_1$, which contradicts $h(X)$ is a k-spacer code.

Case 2 If $|h(w_1)| > |zh(v_1)|$, then there exists $f_2 \in X^+$ such that $h(w_1) = zh(v_1) f_2$. Since $h(X)$ is a k-spacer code, then it is an infix code. This is a contradiction. If $|h(w_1)| = |zh(v_1)|$, then we consider the following two cases. (i) If $z = 1$, then $h(w_1) = h(v_1)$. Since h is injective, then $w_1 = v_1$. Similarly, we have $w_n = v_n$ or

$w_l = v_l$. So $y = 1$ or w is a proper infix of v. This contradicts $y \in X^+$ or L is a k-space code. (ii) If $z \neq 1$, then $h(w_1)$ has a proper infix of $h(v_1)$, which contradicts $h(X)$ is an infix code.

(1-1-2) If $1 \le i \le n$, then $h(u_{i+1}) \cdots h(u_n) z h(v_1) \cdots h(v_m) = h(w_1) \cdots h(w_l) y$. Hence we have $h(u_{i+1}) = h(w_1), h(u_{i+2}) = h(w_2), \cdots$. In fact, if $h(u_{i+1}) <_p h(w_1)$ or $h(w_1) <_p h(u_{i+1})$, then $h(X)$ is not an infix code, which contradicts $h(X)$ is a k-spacer code. Hence we have $h(w_1) = h(u_{i+1})$, we consider the following two cases: (a) If $l \le n - i$, then $h(u_{i+1}) = h(w_1), \cdots, h(u_{i+l}) = h(w_l)$. Since h is injective, then $u_{i+1} = w_1, \cdots, u_{i+l} = w_l$. So w is an infix of u. This is a contradiction. (b) If $l > n - i$, then $z h(v_1) \cdots h(v_n) = h(w_{n-i+1}) \cdots h(w_l) y$. For the case: $|h(w_{n-i+1})| < |z h(v_1)|$ and $|h(w_{n-i+1})| = |z h(v_1)|$, we have contradictions just like (1 − 1 − 1).

(1-2) If $x_1 \neq 1$ and $x_1 \neq h(u_{i+1})$, then $x_2 h(u_{i+2}) \cdots h(u_n) z h(v_1) \cdots h(v_m) = h(w_1) \cdots h(w_l)\ y$ where $x_1 x_2 = h(u_{i+2})$. We divide the proof into the following two cases:

(1-2-1) If $i = n - 1$, then we have $x_2 z h(v_1) \cdots h(v_m) = h(w_1) \cdots \cdots h(w_l) y$.

Case 1 If $|h(w_1)| < |x_2 z h(v_1)|$, then $h(w_1)$ is a proper infix of $h(u_n) z h(v_1)$, which contradicts the fact $h(X)$ is a k-spacer code.

Case 2 If $|h(w_1)| \ge |x_2 z h(v_1)|$, then $h(w_1)$ has a proper infix $h(v_1)$. This is a contradiction.

(1-2-2) If $0 \le i \le n-2$, then $x_2 h(u_{i+2}) \cdots h(u_n) z h(v_1) \cdots h(v_m) = h(w_1) \cdots h(w_l)\ y$.

Case 1 If $|h(w_1)| < |x_2 h(u_{i+2})|$, then $h(w_1)$ is a proper infix of $h(u_{i+1}) h(u_{i+2})$. This is a contradiction.

Case 2 If $|h(w_1)| \geq |x_2 h(u_{i+2})|$, then $h(w_1)$ has a proper infix of $h(u_{i+2})$. This is a contradiction.

(2) If $x = h(u_1)\cdots h(u_n)z_1$ for some $z_1 <_p z$, then $z_2 h(v_1)\cdots h(v_m) = h(w_1)\cdots h(w_l)y$ where $z_1 z_2 = z$ and $z_1, z_2 \in X^+$. Considering $|h(w_1)|$ and $|z_2 h(v_1)|$, we have contradictions just like (1 − 1 − 1).

