*Vietnam Journal of Mathematics* **31**:1 (2003) 57-69

Vietnam Journal
of
MATHEMATICS


# Completing Solid Codes to Maximal Comma-Free Codes

**Nguyen Huong Lam**

*Hanoi Institute of Mathematics*
*P.O. Box 631, BoHo, Hanoi, Vietnam*

Received July 17, 2001
Revised November 20, 2001

**Abstract.** We give a partial answer to the following problem: Is every finite comma-free code is completed to a finite maximal comma-free code? Namely, we show that every finite solid code could be explicitly embedded in a finite maximal comma-free code. The construction relies on an investigation into some structural aspects of maximal solid codes that has an independent interest.

## 1. Introduction

Comma-free code is an object of fairly long history dating back to the early 1950s, in the framework of the fundamental discovery in biochemistry of that time, the problem of DNA structure and its replication in living organisms. In this respect we refer the reader to [2, 7, 19]. Despite their origin, comma-free codes are mathematical objects with their rigorous definition, which is inspired by and generalized from the biological considerations. They form an amazing subclass of codes and had been studied in detail by numerous authors around 1960. We can indicate some references such as [4, 5, 7, 8], in which extensive computations towards the size of comma-free codes were carried out, or their constructions in alphabets of particular cardinality (or even, classification when possible), were suggested.

In this paper, we investigate the comma-free code from the point of view of the theory of codes [1], namely, the problem of completing a given code in a particular class to a maximal code in this class.

It is a conpiscuous fact that every code is contained in a maximal code, but is it still valid when we restrict ourselves to a specific subclass of codes? It had been shown that every regular code is completed to a regular maximal code [3],

but for the finite codes the claim does not hold. For example, the following code $\{a^5, ab, ba^2, b\}$ is not contained in any maximal code which is still finite [14, 16]. Neither does it for the class of finite bifix codes: the code $\{a^m, b^n\}$ with $m \neq n$ is never contained in a finite maximal bifix code. As for positive answer, every prefix code $X$ has, so to speak, an explicit completion, that is, maximal prefix code containing it, $Y = R \setminus RA^+$, where $R = A^* \setminus (P \cup XA^*)$, $A$ is the underlying alphabet and $P$ is the set of proper prefixes of $X$. From this expression, we conclude at once that the completion $Y$ is regular whenever $X$ is regular, and finite, when $X$ is finite. This also holds for suffix codes by the symmetrical reasoning.

In the same spirit, we can complete every *solid* code $X$ in a way as simple as that: the code $Y = Q \setminus A^+Q$, where $Q = P^*A \setminus (A^*P \cup F(X))$ and $F(X)$ is the set of proper factors of $X$. As above, this construction shows that every regular, or more than that, finite, solid code always has regular, or finite, respectively, completion [12].

Now turning to comma-free codes, what has been done about the completion problem? It has been recently proved that every regular comma-free code has regular completion, and what is important, is that the completion can be effectively determined from the original code [13]. Hereafter, in this article, we try to solve the completion problem for finite comma-free codes; more precisely, we do this for an important, more restricted subclass of the finite solid codes: every finite solid code will be shown to have a finite comma-free completion; in other words, considered as a comma-free code, a finite solid code is contained in a finite maximal comma-free code. The completing procedure now requires a further investigation on finite maximal solid codes, which sheds additional light on their structure. We should comment that solid codes constitute an important subject of study in coding theory by their remarkable error-correcting properties and behaviour in a broader class of transmission channels with a bunch of problems that faces the information communication of today [9].

## 2. Notions, Notations and Preliminary Results

To help with the statements of our main results, we discuss the notation and the background here. Let $A$ be a finite alphabet, $A^*$ the set of words, equipped with catenation product, and $A^+$ the set of non-empty words on $A$. In linguistic contexts, we use 1 to denote the empty word, of course, if there is no danger of confusing with its arithmetic meaning. For any word $w$, denote by $|w|$ the length of $w$ and we set by convention $|1| = 0$; also we denote respectively by $P(w), S(w)$ and $F(w)$ the set of proper (that is of length less than $|w|$) prefixes, suffixes and factors of $w$ and, moreover these notations are extended for subsets $X$ of $A^*$ in a natural way: $P(X), S(X)$ and $F(X)$.

We define now the notion of occurence of a factor. Let $w$ be a word and $u$ a factor of $w$. This means that there exist some words $x$ and $y$ such that $w = xuy$. As a matter of fact $u$ may occur in different positions within $w$ and we use the triple $(x, u, y)$ to specify this particular position and call it an *occurence* of $u$ in $w = xuy$. Thus a prefix $p$ is a factor which has an occurence $(1, p, y)$ such that

$w = py$ and a suffix $s$ has an occurence $(x, s, 1)$ with $w = xs$.

Let $u, v$ be arbitrary words. The *left quotient* $u^{-1}v$ of $v$ by $u$ is the word $x$, if it exists, such that $v = ux$ and similarly, the *right quotient* $vu^{-1}$ of $v$ by $u$ is the word $y$ such that $v = yu$.

Now suppose that $(x_1, u_1, y_1)$ and $(x_2, u_2, y_2)$ are some occurences of $u_1$ and $u_2$ in $w$ with $|x_1| < |x_2|$ (that means $u_1$ preceeds $u_2$, or $u_2$ succeeds $u_1$ in these two particular positions within $w$). We call the factor $x_1^{-1}x_2u_2$, or just the same, $u_1y_1y_2^{-1}$, the *spanning factor* of the two occurences above. This formal definition is just a verbal transcription of the more lucid graphical presentation as the term suggests; the reader may make a drawing for easier perception at this point and in the similar situations encountered in the sequel.

