On the Reversibility of Parallel Insertion, and Its Relation to Comma Codes*

Bo Cui, Lila Kari, and Shinnosuke Seki

Department of Computer Science, University of Western Ontario,
London, Ontario, Canada, N6A 5B7
{bcui2,lila,sseki}@csd.uwo.ca

Abstract. This paper studies conditions under which the operation of parallel insertion can be reversed by parallel deletion, i.e., when does the equality $(L_1 \Leftarrow L_2) \Rightarrow L_2 = L_1$ hold for languages L_1 and L_2. We obtain a complete characterization of the solutions in the special case when both languages involved are singleton words. We also define *comma codes*, a family of codes with the property that, if L_2 is a comma code, then the above equation holds for any language $L_1 \subseteq \Sigma^*$. Lastly, we generalize the notion of comma codes to that of comma intercodes of index m. Besides several properties, we prove that the families of comma intercodes of index m form an infinite proper inclusion hierarchy, the first element which is a subset of the family of infix codes, and the last element of which is a subset of the family of bifix codes.

1 Introduction

In combinatorics on words and formal language theory, operations play an essential role in understanding the mechanisms of generating words and languages. Several generalizations of catenation and quotient, such as shuffle, shuffle on trajectories, [14], sequential and parallel insertion and deletion, [5], distributed catenation, [10], mix operation, [11], deletion on trajectories, [2], and hairpin completion and reduction, [13], have been studied in the literature. Follow-up studies investigated properties of languages produced by sequential and parallel insertion and deletion in [3,6,7,8,9]. One particular topic of interest was the reversibility of some of these operations, originally motivated by cryptography applications: If one uses the insertion of a key as one component of a cryptosystem to encrypt a plain-text message, and one step of decryption is accomplished by the deletion of the key, what are the language properties that would ensure that the plain-text can be uniquely deciphered? Motivated by this potential application, the determinism and inversibility of insertion and deletion operations on words were studied in, e.g., [6].

The question can be asked in a more general framework wherein the operations involved are the parallel insertion and deletion. This paper represents a first step

* This research was supported by Discovery Grant of the Natural Science and Engineering Research Council of Canada, and Canada Research Chair Award to L.K.

S. Bozapalidis and G. Rahonis (Eds.): CAI 2009, LNCS 5725, pp. 204–219, 2009.

towards an answer. More precisely, similar to sequential insertion and deletion, if we parallel-delete a word v from the language obtained by parallel-inserting v into u, we will not always obtain u. Thus, the question we ask is "Under what conditions, after parallel-inserting v into u, followed by the parallel deletion of v from the result, do we obtain exactly u?".

In Sect. 3, after the investigation of various properties of parallel insertion and deletion, we give a complete answer to this question for the singleton case, and furthermore we generalize the question to languages. We show that, if L_2 is a comma code (formally introduced in Sect. 4), any language L_1 can be recovered after first parallel-inserting L_2 into L_1 and then parallel-deleting L_2 from the result.

The notion of codes was defined for applications in communication systems. That is, if a message is encoded by using words from a code, then any arbitrary catenation of words should be uniquely decodable into code-words. Various codes with specific algebraic properties, such as prefix codes, infix codes, and comma-free codes [1,16,17], have been defined and used for various purposes. In Sect. 4, we define a family of codes, named *comma codes*, and show that this family is a proper subfamily of that of infix codes. Also, we give a characterization of comma codes, obtain some closure and algebraic properties, as well as compare the comma code family with other families, such as that of comma-free codes and that of solid codes.

Based on the similarity between the definition of comma codes and that of comma-free codes, in Sect. 5, we generalize comma codes and introduce the notion of *comma intercodes*. Similar to the notion of intercodes [16,17,18], the families of comma intercodes of index m form a proper inclusion hierarchy within the family of bifix codes. However, we show that any two families of intercodes and comma intercodes are incomparable.

2 Preliminaries

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, while 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^*. A language with exactly one word is called a *singleton*. In this paper, for a word $w \in \Sigma^*$, we often denote a singleton $\{w\}$ as w. A catenation of two languages $L_1, L_2 \subseteq \Sigma^*$, denoted by L_1L_2, is defined as $L_1L_2 = \{uv \mid u \in L_1, v \in L_2\}$. As mentioned, if an operand is a singleton, say $L_1 = \{u\}$ or $L_2 = \{v\}$, then we write uL_2 or L_1v instead of $\{u\}L_2$ or $L_1\{v\}$.

A word $x \in \Sigma^*$ is called an infix (prefix, suffix) of a word $u \in \Sigma^+$ if $u = zxy$ ($u = xy$, $u = zx$) for some words $y, z \in \Sigma^*$. In this definition, if z and y are nonempty, then such an x is called a *proper* infix, prefix, or suffix of u. For a word $u \in \Sigma^*$, the set of its infixes (prefixes, suffixes) is denoted by $\mathrm{F}(u)$ (resp. $\mathrm{Pref}(u)$, $\mathrm{Suff}(u)$). For a word $u \in \Sigma^*$, we denote the prefix (suffix) of length

$n \geq 0$ by $\mathrm{pref}_n(u)$ (resp. $\mathrm{suff}_n(u)$). These notations can be naturally extended to languages, e.g., $\mathrm{Pref}(L)$ is the set of prefixes of the words in L.

A nonempty word $u \in \Sigma^+$ is said to be *primitive* if $u = v^n$ implies $n = 1$ and $u = v$ for any $v \in \Sigma^+$. Any non-primitive word can be written as a power of a unique primitive word [16], which is called the *primitive root* of the word.

It is well known that [16], 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 called *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 $x, y \in \Sigma^+$ [16].