The case (3) is similar to the case (1).

From all above, we can obtain that h preserves k-space codes. This completes the proof.

Theorem 3.2. Let $h: X^* \to X^*$ be a monomorphism. If $h(X)$ is a comma-free code, then h preserves 1-comma-free codes.

Proof. Suppose L is a 1-comma-free code, but $h(L)$ isn't a 1-comma-free code. There exists $u \in L$ such that $\{u\}$ is comma-free code, but $\{h(u)\}$ is not a comma-free code. Then there exists $x, y \in X^+$ such that $h(u)h(u) = xh(u)y$. Let $u = u_1 \cdots u_n$ for some $u_1, \cdots, u_n \in X$. Then $h(u_1)\cdots(u_n)h(u_1)\cdots h(u_n) = xh(u_1)\cdots h(u_n)y$. Let $x = h(u_1)\cdots h(u_i)x_1$ for some $x_1 \leq_p h(u_{i+1})$ and $0 \leq i \leq n$.

(1) If $x_1 = 1$ or $x_1 = h(u_{i+1})$, without loss of generality, we assume $x_1 = 1$. This implies that $h(u_{i+1})\cdots h(u_n)h(u_1)\cdots h(u_n) = h(u_1)\cdots h(u_n)y$. Since $h(X)$ is a comma-free code, then it is an infix code. Hence $h(u_{i+1}) = h(u_1), \cdots h(u_i) = h(u_n)$. Since h is injective, then $\{u\}$ is not a comma-free code, which contradicts L is a 1-comma-free code.

(2) If $x_1 \neq 1$ and $x_1 = h(u_{i+1})$, then $x_2 \cdots h(u_n)h(u_1)\cdots h(u_n) = h(u_1)\cdots h(u_n)y$. We divide the proof into the following two cases:

Case 1 If $|h(u_1)| < |x_2 h(u_{i+2})|$, then $h(u_1)$ is a proper infix of $h(u_{i+1})h(u_{i+2})$. This is a contradiction.

Case 2 If $|h(u_1)|\geq|x_2h(u_{i+2})|$, then $h(u_1)$ has a proper infix $h(u_{i+2})$. This is a contradiction.

From all above, we can obtain that h preserves 1-comma-free codes. This completes the proof.

Theorem 3.3. Let $h: X^* \to X^*$ be a monomorphism. If $h(X)$ is a comma-free code, then h preserves 2-comma-free codes.

Proof . Suppose L is a 2-comma-free code, but $h(L)$ isn't a 2-comma-free code. There exist $u, v \in L$ such that $\{u, v\}$ is a comma-free code and $\{h(u), h(v)\}$ is not a comma-free code. There exists $x, y \in X^+$ such that one of the following cases holds: (1) $h(u)h(u) = xh(u)y$, (2) $h(u)h(v) = xh(u)y$, (3) $h(v)h(u) = xh(u)y$, or (4) $h(v)h(v) = xh(u)y$. By the theorem 3.2, case (1) doesn't hold. Without loss of generality, now we show case (2) doesn't hold. Let $u = u_1 \mathbf{L}\, u_n$, $v = u_1 \mathbf{L}\, v_m$ where $u_1, \mathbf{L}, u_n$, $v_1, \mathbf{L}, v_m \in X$. Hence we have $h(u_1)\,\mathbf{L}\, h(u_n)h(v_1)\,\mathbf{L}\, h(v_m) = xh(u_1)\,\mathbf{L}\, h(u_n)y$. Let $w_1 = u_1, \mathbf{L}, w_{m+n} = v_m$. Denote that $h(w_1)\,\mathbf{L}\, h(w_n)\,\mathbf{L}\, h(w_{m+n}) = xh(w_1)\,\mathbf{L}\, h(w_n)y$. Then we have the following two cases:

(i) If $x = h(w_1)\,\mathbf{L}\, h(w_i)$ for some $1 \leq i \leq m+n$, then $h(w_{i+1})\,\mathbf{L}\, h(w_{m+n}) = h(w_1)\,\mathbf{L}\, h(w_n)y$. If $1 \leq i < m$, then $h(w_1) = h(w_{i+1}), \mathbf{L}, h(w_n) = h(w_{i+n})$. Since h is injective, then $\{u, v\}$ is not a comma-free code, which contradicts L is a 2-comma-free code. If $i \geq m$, then $y = 1$. This is a contradiction.

(ii) If $h(w_1)\,\mathbf{L}\, h(w_{j-1}) <_p x <_p h(w_1)\,\mathbf{L}\, h(w_j)$ for some $1 \leq j \leq m+n$. If $h(w_1)\,\mathbf{L}\, h(w_{j+1}) <_p xh(w_1)$, then $h(w_{j+1})$ is an proper infix of $h(w_1)$. This is a contradiction. If $xh(w_1) <_p h(w_1)\,\mathbf{L}\, h(w_{j+1})$, then $h(w_1)$ is a proper infix of $h(w_j)h(w_{j+1})$. This is a contradiction.

From all above, we can obtain that h preserves 2-comma-free codes. This completes the proof.

Since $h(X)$ is a comma-free code if and only if h is comma-free codes preserving (see[2]), and L

is 3-comma-free code if and only if L is comma-free code by the definition, then we see the following fact: Let $h: X^* \to X^*$ be a monomorphism. For any $n \geq 1$, if $h(X)$ is a comma-free code, then h preserves n-comma-free codes.

Let $X = \{a, b\}$ and $k \geq 0$, then $L = \{a^{k+2}b^k, ab^k ab^k\}$ is a k-comma code and a comma-free code. Hence, for $k \geq 0$, a language which is a k-comma code and is a comma-free code exists indeed.

Proposition 3.4. Let $h: X^* \to X^*$ be a monomorphism. If $h(X)$ is a k-comma code and a comma-free code, then h preserves 1-k-comma codes for $k \geq 0$.

Proof. Suppose L is a 1-k-comma code, but $h(L)$ isn't a 1-k-comma code. Then there exists $u \in L$ such that $\{u\}$ is a k-comma code, but $\{h(u)\}$ is not a k-comma code. Then there exist $v \in X^k$ and $x, y \in X^+$ such that $h(u)vh(u) = xh(u)y$. Let $u = u_1 \cdots u_n$ for some $u_1, \cdots, u_n \in X$. Then $h(u_1) \cdots h(u_n)vh(u_1) \cdots h(u_n) = xh(u_1) \cdots h(u_n)y$. So one of the following cases holds: (1) $x = h(u_1) \cdots h(u_i)x_1$ where $0 \leq i \leq n$ and $x_1 \leq_p h(u_{i+1})$ or (2) $x = h(u_1) \cdots h(u_n)v_1$ where $v_1 <_p v$.

For the case (1), we consider the following cases.

(1-1) If $x_1 = 1$ or $x_1 = h(u_{i+1})$, we assume $x_1 = 1$. Then $h(u_{i+1}) \cdots h(u_n)vh(u_1) \cdots h(u_n) = h(u_1) \cdots h(u_n)y$. Hence $h(u_1) = h(u_{i+1}), \cdots, h(u_{n-i}) = h(u_n)$. This implies that $vh(u_1) \cdots h(u_n) = h(u_{n-i+1}) \cdots h(u_n)y$. If $|h(u_{n-i+1})| < |vh(u_1)|$, then $vh(u_1)$ has a proper infix $h(u_{n-i+1})$, which contradicts $h(X)$ is a k-comma code. If $|h(u_{n-i+1})| \geq |vh(u_1)|$, then $h(u_1)$ is a proper infix $h(u_{n-i+1})$. Since $h(X)$ is a k-comma code, then $h(X)$ is an infix code. This is a contradiction.