Parallel insertion and deletion on words and languages are variants of well-known (sequential) insertion and deletion, introduced in [5]. For two words $u, v \in \Sigma^*$, the *parallel insertion of v into u* results in a word $va_1va_2 \cdots a_nv$, where $u = a_1a_2 \cdots a_n$ for letters $a_1, \ldots, a_n \in \Sigma$. We denote this resulting word by $u \Leftarrow v$. This operation can be generalized to languages as follows: for two languages $L_1, L_2 \subseteq \Sigma^*$, the *parallel insertion of L_2 into L_1* generates a language

$$L_1 \Leftarrow L_2 = \bigcup_{n \geq 1,\, a_1, \ldots, a_n \in \Sigma \text{ s.t. } a_1a_2 \cdots a_n \in L_1} L_2a_1L_2a_2 \cdots L_2a_nL_2.$$

Example 1. For $L_1 = \{cd\}$ and $L_2 = \{a, b\}$,

$$\begin{aligned} L_1 \Leftarrow L_2 &= L_2cL_2dL_2 \\ &= \{acada, acadb, acbda, acbdb, bcada, bcadb, bcbda, bcbdb\}. \end{aligned}$$

In contrast, the parallel deletion of a language L_2 from another language L_1 results in a set of words which can be obtained by deleting elements of L_2 from an element of L_1 in a "maximal parallel manner". We denote the resulting set by $L_1 \Rightarrow L_2$. For $u \in L_1$, let

$$\begin{aligned} u \Rightarrow L_2 = \big\{ & u_1u_2 \cdots u_ku_{k+1} \mid u_1, \ldots, u_{k+1} \in \Sigma^*,\ k \geq 1,\ u \in u_1L_2u_2L_2 \cdots L_2u_{k+1} \\ & \text{and } \mathrm{F}(u_i) \cap (L_2 \setminus \{\lambda\}) = \emptyset \text{ for all } 1 \leq i \leq k+1 \big\}. \end{aligned}$$

By this definition, it is clear that if u does not contain any word in L_2 as its infix, then $u \Rightarrow L_2 = \emptyset$. Then we define $L_1 \Rightarrow L_2 = \bigcup_{u \in L_1}(u \Rightarrow L_2)$.

Example 2. Let $L_1 = \{abababa, aababa, abaabaaba\}$ and $L_2 = \{aba\}$. Then

$$\begin{aligned} L_1 \Rightarrow L_2 &= (\{abababa\} \Rightarrow L_2) \cup (\{aababa\} \Rightarrow L_2) \cup (\{abaabaaba\} \Rightarrow L_2) \\ &= \{b, abba\} \cup \{aba, aab\} \cup \{\lambda\} = \{b, abba, aba, aab, \lambda\}. \end{aligned}$$

3 When Does $(L_1 \Leftarrow L_2) \Rightarrow L_2$ Equal L_1?

By definitions, parallel insertion and deletion are not inverse operations in the sense that L_1 may not equal to $(L_1 \Leftarrow L_2) \Rightarrow L_2$. Thus, a question of interest is

to find under what conditions does the equality $(L_1 \Leftarrow L_2) \Rightarrow L_2 = L_1$ hold. We start by providing some properties of parallel insertions and deletions relevant to this question.

The simplest case is when the operation is the parallel insertion and both operands are singleton words. The next theorem will show that, unless w and u are unary words over the same letter, $w \Leftarrow u$ is primitive.

Lemma 1. *Let $u \in \Sigma^+$ and $u_s \in \mathrm{Suff}(u)$. If $u_s a u \in \mathrm{Pref}(u^2)$ for some $a \in \Sigma$, then u is a power of a.*

Proof. Due to the assumption, $u = u_s a u'_p = u'_p u_s a$ for some $u'_p \in \Sigma^*$. It well known that, for two words $u, v \in \Sigma^+$, if $uv = vu$, then they share their primitive roots. Therefore, the primitive root of u is same as that of $u_s a$. Hence, if u_s is empty, it is clear that $u \in a^+$. Even, otherwise, since $u_s \in \mathrm{Suff}(u'_p u_s a)$, u_s is a power of a. Thus, this lemma holds. □

Theorem 1. *Let $u, w \in \Sigma^+$. Then $w \Leftarrow u$ is not primitive if and only if w and u are unary words over the same letter $a \in \Sigma$.*

Proof. The if-direction is trivial. So we consider here the only-if direction under the assumption that $w \Leftarrow u$ is non-primitive. Then $w \Leftarrow u$ overlaps with its square in a nontrivial way. Let $w = a_1 a_2 \dots a_n$ for some $n \geq 1$ and $a_1, \dots, a_n \in \Sigma$. Also let $w \Leftarrow u = v^k$ for some $v \in \Sigma^+$ and $k \geq 2$. In the following, we prove that in all possible cases v is a unary word, which trivially implies what we want.

Firstly we consider the case when there is an integer ℓ such that $u a_1 \cdots u a_\ell = v^i$ for some $i \geq 1$, which further implies that $u a_1 \cdots u a_\ell = a_{n-\ell+1} u \cdots a_n u$. In this case, we can always find such ℓ in the range $\lceil n/2 \rceil \leq \ell$. For such ℓ, this equation implies that all of $a_1, \dots, a_n$ are the same, say a, and v is a power of a. If $|u| = 1$, this is always the case so that all we have to consider is the case $|u| \geq 2$ under the assumption that such ℓ cannot be found. Note that then we cannot find an integer $\ell' \geq 0$ such that $u a_1 \cdots a_{\ell'} u$ is a power of v, either.

Under the assumption, one of the occurrences of u in $w \Leftarrow u$ overlaps with the factor u^2 of $(w \Leftarrow u)^2$ nontrivially ($x \neq \lambda$ and $y \neq \lambda$ in Fig. 1.) As shown there, we have $u_s a_m u \in \mathrm{Pref}(u^2)$ for some $1 \leq m \leq n$. Lemma 1 implies that u is a unary word over a_m longer than 1. Note that the overlap between $w \Leftarrow u$ and its square implies that for all $1 \leq i \leq n$, $a_i = a_n$ because these characters in $w \Leftarrow u$ must be contained within some u in $(w \Leftarrow u)^2$. □

As mentioned before, $(L_1 \Leftarrow L_2) \Rightarrow L_2 = L_1$ is not always the case. Even if we limit L_1 and L_2 to be singletons $\{w\}$ and $\{u\}$, $(w \Leftarrow u) \Rightarrow u$ can contain

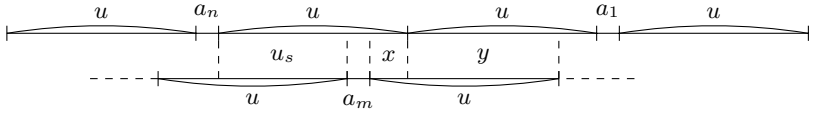

Fig. 1. How $u a_m u$ overlaps with $u a_n u^2$

words except w. Since parallel insertion of a word into another word certainly generates a singleton, it is the parallel deletion that creates such words. We initiate our investigation on this problem with a more general question: under what conditions, parallel deletion results in a singleton.

Note that $w \Rightarrow u = \emptyset$ if and only if w does not contain u as its infix. In the following, we only consider cases where w contains u as its infix. Also, note that two occurrences of u in w have to overlap in a nontrivial manner for $w \Rightarrow u$ not to be a singleton. If u is unbordered, two occurrences of u never overlap non-trivially regardless of what w is. Thus we have the following proposition.