(1-2) If $x_1 \neq 1$ and $x_1 \neq h(u_{i+1})$, then $x_2 h(u_{i+2}) \cdots h(u_n)vh(u_1) \cdots h(u_n) = h(u_1) \cdots h(u_n)y$. This case can't hold just like in the proof of Theorem 3.1.

For the case (2), then $x = h(u_1)\cdots h(u_n)v_1$ for some $v_1 <_p v$, then $v_2 h(u_1)\cdots h(u_n) = h(u_1)\cdots h(u_n)y$, where $v_1 v_2 = v$. Since $|h(u_1)| < |v_2 h(u_1)|$, then $h(u_1)$ is a proper infix of $v_2 h(u_1)$, which contradicts $h(X)$ is a k-comma code.

From all above, we can obtain that h preserves 1-k-comma codes. This completes the proof.

Theorem 3.5. Let $h: X^* \to X^*$ be a monomorphism. If $h(X)$ is a k-comma code and a comma-free code, then h preserves 2-k-comma codes for $k \geq 0$.

Proof. Suppose L is a 2-k-comma code, but $h(L)$ is not a 2-k-comma code. Then there exist $u, v \in L$ such that $L_1 = \{u, v\}$ is a k-comma code, but $h(L_1)$ is not a k-comma code. That is, $h(L_1)X^k h(L_1) \cap X^+ h(L_1) X^+ \neq \varnothing$. Then there exist $w \in X^k$, $x, y \in X^+$ such that one of the following cases holds: (1) $h(u)wh(u) = xh(u)y$; (2) $h(u)wh(v) = xh(u)y$; (3) $h(v)wh(u) = xh(u)y$; (4) $h(v)wh(v) = xh(u)y$. By the proposition 3.4, case (1) does not hold. Without los of generality, case (3) and (4) are similar to case (2), now we only show case (2) doesn't hold. Let $u = u_1 \cdots u_n$, $v = u_1 \cdots v_m$ where $u_1, \cdots, u_n, v_1, \cdots, v_m \in X$. Then $h(u_1)\cdots h(u_n)wh(v_1)\cdots h(v_m) = xh(u_1)\cdots h(u_n)y$. This implies that one of the following cases holds: (i) $x = h(u_1)\cdots h(u_i)x_1$, where $x_1 \leq_p h(u_{i+1})$ and $0 \leq i \leq n$, (ii) $x = h(u_1)\cdots h(u_n)w_1$ where $w_1 <_p w$, or (iii) $x = h(u_1)\cdots h(u_n)wh(v_1)\cdots h(v_j)y_1$ where $y_1 \leq_p h(v_{j+1})$ and $0 \leq j \leq m-1$.

For the case (i), we consider the following two case.

(i-1) If $x_1 = 1$ or $x_1 = h(u_{i+1})$, without loss of generality, we assume $x_1 = 1$. Hence $h(u_{i+1})\cdots h(u_n)wh(v_1)\cdots h(v_m) = h(u_1)\cdots h(u_n)y$. This implies that $h(u_{i+1}) = h(u_1), \cdots,$ $h(u_n) = h(u_{n-i+1})$. So $wh(v_1)\cdots h(v_m) = h(u_{n-i+1})\cdots h(u_n)y$. If $|h(u_{n-i+1})| < |wh(v_1)|$, then $wh(v_1)$ has a proper infix $h(u_{n-i+1})$, which contradicts $h(X)$ is a

k-comma code. If $|h(u_{n-i+1})| \geq |wh(v_1)|$, then $h(v_1)$ is a proper infix of $h(u_{n-i+1})$, which contradicts $h(X)$ is an infix code.

(i-2) If $x_1 \neq 1$ and $x_1 \neq h(u_{i+1})$, then $x_2 h(u_{i+2}) \cdots h(u_n) w h(v_1) \cdots h(v_m) = h(u_1) \cdots h(u_n) y$.