Proposition 1. *If $u \in \Sigma^*$ is unbordered, then $w \Rightarrow u$ is a singleton for any word $w \in \Sigma^*$ that contains u as its infix.*

This also suggests that, even for a bordered word u, $w \Rightarrow u$ is at most a singleton as long as the form of w guarantees that nontrivial overlaps between u's do not occur in it. We will give a necessary and sufficient condition for $w \Rightarrow u$ to be a singleton in the case when w and u share the same primitive root.

Proposition 2. *For $a \in \Sigma$, let $w = a^j$ and $u = a^k$ for some $j \geq k \geq 1$. Then $w \Rightarrow u$ is a singleton if and only if either $k = 1$, $k \leq j \leq 2k - 1$, or $j = 3k - 1$.*

Proof. We consider the if-direction first. If $k = 1$, then this operation results in a singleton of the empty word. If $j < k$, then we cannot delete any u from w so that $w \Rightarrow u = \{w\}$. If $k \leq j \leq 2k - 1$, then by the definition of parallel deletion, the operation deletes exactly one u from w, and hence $w \Rightarrow u = \{a^{j-k}\}$. In the case when $j = 3k - 1$, we let $w = a^{i_1}a^k a^{i_2}$ for some $0 \leq i_1 < k$. Then $k \leq i_2 \leq 2k - 1$. We know that $a^{i_2} \Rightarrow u = \{a^{i_2-k}\}$. Hence $w \Rightarrow u$ is a singleton.

On the other hand, we show that if k and j do not satisfy these conditions, then $w \Rightarrow u$ contains at least two elements. If $2k \leq j \leq 3k - 2$, then it is clear that we can delete two u's from w. In addition, we can write w as $a^{k-1}a^k a^{j-2k-1}$. Since $j - 2k - 1 < k$, $a^{k-1}a^{j-2k-1}$ is also included in $w \Rightarrow u$. In the case $3k \leq j$, note that $(a^{2k} \Rightarrow u)(a^{j-2k} \Rightarrow u) \subseteq w \Rightarrow u$. We know that $(a^{2k} \Rightarrow u)$ is not a singleton, and hence $w \Rightarrow u$ cannot be a singleton. □

Since a primitive word cannot be a proper infix of its square [17], this proposition has the following corollary.

Corollary 1. *Let $w = g^j$ and $u = g^k$ for some primitive word g and $j \geq k \geq 1$. Then $w \Rightarrow u$ is a singleton if and only if either $k = 1$, $k \leq j \leq 2k$, or $j = 3k - 1$.*

Next we consider the more general case when w and u may have distinct primitive roots. If the primitive root of u is unbordered, then we can give a condition similar to the one given in Proposition 2. The proof for this proposition works to prove the next proposition.

Proposition 3. *Let $w \in \Sigma^*$ and $u = g^k$ for some unbordered primitive word g and $k \geq 1$. If the following condition holds, then $w \Rightarrow u$ is a singleton.*
(Condition) whenever $w = w_p g^j w_s$ for some $w_p, w_s \in \Sigma^*$ with $g \notin \mathrm{Suff}(w_p)$ and $g \notin \mathrm{Pref}(w_s)$, and $j \geq 1$, either $k = 1$, $k \leq j \leq 2k - 1$, or $j = 3k - 1$.

Now we consider the main problem of finding conditions for $(L_1 \Leftarrow L_2) \Rightarrow L_2$ to be equal to L_1. We start our investigation of this problem with the special case when $L_1 = \{w\}$ and $L_2 = \{u\}$. Hence our first aim is to clarify when $(w \Leftarrow u) \Rightarrow u$ does not contain any word other than w. If either w or u is the empty word, then $(w \Leftarrow u) \Rightarrow u$ is always $\{w\}$. Therefore in the remainder of this paper we will assume, without loss of generality, that u and w are nonempty. Let $w = a_1 a_2 \cdots a_n$ for some $n \geq 1$ and $a_1, \ldots, a_n \in \Sigma$. In order for the parallel deletion to create another word besides w, there must exist at least two different ways to parallel-delete the occurrences of u from $w \Leftarrow u$. In other words, we have to delete some occurrences of u that have not been parallel-inserted into w. Formally speaking, u has to be a proper infix of ua_iu for some $1 \leq i \leq n$. Based on this idea, we define the set:

$$X = \big\{u \in \Sigma^+ \mid \mathrm{pref}_x(u) \neq \mathrm{suff}_x(u) \text{ or } \mathrm{pref}_y(u) \neq \mathrm{suff}_y(u) \text{ for any } (x,y) \in N^2 \text{ with } x + y + 1 = |u|\big\}.$$

Informally, X contains words u which cannot be proper infixes of ubu for any letter $b \in \Sigma$. For such words $u \in X$, there cannot exist two different ways to parallel-delete the occurrences of u from $w \Leftarrow u$, and hence we have the following result.

Proposition 4. *If $u \in X$, then $(w \Leftarrow u) \Rightarrow u = \{w\}$ for any $w \in \Sigma^*$.*

In the following, we give a characterization of X. First of all, no unary word can be in X. By the informal definition of X, the set of all unbordered words of length at least 2, denoted by $U^{>1}$, is a subset of X. 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 ubu, for any $b \in \Sigma$.

Lemma 2. $N_{(>1)} \subseteq X$.

Proof. Suppose that there were $u \in N_{(>1)}$ such that $u \notin X$. 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 3. $Q_B^{(>1)} \subseteq X$.

Proof. Suppose that there exists $u \in Q_B^{(>1)}$ but $u \notin X$. 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 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 Sect. 2, 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. $\left(Q_B^{(=1)} \cup \{a^i \mid a \in \Sigma, i \geq 1\}\right) \cap X = \emptyset$.

Proof. As mentioned above, any unary word cannot be in X. 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 $w \notin X$. □

The next proposition 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.

Proposition 5. $X = U^{>1} \cup Q_B^{(>1)} \cup N_{(>1)}$.

Proof. Note that $\Sigma^+ = U^{>1} \cup Q_B \cup N_{(>1)} \cup \{a^i \mid a \in \Sigma, i \geq 1\}$. Combining Lemmas 2, 3, and 4 together, we can reach this proposition. □

As mentioned in Proposition 4, u being an element of X is sufficient for it to satisfy $(w \Leftarrow u) \Rightarrow u = \{w\}$ for any word w. In the following, we give necessary and sufficient conditions for the equality to be true in the case when $u \notin X$, that is, either u is unary or $u \in Q_B^{(=1)}$.