Case 1 If $i = n-1$, then $x_2 w h(v_1) \cdots h(v_m) = h(u_1) \cdots h(u_n) y$. If $|h(u_1)| < |x_2 w h(v_1)|$, then $h(u_1)$ is a proper infix of $h(u_n) w h(v_1)$,which contradicts $h(X)$ is a k-comma code. If $|h(u_1)| \geq |x_2 w h(v_1)|$, then $h(u_1)$ has a proper infix of $h(v_1)$, which contradicts $h(X)$ is an infix code.

Case 2 If $0 \leq i \leq n-1$, then $x_2 h(u_{i+2}) \cdots h(u_n) w h(v_1) \cdots h(v_m) = h(u_1) \cdots h(u_n) y$. If $|h(u_1)| < |x_2 h(u_{i+2})|$, then $h(u_{i+1}) h(u_{i+2})$ has a proper infix $h(u_1)$, which contradicts $h(X)$ is a comma-free code. If $|h(u_1)| \geq |x_2 h(u_{i+2})|$ then $h(u_1)$ has a proper infix $h(u_{i+2})$, which contradicts $h(X)$ is an infix code.

For the case (ii), then $x = h(u_1) \cdots h(u_n) w_1$ for some $w_1 <_p w$, then $w_2 h(v_1) \cdots h(v_m) = h(u_1) \cdots h(u_n) y$ where $w_1 w_2 = w$. If $|h(u_1)| < |w_2 h(v_1)|$, then $w_2 h(v_1)$ has a proper infix $h(u_1)$, which contradicts $h(X)$ is a k-comma code. If $|h(u_1)| \geq |w_2 h(v_1)|$, then $h(v_1)$ is a proper infix of $h(u_1)$, which contradicts $h(X)$ is an infix code.

The case (iii) is similar to the case (i).

From all above, we can obtain h preserves 2-k-comma codes. This completes the proof.

Theorem 3.6. Let $h: X^* \to X^*$ be a monomorphism. If $h(X)$ is a k-comma intercode of index m and a comma-free code, then h preserves 2-k-comma intercodes of index m, where $k \geq 0$ and $m \geq 1$.

Proof. If $m = 1$, the proposition holds by Theorem 3.5. Assume $m \leq n$, the proposition holds. Let

$m=n+1$. Suppose L is a 2-k-comma intercode of index $n+1$, but $h(L)$ is not a 2-k-comma intercode of index $n+1$. Then there exist $u,v\in L$ such that $L_1=\{u,v\}$ is a k-comma intercode of index $n+1$, but $h(L_1)=\{h(u),h(v)\}$ is not a k-comma intercode of index $n+1$. That is $(h(L_1)X^k)^{n+1}h(L_1)\cap X^+(h(L_1)X^k)^n X^+\neq\varnothing$. This implies that one of the following two cases holds: (1) $h(u)u_1\ldots=xh(u)v_1\ldots y$; (2) $h(v)u_1\ldots=xh(u)v_1\ldots y$, where $x,y\in X^+$, $u_1,\ldots,v_1,\ldots\in X^k$.

For the case (1), since $|h(u)u_1|<|xh(u)v_1|$, then we cancel the initial $|h(u)u_1|$ letters of both sides from the equation. Thus $(h(L_1)X^k)^n h(L_1)\cap X^+(h(L_1)X^k)^{n-1}h(L_1)X^+\neq\varnothing$, which contradicts the fact that h preserves the k-comma intercode of index n.

For the case (2), If $|h(v)u_1|<|xh(u)v_1|$, then we cancel the initial $|h(v)u_1|$ letters of both sides from the equation. Thus $(h(L_1)X^k)^n h(L_1)\cap X^+(h(L_1)X^k)^{n-1}h(L_1)X^+\neq\varnothing$, which is a contradiction. If $|h(v)u_1|\geq|xh(u)v_1|$, then $h(v)u_1=xh(u)v_1w$ for some $w\in X^*$. Hence $h(v)u_1h(u)=xh(u)(v_1wh(u))$. This is a contradiction.

Thus h preserves 2-k-comma intercodes of index m. This completes the proof.

Proposition 3.7. Let h : $X^*\to X^*$ be a monomorphism. For any $m\geq 1$, If $h(X)$ is an intercode of index m, then h preserves 1-intercodes of index m.

Proof. When $m=1$, the proposition holds by the proposition 3.2. Suppose $m\leq n$, the proposition is true. Now let $m=n+1$, suppose L is a 1-intercode of index $n+1$, but $h(L)$ is not a 1-intercode of index $n+1$. Then there exists $u\in L$ such that $\{u\}$ is an intercode of index $n+1$, $\{h(u)\}$ is not an intercode of index $n+1$. Thus there exist $x,y\in X^+$ such that $(h(u))^{n+2}=x(h(u))^{n+1}y$. Since $|h(u)|<|xh(u)|$, then we cancel the initial $|h(u)|$ letters from both sides of the equation, thus

$\left(h\left(L\right)\right)^{n+1} \cap X^{+}\left(h\left(L\right)\right)^{n} X^{+}=\varnothing$. This is a contradiction. Thus h preserves 1-intercodes of index m. This completes the proof.

Proposition 3.8. Let h : $X^{*} \rightarrow X^{*}$ be a monomorphism. If $h\left(X\right)$ is an intercode of index m, then h preserves 2-intercodes of index m for any $m \geq 1$.

Proof. If $m=1$, it is proposition 3.3. Assume $m \leq n$, the result is true. Now let $m=n+1$. Suppose L is a 2-intercode of index $n+1$, but $h\left(L\right)$ is not a 2-intercode of index $n+1$. Then there exists $u, v \in L$ such that $L_{1}=\{u, v\}$ is an intercode of index $n+1$, but $h(L_{1})=\{h(u), h(v)\}$ isn't an intercode of index $n+1$. That is, $(h(L_{1}))^{n+2} \cap X^{+}(h(L_{1}))^{n+1} X^{+} \neq \varnothing$. Thus there exist $x, y \in X^{+}$ such that one of the following cases holds: (1) $h(u) \cdots = x h(u) \cdots y$, or (2) $h(u) \cdots = x h(v) \cdots y$.

For the case (1), since $|h(u)|<|x h(u)|$, then we cancel the initial $|h(u)|$ letters from both sides of the equation. So $(h(L_{1}))^{n+1} \cap X^{+}(h(L_{1}))^{n} X^{+} \neq \varnothing$, which is a contradiction.

For the case (2), we consider the following cases. (i) If $|h(u)|<|x h(v)|$, then we cancel the initial $|h(u)|$ letters from both sides of the equation. Thus $(h(L_{1}))^{n+1} \cap X^{+}(h(L_{1}))^{n} X^{+} \neq \varnothing$, which contradicts the assumption. (ii) If $|h(u)| \geq |x h(v)|$, then $h(u)=x h(v) w$ for some $w \in X^{*}$. Hence we have $h(u v)=x h(v)(w h(v))$. This is a contradiction.

Thus h preserves 2-intercodes of index m. This completes the proof.

4. Inverse homomorphism preserving

If $h(x)=1$ implies that $x=1$, then h is called 1-free homomorphism. Let $h: X^{*} \rightarrow X^{*}$ be a 1-free homomorphism, we denote $h^{-1}(L)=\{x \in X^{*} \mid h(x) \in L\}$ for all $L \subseteq X^{*}$.

Proposition 4.1. Let $h: X^{*} \rightarrow X^{*}$ be a 1-free homomorphism. If h preserves uniform codes, then h^{-1} preserves k-comma codes for $k \geq 0$.

Proof. Suppose $L \subseteq X^{+}$ is a k-comma code, but $h^{-1}(L)$ is not a k-comma code. Then there exists

$u,v,w\in h^{-1}(L)$ and $x,y\in X^{+},z\in X^{k}$ such that $uzv=xwy$. Since h preserves uniform codes, then $|h(z)|=k$. Since h is a 1-free homomorphism, then $h(u),h(v),h(w)\in L$ and $h(x),h(y)\neq 1$, which contradicts L is a k-comma code. This completes the proof.