Proposition 6. *Let $w \in \Sigma^*$ and $u = a^k$ for some $a \in \Sigma$ and $k \geq 1$. Then $(w \Leftarrow u) \Rightarrow u = \{w\}$ if and only if*

1. *if* $k = 2$, *then* $aa \notin F(w)$;
2. *otherwise,* $w \in (\Sigma \setminus \{a\})^*$.

Proof. If w contains aa as its infix, then $a^{3k+2} \in \mathrm{F}(w \Leftarrow u)$. Proposition 3 implies that $(w \Leftarrow u) \Rightarrow u$ is not a singleton. Next we consider the case when w contains no aa but a as its infix, and $k = 2$. Then $a^5 \in \mathrm{F}(w \Leftarrow u)$. Since $5 = 3k - 1$, $(w \Leftarrow u) \Rightarrow u$ is a singleton due to the proposition. It is clear that for $w \in (\Sigma \setminus \{a\})^*$, $(w \Leftarrow u) \Rightarrow u = \{w\}$. □

Having considered the case of u being unary, now the only one remaining case is when u is an element of $Q_B^{(=1)}$. For such a word u, there exist $\alpha \in \Sigma^+$, $b \in \Sigma$, and $k \geq 1$ such that $u = (\alpha b)^k \alpha$. We define $M_u = \{a \in \Sigma \mid u \in \mathrm{Suff}(u)a\mathrm{Pref}(u)\}$. By definition, $M_u \neq \emptyset$ if and only if $u \notin X$.

Lemma 5. *For a bordered primitive word* u, *if* $b \in M_u$, *then there exists a nonempty word* $\alpha \in \Sigma^+$ *such that* $u = \alpha(b\alpha)^k$ *for some* $k \geq 1$ *and* αb *is primitive.*

Proof. Since $b \in M_u$, $u = u_p b u_s = u_s b u_p$ for some $u_p, u_s \in \Sigma^*$. Then $u_s b = (\alpha\beta)^i$ and $u_p = \alpha(\beta\alpha)^j$ for some $i \geq 1$, $j \geq 0$, and $\alpha, \beta \in \Sigma^*$ such that $\alpha\beta$ is primitive. Suppose that α were empty. Then $u = \beta^{i+j}$. On one hand, $i + j$ has to be 1 because u is primitive; on the other hand, $i + j \geq 2$ because u_p cannot be empty, otherwise, u is a unary word over b longer than 2. Thus, α is nonempty. So $ub = u_p b u_s b = \alpha(\beta\alpha)^{i+j} b$, and hence $b(\alpha\beta)^i = (\beta\alpha)^i b$. Since $\alpha\beta$ is primitive, β has to be of length 1, and hence $\beta = b$. □

Lemma 6. *For* $u \in Q_B^{(=1)}$, $|M_u| = 1$.

Proof. Suppose $|M_u| > 1$, say two distinct characters b, d are in M_u. Then Lemma 5 implies that $u = \alpha(b\alpha)^i = \gamma(d\gamma)^j$ for some $i, j > 0$ and $\alpha, \gamma \in \Sigma^*$ such that both αb and γd are primitive. Without loss of generality, we assume $|\alpha b| > |\gamma d|$. Then by Fine-and-Wilf's theorem [12], $i = 1$. Hence $u = \alpha b \alpha = \gamma(d\gamma)^j$. If j is odd, then clearly $b = d$, a contradiction. Otherwise, $\alpha = (\gamma d)^{j/2}\gamma_p = \gamma_s(d\gamma)^{j/2}$ and $\gamma = \gamma_p b \gamma_s$ for some $\gamma_p, \gamma_s \in \Sigma^*$ of same length. Then we have $(\gamma d)^{j/2-1}\gamma d\gamma_p = \gamma_s(d\gamma)^{j/2-1} d\gamma_p b\gamma_s$, and hence $b = d$, the same contradiction. □

Proposition 7. *Let* $u \in Q_B^{(=1)}$. *Then* $(w \Leftarrow u) \Rightarrow u = \{w\}$ *for* $w \in \Sigma^+$ *if and only if* $w \in (\Sigma \setminus M_u)^+$.

Proof. If w does not contain any letter in M_u, then it is clear that $(w \Leftarrow u) \Rightarrow u = \{w\}$.

We prove the converse implication. Due to Lemmas 5 and 6, $M_u = \{b\}$ and there exists $\alpha \in \Sigma^+$ such that $u = \alpha(b\alpha)^k$ for some $k \geq 1$ and αb is primitive. Let $w = a_1 \cdots a_n$ for some $n \geq 1$ and $a_i \in \Sigma$ for all $1 \leq i \leq n$, and assume that w contains b. Then we can find an integer $1 \leq m \leq n$ such that $a_{m-1} \neq b$ (if any), $a_m = \cdots = a_{m+j-2} = b$, and $a_{m+j-1} \neq b$ (if any) for some $j \geq 2$. Now

$$w \Leftarrow u = ua_1 \cdots ua_{m-1}[\alpha(b\alpha)^k b\alpha(b\alpha)^k b \cdots b\alpha(b\alpha)^k]a_{m+j-1}u \cdots a_n u.$$

We can parallel-delete u's from the bracketed infix in two ways: one is to delete j u's that were actually inserted by the preceding insertion; the other is to leave the first $\alpha\beta$ and delete u from every $(k+1)|\alpha\beta|$ position. Note that in the latter way, we delete exactly $j-1$ u's. If in both cases, we parallel-delete the inserted u's from the prefix and suffix, then we obtain two distinct words $w, a_1 \cdots a_{m-1}\alpha b b^{j-2}(b\alpha)^k a_{m+j-1} \cdots a_n$. We still need to check that the latter parallel deletion is valid. For that, it is enough to check that neither $a_{m-1}\alpha b$ or $(b\alpha)^k a_{m+j-1}$ contain u. Their lengths are at most $|u|$ so that if one of them contains u, then it is u itself. However, this is not the case because of the primitivity of αb and $\alpha \neq \lambda$. □

Since

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

Propositions 4, 5, 6, 7 completely characterize the solutions to the equation $(w \Leftarrow u) \Rightarrow u = \{w\}$.

Hence now we are ready to consider the more general equation $(L_1 \Leftarrow L_2) \Rightarrow L_2 = L_1$. When L_2 is a singleton, say $L_2 = \{u\}$, the set X plays an important role.