Proposition 4.2. Let $h : X^{*}\to X^{*}$ be a 1-free homomorphism. If h preserves uniform codes, then h^{-1} preserves 1-k-comma codes for $k\geq 0$.

Proof. Suppose $L\subseteq X^{+}$ is a 1-k-comma code, but $h^{-1}(L)$ is not a 1-k-comma code. Then there exists $u\in L$ such that $\{u\}$ is a k-comma code, but $h^{-1}(u)$ is not a k-comma code. There exist $x,y\in X^{+},w\in X^{k}$ such that $h^{-1}(u)wh^{-1}(u)=xh^{-1}(u)y$. Since h is a 1-free homomorphism, then $uh(w)u=h(x)uh(y)\in X^{+}LX^{+}$. Since h preserves uniform codes, we have $h(w)\in X^{k}$, which contradicts that L is a 1-k-comma code. Thus h^{-1} preserves 1-k-comma codes for $k\geq 0$.This completes the proof.

Proposition 4.3. Let $h: X^{*}\to X^{*}$ be a 1-free homomorphism. If h preserves uniform codes, then h^{-1} preserves 2-k-comma codes for $k\geq 0$.

Proof. Suppose $L\subseteq X^{+}$ is a 2-k-comma code, but $h^{-1}(L)$ is not a 2-k-comma code. Then there exist $u,v\in L$ such that $L_1=\{u,v\}$ is a k-comma code, but $h^{-1}(L_1)$ is not a k-comma code . Then there exists $x,y\in X^{+},w\in X^{k}$ such that one of the following cases holds

(1) $h^{-1}(u)wh^{-1}(u)=xh^{-1}(u)y$; (2) $h^{-1}(u)wh^{-1}(v)=xh^{-1}(u)y$;

(3) $h^{-1}(v)wh^{-1}(u)=xh^{-1}(u)y$; (4) $h^{-1}(v)wh^{-1}(v)=xh^{-1}(u)y$.

By the Proposition 4.2, we know the case (1) doesn't hold. Without loss of generality, the case (3) and (4) are similar to the case (2),now we only show the case (2) doesn't hold. Since h is a 1-free homomorphism, then $uh(w)v=h(x)uh(y)\in X^{+}L_1X^{+}$. Since h preserves uniform codes, then $h(w)\in X^{k}$. This is a contradiction. We can also prove both the cases (3) and (4) don't hold by the similar way. Thus h^{-1} preserves 2-k-comma codes for $k\geq 0$. This completes the proof.

Proposition 4.4. Let $h: X^* \to X^*$ be a 1-free homomorphism. If h preserves uniform codes, then h^{-1} preserves n-k-comma codes for $k \geq 0$ and $n \geq 1$.

Proof. Since $L \subseteq X^+$ is a k-comma code if and only if L is a n-k-comma code for $n \geq 3$, then we have h^{-1} preserves n-k-comma codes for $k \geq 0$ and $n \geq 1$ by the Proposition 4.1, the Proposition 4.2 and the Proposition 4.3.Thus h^{-1} preserves n-k-comma codes for $k \geq 0$ and $n \geq 1$. This completes the proof.

Lemma 4.5. Let $h: X^* \to X^*$ be a mapping and $A, B \subseteq X^+$, then $h(h^{-1}(A)) \subseteq A$ and $h(A \cap B) \subseteq h(A) \cap h(B)$.

Proposition 4.6. Let $h: X^* \to X^*$ be a 1-free homomorphism. If h preserves uniform codes, then h^{-1} preserves n-k-comma intercodes of index m for $k \geq 0$ and $m, n \geq 1$.

Proof. We have proved that h^{-1} preserves k-comma codes and n-k-comma codes for any $k \geq 0$, $n \geq 1$ in the Proposition 4.4. Now we show that (1) If $k = 0$, $m, n \geq 1$, $(m,n) \neq (1,1)$, h^{-1} preserves n-intercode of index m. (2) If $k \geq 1$, $n \geq 1, m > 1$, h^{-1} preserves n-k-comma intercodes of index m.