Proposition 8. *If $u \in X$, then $(L \Leftarrow u) \Rightarrow u = L$ for any language $L \subseteq \Sigma^*$.*

Proof. By definition, $(L \Leftarrow u) \Rightarrow u = \bigcup_{w \in L}(w \Leftarrow u) \Rightarrow u$. Then this result is immediate from Proposition 4. □

4 Comma Codes

In the previous section, we saw that if $u \in X$, then $(L \Leftarrow u) \Rightarrow u = L$ for any language $L \subseteq \Sigma^*$. The aim of this section is to introduce a new language family with the property that if a language L_2 belongs to this family, then $(L_1 \Leftarrow L_2) \Rightarrow L_2 = L_1$ holds for any language $L_1 \subseteq \Sigma^*$.

Definition 1. *A set $L \subseteq \Sigma^+$ is called a* comma code *if $L\Sigma L \cap \Sigma^+ L \Sigma^+ = \emptyset$.*

Intuitively, a comma code is a set L with the property that none of its words can be a proper infix of $u_1 a u_2$ where u_1 and u_2 are words in L, and $a \in \Sigma$ is a "comma". As it turns out (Corollary 2) a comma code is indeed a code.

As examples, $L = \{ab^k a \mid k > 1\}$ is a comma code, while any language that contains unary words or words in $Q_B^{(=1)}$ is not a comma code.

Theorem 2. *If the language $L_2 \subseteq \Sigma^+$ is a comma code, the equation $(L_1 \Leftarrow L_2) \Rightarrow L_2 = L_1$ holds for any language $L_1 \subseteq \Sigma^*$.*

The definition of comma codes reminds us of that of comma-free codes. A nonempty set $L \subseteq \Sigma^+$ is a *comma-free code* if $L^2 \cap \Sigma^+ L \Sigma^+ = \emptyset$. 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 [17]. We establish a relationship among these three codes, which leads us to the fact that comma codes are actually codes.

Lemma 7. *For a language $A \subseteq \Sigma^*$, A is a comma code if and only if $A\Sigma$ is a comma-free code.*

Proof

(If) We assume that $A\Sigma$ is a comma-free code, and suppose that A were not a comma code. Then there exist $w_1, w_2, w_3 \in A$, $a \in \Sigma$, and $x, y \in \Sigma^+$ such that $w_1 a w_2 = x w_3 y$. By putting some $b \in \Sigma$ at the ends of both sides, we can reach a contradiction with $A\Sigma$ being a comma-free code.

(Only-if) Suppose that $A\Sigma$ were not a comma-free code. Then we have $u_1 a_1 u_2 a_2 = x' u_3 a_3 y'$ for some $u_1, u_2, u_3 \in A$, $a_1, a_2, a_3 \in \Sigma$, and $x', y' \in \Sigma^+$. Since y' is nonempty, we can cut the rightmost letters of both sides from this equation, and reaches the contradiction. □

Lemma 8. *For a language $A \subseteq \Sigma^*$, A is an infix code if and only if $A\Sigma$ is an infix code.*

Proof. The only-if direction is trivial because the family of infix codes is closed under concatenation. As of the if direction, under the assumption that $A\Sigma$ is an infix code, suppose that A were not. Then there exist $u \in A$ and $x, y \in \Sigma^*$ such that $xuy \in A$ and $xy \neq \lambda$. Then for any $b \in \Sigma$, $xuyb \in A\Sigma$, which contains $uc \in A\Sigma$ as its factor, where c is a first letter of yb. Since $uc \neq xuyb$, this is a contradiction. □

Corollary 2. *A comma code is an infix code, and hence a code.*

Actually, the family of comma codes is a proper subset of the family of infix codes. For example, $L = \{ab, ba\}$ is an infix code, but not a comma code. Hence we give a characterization of infix codes which are comma codes. For this purpose, we define the following terms:

$$
\begin{aligned}
L_{\overline{p}} &= \{x \in \Sigma^+ \mid xy, yz \in L \text{ for some } y, z \in \Sigma^+\},\\
L_i &= \{y \in \Sigma^+ \mid xy, yz \in L \text{ for some } x, z \in \Sigma^+\},\\
L_{\overline{s}} &= \{z \in \Sigma^+ \mid xy, yz \in L \text{ for some } x, y \in \Sigma^+\},\\
L_{\overline{\overline{p}}} &= \{x \in \Sigma^+ \mid xa \in L_{\overline{p}} \text{ for some } a \in \Sigma\},\\
L_{\overline{\overline{s}}} &= \{x \in \Sigma^+ \mid ax \in L_{\overline{s}} \text{ for some } a \in \Sigma\}.
\end{aligned}
$$

Proposition 9 ([16]). *Let $L \subseteq \Sigma^+$. If L is an infix code, then the following four conditions are equivalent and make L a comma-free code: (1) $L_{\overline{s}} \cap L_i = \emptyset$, (2) $L_{\overline{p}} \cap L_i = \emptyset$, (3) $L \cap L_{\overline{s}} L_{\overline{s}} = \emptyset$, and (4) $L \cap L_{\overline{p}} L_{\overline{p}} = \emptyset$. Conversely, if L is a comma-free code, then L is an infix code with these properties.*

Proposition 10. *Let $L \subseteq \Sigma^+$. If L is an infix code such that $L \cap \Sigma = \emptyset$ and $(L_{\overline{s}} \cup L_{\overline{p}}) \cap \Sigma = \emptyset$, then the following four conditions are equivalent and make L a comma code: (1) $L_{\overline{\overline{s}}} \cap L_i = \emptyset$, (2) $L_{\overline{\overline{p}}} \cap L_i = \emptyset$, (3) $L \cap L_{\overline{\overline{s}}} L_{\overline{s}} = \emptyset$, and (4) $L \cap L_{\overline{p}} L_{\overline{\overline{p}}} = \emptyset$. Conversely, if L is a comma code, then L is an infix code with these properties.*

Proof. Note that the emptiness of $L \cap \Sigma$ and $(L_{\overline{s}} \cup L_{\overline{p}}) \cap \Sigma$ is the minimal requirement for L to be a comma code.