For the case (1) ,Suppose $L \subseteq X^+$ is a n-intercode of index m, but $h^{-1}(L)$ is not an n-intercode of index m for some $m > 1$. Let $L_1 \subseteq L$ and $|L_1| = n$. Hence $L_1^{m+1} \cap X^+ L_1^m X^+ \neq \emptyset$, but $(h^{-1}(L_1))^{m+1} \cap X^+(h^{-1}(L_1))^m X^+ \neq \emptyset$. Since h is a 1-free, then

$$\emptyset \neq h[(h^{-1}(L_1))^{m+1} \cap X^+(h^{-1}(L_1))^m X^+] \subseteq h((h^{-1}(L_1))^{m+1}) \cap h(X^+(h^{-1}(L_1))^m X^+) \subseteq (L_1)^{m+1} \cap X^+(L_1)^m X^+.$$

Hence $(L_1)^{m+1} \cap X^+(L_1)^m X^+ \neq \emptyset$. This is a contradiction.

For the case (2), Suppose A is an n-k-comma intercode of index m and $h^{-1}(A)$ isn't an

n - k -comma intercode of index m for some $k,n \geq 1$ and $m \geq 1$. Let $A_1 \subseteq A$ and $|A_1| = n$. So $A_1^{m+1} \cap X^+ A_1^m X^+ = \varnothing$, but $(h^{-1}(A_1))^{m+1} \cap X^+(h^{-1}(A_1))^m X^+ \neq \varnothing$. Since h preserves uniform codes, then we have $|h(z)| = k$. Since h is a 1-free, then

$\varnothing \neq h((h^{-1}(A_1))^{m+1} \cap X^+(h^{-1}(A_1))^m X^+) \subseteq h((h^{-1}(A_1))^{m+1}) \cap h(X^+(h^{-1}(A_1))^m X^+) \subseteq (A_1)^{m+1} \cap X^+(A_1)^m X^+$. Hence $(A_1)^{m+1} \cap X^+(A_1)^m X^+ \neq \varnothing$. This is a contradiction.

Thus h^{-1} preserves n - k -comma intercodes of index m for $k \geq 0$ and $m,n \geq 1$. This completes the proof.

5. Conclusion

In this paper, we succeed to give some sufficient conditions for h and h^{-1} who preserve space codes, 1-comma-free codes, 2-comma-free codes, 1- k -comma codes, 2- k -comma codes, 2- k -comma intercodes, 1-intercodes and 2-intercodes. These results enriched the theory of these codes. Further work includes finding necessary and sufficient conditions for h and h^{-1} preserving these codes, experimental testing of, e.g., whether the language of genes of a certain organism is indeed a –space code for some k or not?

Acknowledgements: The authors would like to thank the anonymous referee for his/her careful reading of the manuscript and valuable suggestions.

References

[1]. Bo Cui, Lila Kari, Shinnosuke Seki(2011), K-comma codes and their Generaliazations. *Fundamenta Information*, 107(1): 1-18.

[2]. Chunhua Cao, Di Yang(2005), Notes on homomorphisms which preserve certain families of language, *Southeast Asian Bulletin of MathemaTics*, 29(3):415-422.

[3]. Shyr H.J., Free Monoids and Languages(2001), 3rd Edition, *Hon Min Book Company*, Taichung, Taiwan.

[4]. Shy H.J., Yu,S.S.(1990), Intercodes and related properties, Soochow J Math., 16(1):95-107.

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

[6]. Yu S.S.(1990), A characterization of intercodes, *Intern. J. Computer Math.*, 36(1-2), 39-45.

[7]. Ito M., Jurgensen H., Shyr H.J, Thierrin G.(1991), Outfix and infix codes and related classes of language, *Journal of Computer and System Sciences*, 43(3):484-508.

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

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