(Only-if) Lemma 7 implies that $L\Sigma$ and ΣL are comma-free codes. Using Proposition 9, we have the four properties: (a) $(L\Sigma)_{\overline{s}} \cap (L\Sigma)_i = \emptyset$, (b) $(\Sigma L)_{\overline{p}} \cap (\Sigma L)_i = \emptyset$, (c) $L\Sigma \cap (L\Sigma)_{\overline{s}}(L\Sigma)_{\overline{s}} = \emptyset$, and (d) $\Sigma L \cap (\Sigma L)_{\overline{p}}(\Sigma L)_{\overline{p}} = \emptyset$. Suppose that there were $u \in L_{\overline{\overline{s}}} \cap L_i$. Then there exist $x, y, z, w \in \Sigma^+$ and $a \in \Sigma$ such that $xy, yau, zu, uw \in L$. Let $w = bw'$ for some $w' \in \Sigma^*$. Then $xya, yaub \in L\Sigma$ and hence $ub \in (L\Sigma)_{\overline{s}}$. Moreover, $zub, ubw'c \in L\Sigma$ for any $c \in \Sigma$, and hence $ub \in (L\Sigma)_i$. These two results cause a contradiction with the property (a). The 2nd one derives from the property (b) in the same manner. Next we prove the 3rd property from (c). Suppose that $L \cap L_{\overline{\overline{s}}}L_{\overline{s}} \neq \emptyset$. Then there exist $x, y, z, w, u, v \in \Sigma^+$ and $a \in \Sigma$ such that $xy, yau, zw, wv \in L$ and $uv \in L$. Let $v = bv'$ for some $v' \in \Sigma^*$. Then $xya, yaub, zwb, wbv'c \in L$ for any $c \in \Sigma$. Thus, $ub, v'c \in (L\Sigma)_{\overline{s}}$ and $ubv'c \in L\Sigma$, a contradiction. The 4th derives from the property (d) in this way.

(If) Suppose L were not a comma code. Then there exist $u, v, w \in L$, $x, y \in \Sigma^+$, and $a \in \Sigma$ such that $uav = xwy$. Since $L \cap \Sigma = \emptyset$, $(L_{\overline{s}} \cup L_{\overline{p}}) \cap \Sigma = \emptyset$, and L is an infix code, $u = x\alpha$, $v = \beta y$, and $w = \alpha a \beta$ for some $\alpha, \beta \in \Sigma^+$. Therefore, $\beta \in L_{\overline{\overline{s}}} \cap L_i$, $\alpha \in L_{\overline{\overline{p}}} \cap L_i$, $\beta y \in L \cap \in L_{\overline{\overline{s}}}L_{\overline{s}}$, and $x\alpha \in L \cap \in L_{\overline{p}}L_{\overline{\overline{p}}}$. These contradict the properties 1-4. □

Example 3. Let $L_1 = \{aba, abba\}$. While this is a comma-free code, $abababa \in L\Sigma L \cap \Sigma^+ L \Sigma^+$ and hence L_1 is not a comma code. On the other hand, let us consider $L_2 = \{aaab, abab\}$. This is a comma code but not a comma-free code because any element of comma-free codes has to be primitive [17]. Moreover, there is a language which is both a comma and comma-free code. An example is $L_3 = \{abba, abbba\}$.

This example is enough to verify the following result.

Proposition 11. *The family of comma codes and the family of comma-free codes are incomparable, but not disjoint.*

Another important subfamily of infix codes is the family of solid codes. A nonempty set $L \subseteq \Sigma^+$ is called a *solid code* if L is an infix code and $\mathrm{Pref}(L) \cap \mathrm{Suff}(L) \cap \Sigma^+ = \emptyset$. This is a strict requirement. In fact, if L is a solid code, then all of L_i, $L_{\overline{s}}$, $L_{\overline{p}}$, $L_{\overline{\overline{s}}}$, and $L_{\overline{\overline{p}}}$ are empty. Thus, the following is a corollary of Proposition 10.

Corollary 3. *Let L be a solid code. If $L \cap \Sigma = \emptyset$, then L is a comma code.*

Since there exists a solid code all of whose elements are of length at least 2, this corollary clarifies that the family of solid codes and that of comma codes are not disjoint. However, these two families are incomparable as shown in the next example.

Example 4. Let $L_1 = \{ab, c\}$. This is a solid code, but not a comma code because it contains a word of length 1. On the other hand, L_2 in Example 3 provides an example of a comma code which is not a solid code.

Proposition 12. *The family of comma codes and the family of solid codes are incomparable.*

Next we consider the closure properties of comma codes under certain operations. For alphabets Σ_1, Σ_2, let $f : \Sigma_1^* \to \Sigma_2^*$ be a homomorphism. Then the inverse homomorphism $f^{-1} : \Sigma_2^* \to 2^{\Sigma_1^*}$ is defined as: for $u \in \Sigma_2^*$, $f^{-1}(u) = \{v \in \Sigma_1^* \mid f(v) = u\}$.

Proposition 13. *The family of comma codes is not closed under union, catenation, +, complement, non-erasing homomorphism, and inverse non-erasing homomorphism. On the contrary, it is closed under reversal and intersection with an arbitrary set.*

Proof. The union of comma codes $\{ab\}$ and $\{ba\}$ is not a comma code. The catenation AB of comma codes $A = \{aaba\}$ and $B = \{abaa\}$ is not so because $(aaba)(abaa)b(aaba)(abaa)$ contains $(aaba)(abaa)$ as a proper infix. For a comma code $L = \{abab\}$, $abababaabab \in L^+\Sigma L^+ \cap \Sigma^+ L^+ \Sigma^+$. Thus L^+ is not a comma code. The complement of a comma code $\{ab\}$ contains a word of length 1 and hence not a comma 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 comma code $\{aaab, abab\}$ onto $\{aaaa\}$, which is not a comma code. Consider alphabets $\Sigma_3 = \{a\}$ and $\Sigma_4 = \{a, b\}$, and let $g : \Sigma_3^* \to \Sigma_4^*$ be a homomorphism defined as $g(a) = ab$. Since $L = \{abab\}$ is a comma code but $g^{-1}(L) = \{aa\}$ is not, the class of comma codes is not closed under inverse non-erasing homomorphisms.

By definition, it is clear that the family of comma codes is closed under reversal or intersection with an arbitrary set. □

Proposition 13 says that the catenation of two comma codes is not always a comma code. So we investigate a condition under which a catenation of two languages A and B becomes a comma code under the assumption that $A \cup B$ is an infix code. Under this assumption, an element of AB can be a proper infix of an element of $AB\Sigma AB$ only in two ways as shown in Fig. 2. The following results offer additional conditions on A and B, which make AB a comma code by preventing both cases in Fig. 2 from occurring.

Proposition 14. *Let $A, B \subseteq \Sigma^*$ such that $A \cup B \neq \emptyset$. If $A \cup B$ is either a comma code or a comma-free code, then AB is a comma code.*

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

Proposition 15. *Let $A, B \subseteq \Sigma^*$ such that $A \cap B = \emptyset$ and $A \cup B$ is an infix code. If $A_{\overline{s}} \cap B_{\overline{p}} = \emptyset$, then AB is a comma code.*

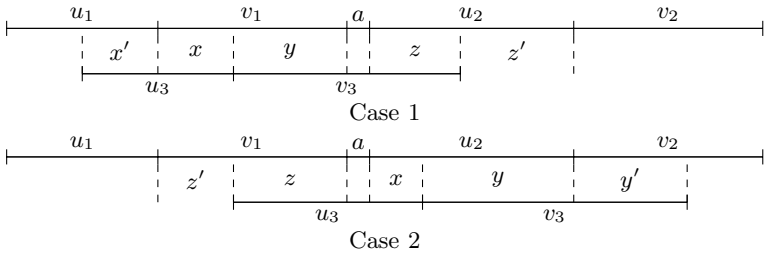

Fig. 2. For $u_1, u_2, u_3 \in A$ and $v_1, v_2, v_3 \in B$, if $A \cup B$ is an infix code, u_3v_3 can be a proper infix of $u_1v_1au_2v_2$ only in these two ways. Note that x' and y in Case 1 can be empty at the same time, and x and y' in Case 2 can be empty at the same time.

Proof. Suppose that AB were not a comma code. Then there exist $u_1, u_2, u_3 \in A$, $v_1, v_2, v_3 \in B$, and $a \in \Sigma$ such that $u_1v_1au_2v_2 = ru_3v_3s$ for some $r, s \in \Sigma^+$. Since $A \cup B$ is an infix code and $A \cap B = \emptyset$, we have only two cases: (1) $u_3 = x'x$, $v_1 = xy$, $v_3 = yaz$, and $u_2 = zz'$, or (2) $v_1 = z'z$, $u_3 = zax$, $u_2 = xy$, and $v_3 = yy'$ for some $x', x, y, z \in \Sigma^+$ and $a \in \Sigma$. Then x in case (1) or y in case (2) is in $A_{\overline{s}} \cap B_{\overline{p}}$, a contradiction. □

Note that the condition in the above proposition is also the condition for AB to be a comma-free code [16]. Therefore, if A and B are two disjoint languages such that $A \cup B$ is an infix code and $A_{\overline{s}} \cap B_{\overline{p}} = \emptyset$, then AB is in the intersection of the family of comma codes and that of comma-free codes.

5 Comma Intercodes

In coding theory, the notion of comma-free code was extended to the more general one of intercode. For $m \geq 1$, a nonempty set $L \subseteq \Sigma^+$ is called an *intercode of index* m if $L^{m+1} \cap \Sigma^+L^m\Sigma^+ = \emptyset$. An intercode of index 1 is a comma-free code. Based on the similarity between the definition of comma code and that of comma-free code, we introduce the comma intercode as a generalization of comma code.

For $m \geq 1$, a nonempty set $L \subseteq \Sigma^+$ is called a *comma intercode of index* m if $(L\Sigma)^mL \cap \Sigma^+(L\Sigma)^{m-1}L\Sigma^+ = \emptyset$. It is immediate that a comma intercode of index 1 is a comma code. A language L is called a *comma intercode* if there exists an integer $m \geq 1$ such that L is a comma intercode of index m. First of all, we have to prove that a comma intercode is actually a code. 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 16. *A comma intercode is a bifix code.*

Proof. Let L be a comma intercode of index m for some $m \geq 1$. 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 $a_1, \ldots, a_m \in \Sigma$, $wa_1wa_2 \cdots a_mw = wa_1(wa_2 \cdots a_mu)v \in \Sigma^+(L\Sigma)^{m-1}L\Sigma^+$, which contradicts that L is a comma intercode. In the same way, we can prove that L must be a suffix code. Thus, L is a bifix code. □

Like comma codes, a comma intercode consists of only non-unary words of length at least 2. From now, we introduce several properties of comma intercodes.

Proposition 17. *Let L be a regular language. Then for a given integer $m \geq 1$, it is decidable whether or not L is a comma intercode of index m.*

Proof. Since the family of regular languages is closed under catenation and intersection, $(L\Sigma)^m L \cap \Sigma^+(L\Sigma)^{m-1}L\Sigma^+$ is regular. Hence it is decidable whether this language is empty. □

Proposition 18. *Let L be a comma intercode of index m for some $m \geq 1$. Then $L \subseteq X$.*

Proof. Suppose that there were $w \in L$ but $w \notin X$. Then $w = w_s a w_p$ for some $w_s \in \mathrm{Suff}(w)$, $a \in \Sigma$, and $w_p \in \mathrm{Pref}(w)$. This implies that $w = w_p a w_s$. Then $(wa)^m w = w_p a (w_s a w_p a)^{m-1} w_s a w_p a w_s \in \Sigma^+(L\Sigma)^{m-1}L\Sigma^+$, a contradiction. □

Proposition 19. *For any $m \geq 1$, every comma intercode of index m is a comma intercode of index $m+1$.*

Proof. Let L be a comma intercode of index m. By definition, we have $(L\Sigma)^m L \cap \Sigma^+(L\Sigma)^{m-1}L\Sigma^+ = \emptyset$. Suppose that L were not a comma code of index $m+1$. Then $(L\Sigma)^{m+1}L \cap \Sigma^+(L\Sigma)^m L\Sigma^+ \neq \emptyset$. That is, there exist $x_1, \ldots, x_{m+2} \in L$, $y_1, \ldots, y_{m+1} \in L$, $a_1, \ldots, a_{m+1}, b_1, \ldots, b_m \in \Sigma$, and $u, v \in \Sigma^+$ such that $x_1 a_1 \cdots a_{m+1} x_{m+2} = u y_1 b \cdots b_m y_{m+1} v$. Because of L being a comma intercode of index m, $|u| < |x_1|$ and $|v| < |x_{m+2}|$ must hold. However, even so, $y_1 b_1 \cdots b_m y_{m+1}$ is in $\Sigma^+ x_2 a_2 \cdots a_m x_{m+1} \Sigma^+$, and hence $(L\Sigma)^m L \cap \Sigma^+(L\Sigma)^{m-1} L\Sigma^+ \neq \emptyset$. This is a contradiction. □

For any $m \geq 1$, we denote the family of comma intercodes of index m by I_m. Proposition 19 implies that $I_m \subseteq I_{m+1}$ for any $m \geq 1$. This inclusion is actually proper. Let $\{a, b\} \subseteq \Sigma$ and $u_i = ab^i a$ for some $i \geq 1$. Then, for some $a_1, \ldots, a_{m+1} \in \Sigma$, $L = \{u_1 a_1 \cdots u_{m+1} a_{m+1} u_{m+2}, u_2, u_3, \ldots, u_m, u_{m+1}\}$ satisfies the condition $(L\Sigma)^{m+1}L \cap \Sigma^+(L\Sigma)^m L\Sigma^+ = \emptyset$, and hence $L \in I_{m+1}$. On the other hand, $L \notin I_m$. This is because a word $u_1 a_1 \cdots u_{m+1} a_{m+1} u_{m+2} \in \Sigma^+ u_2 a_2 \cdots u_{m+1} \Sigma^+$.

Moreover, let C_b denote the family of bifix codes. Then $\{aba, abba\}$ is in C_b but not in I_m for any $m \geq 1$. Combining Proposition 19 with this example, we have the following hierarchy, where $\subset$ denotes proper inclusion.

Theorem 3. *$I_1 \subset I_2 \subset \cdots \subset I_m \subset \cdots \subset C_b$ holds.*

Let I'_m denote the family of intercodes of index m for any $m \geq 1$. It is known that $I'_1 \subset I'_2 \subset \cdots \subset I'_m \subset \cdots \subset C_b$ holds [16]. Due to these results and Proposition 11, we obtain the following corollary.

Corollary 4. *For any $m, n \geq 1$, the family of intercodes of index m and the family of comma intercode of index n are incomparable.*

Furthermore, we know that the family of comma-free codes and that of comma codes are proper subsets of the family of infix codes. Thus, we can draw the proper inclusion hierarchy of the families of bifix codes, intercodes, comma intercodes, and infix codes as follows.

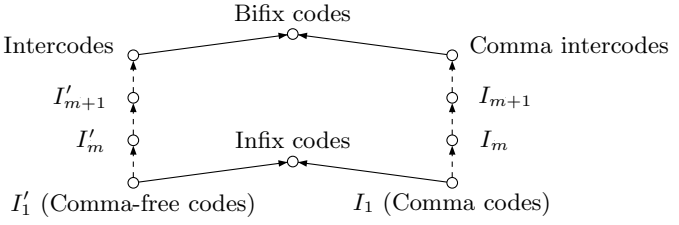

Fig. 3. The inclusion hierarchy of bifix codes, intercodes, comma intercodes, and infix codes, where arrows indicate proper inclusion

Although the definition and some properties of comma intercodes are similar with those of intercodes, we show in the following that these two codes are not similar 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 [17]. In contrast, comma intercodes do not have such a property.

Proposition 20. *Let $L \subseteq \Sigma^+$ be a comma intercode of index n. Then L is not necessarily synchronously decipherable with delay less than or equal to n.*

Proof. Consider $L = \{abab, aaab\}$, which is a comma intercode of index 1, and hence a comma code of any index. For $m \geq 1$, $aaab(abab)^m = aa(abab)^m ab \in L^{m+1}$ and $(abab)^m \in L^m$ but $aa, ab \notin L$. Therefore, L is not with delay m. □

6 Conclusion

In this paper, we obtained some properties of parallel insertion and deletion, and investigated conditions for the equation $(L_1 \Leftarrow L_2) \Rightarrow L_2 = L_1$ to hold. We obtained a complete characterization of solutions in the special case when L_1 and L_2 are singleton languages. For the general case, we introduced the definition of comma codes and proved that, if L_2 is a comma code, then the equation holds for any language $L_1 \subseteq \Sigma^*$. We also obtained a characterization, some closure properties, and algebraic properties of comma codes, and compared this family of codes with the families of comma-free codes and solid codes. Lastly, we generalized the notion of comma codes to that of comma intercodes of index m. As it turns out, the families of comma intercodes of index m form an infinite proper inclusion hierarchy within the family of bifix codes. The first element

of this hierarchy, the family of comma codes, is a subset of the family of infix codes, while the last element of which is a subset of the family of bifix codes. This hierarchy parallels, but is different from, the one that starts with comma-free codes (which are infix codes), and continues with intercodes of index m (which are bifix codes).

Acknowledgement

The authors acknowledge the anonymous referees for their useful comments.

References

1. Berstel, J., Perrin, D.: Theory of Codes. Academic Press. Inc., Orlando (1985)
2. Domaratzki, M.: Deletion along trajectories. Theoretical Computer Science 320, 293–313 (2004)
3. Ito, M., Kari, L., Thierrin, G.: Insertion and deletion closure of languages. Theoretical Computer Science 183, 3–19 (1997)
4. Jürgensen, H., Konstantinidis, S.: The hierarchy of codes. In: Ésik, Z. (ed.) FCT 1993. LNCS, vol. 710, pp. 50–68. Springer, Heidelberg (1993)
5. Kari, L.: On Insertion and Deletion in Formal Languages. Ph.D. Thesis, University of Turku (1991)
6. Kari, L.: Insertion and deletion of words: determinism and reversibility. In: Havel, I.M., Koubek, V. (eds.) MFCS 1992. LNCS, vol. 629, pp. 315–326. Springer, Heidelberg (1992)
7. Kari, L., Thierrin, G.: Words insertions and primitivity. Utilitas Mathematica 53, 49–61 (1998)
8. Kari, L., Mateescu, A., Paun, G., Salomaa, A.: Deletion sets. Fundamenta Informatica 18(1), 355–370 (1993)
9. Kari, L., Mateescu, A., Paun, G., Salomaa, A.: On parallel deletions applied to a word. RAIRO. Theoret. Inform. Appl. 29, 129–144 (1995)
10. Kudlek, M., Mateescu, A.: On distributed catenation. Theoretical Computer Science 180, 341–352 (1997)
11. Kudlek, M., Mateescu, A.: On mix operation. New Trends in Formal Languages. In: Păun, G., Salomaa, A. (eds.) New Trends in Formal Languages. LNCS, vol. 1218, pp. 35–44. Springer, Heidelberg (1997)
12. Lothaire, M.: Algebraic Combinatorics on Words. Cambridge University Press, Cambridge (2002)
13. Manea, F., Mitrana, V., Yokomori, T.: Two complementary operations inspired by the DNA hairpin formation: Completion and reduction. Theoretical Computer Science 410, 417–425 (2009)
14. Mateescu, A., Rozenberg, G., Salomaa, A.: Shuffle on trajectories: syntactic constraints. Theoretical Computer Science 197, 1–56 (1998)
15. Parikh, R.J.: On context-free languages. Journal of the Association for Computing Machinery 13, 570–581 (1966)
16. Shyr, H.J.: Free Monoids and Languages. Lecture Notes, Institute of Applied Mathematics, National Chung-Hsing University, Taichung, Taiwan (2001)
17. Yu, S.S.: Languages and Codes. Lecture Notes, Department of Computer Science, National Chung-Hsing University, Taichung, Taiwan 402 (2005)
18. Yu, S.S.: A characterization of intercodes. Intern. J. Computer Math. 36, 39–48 (1990)