The catenation product defined for words extends in a natural way for subsets of words. Let $X, Y$ be two subsets of $A^*$, or languages, we define $XY = \{xy : x \in X, y \in Y\}$; when $X = Y$ we may write $X^2$ instead of $XX$, $X^3$ instead of $XXX$, etc. As usual $X^*$ denotes the star (Kleene) closure of $X$, which is $\cup_{i\geq 0}X^i$, where $X^0 = \{1\}$, and $X^+$ stands for $X^* \setminus \{1\}$. A non-empty word $u$ is called primitive if for $n > 0$ and $v \in A^*$, $u = v^n$ implies $n = 1$; otherwise it is called imprimitive, that is, when $u = v^n$ for some $n > 1, v \in A^*$. Now we are in a position to give the following definitions of comma-free codes [S] which is first shown in [21].

**Definition 2.1.** *A subset $X \subseteq A^+$ is said to be a* comma-free *code if $X^2 \cap A^+XA^+ = \emptyset$.*

By its definition, it is straightforward to see that a comma-free is a code in the general sense of [1] and, moreover, is a *suffix* code, *prefix* code, hence a *bifix* code, and even an *infix* code. We recall that a subset is infix if $X \cap F(X) = \emptyset$; infix code is an infix set not containing the emptyword. It is noteworthy that all words of a comma-free code must be primitive and that a subset $X$ is a comma-free code if and only if $X$ is an infix code and $X \cap S(X)P(X) = \emptyset$ [S].

*Example 2.2.* The set $X = \{a^2b, ab^2\}$ is a comma-free code on the binary alphabet $A = \{a, b\}$.

We present the following convenient criterion for testing comma-freeness.

**Proposition 2.3.** *A subset $X \subseteq A^+$ is a comma-free code if and only if* (i) *it is an infix set, i.e., $X \cap F(X) = \emptyset$; and* (ii) $(P(X) \cap S(X))^2 \cap X = \emptyset$.

*Proof.* The "only if" part is directly derived from the note mentioned just before the Example 2.2, since $(P(X) \cap S(X))^2 \cap X \subseteq S(X)P(X) \cap X$.

Let $X$ now satify (i) and (ii) and suppose on the contrary that $X$ is not comma-free, that is, $x_1x_2 = ux_3v$ for some $x_1, x_2, x_3 \in X$ and $u, v \in A^+$. Since $X$ is infix, we have $|u| < |x_1|$ and $|v| < |x_2|$, hence $x_1 = us$ and $x_2 = pv$ for some $P, S \in A^+$. Then note that $p \in P(X)$ and $s \in S(X)$ and by the equality $sp = x_3$, we get also $s \in P(X)$ and $p \in S(X)$. All together, that means $p, s \in P(X) \cap S(X)$ and $x_3 = sp \in (P(X) \cap S(X))^2$, or $(P(X) \cap S(X))^2 \cap X$ is

not empty contradicting the assumption (ii). So $X$ must be comma-free and the proof is achieved.

A comma-free code is said to be *maximal* provided it is not contained in another comma-free code. By a routine technique, Zorn's lemma, we see that every comma-free code is embedded in a maximal one, which we call a *completion*; at the same time, a set of words with which a comma-free code can be added to a completion is called a *complement* of the former. Our question now is a stronger one than a simple application of Zorn's lemma may answer: does every finite comma-free code have a finite completion? The aim of this paper is to answer this question for the subclass of solid codes and we are going to give the definition of the main object of our treatment.

**Definition 2.4.** *A subset $X \subseteq A^+$ is called a solid code if it is* (i) *infix: $X \cap F(X) = \emptyset$; and* (ii) *overlap-free: $S(X) \cap P(X) = \emptyset$.*

In view of Proposition 3, every solid code is clearly a comma-free code. Naturally, we define a maximal solid code as one that is not included in any other solid code.

We recall that $u$ overlaps $v$ (on the right of $u$, or on the left of $v$, to be specific) if $uy = xv$ for some $x, y \in A^+$ such that $|x| < |u|, |y| < |v|$; or equivalently, $S(u) \cap P(v) \neq \emptyset$, whose elements are called overlaps; $x$ and $y$ are called left and right borders respectively.

*Example 2.5.* Let $A = \{a, b\}$. For each possitive integer $n$, $X_n = \{ab^n\}$ is a maximal solid code. Moreover, $X_1 = \{ab\}$ is also maximal as a comma-free code, while $X_2 = \{ab^2\}$ is not.

In order to complete a finite solid code to a finite maximal comma-free code, our strategy is as follows. First, we embed it in a finite maximal solid code; this step is readily performed by the procedure in [12] and already mentioned in the introduction. Next, we complete the finite maximal solid code to a finite maximal comma-free code; this step constitutes the core of the present paper. In the next section, we give a characterization of finite maximal solid codes and apply it to describe those words that are feasible to adjoin to give a maximal solid code without changing its comma-free status. The ensuing section, the last one, is devoted to the embedding construction itself and an illustrative example.

## 3. Characterization of Maximal Solid Codes

Let $X$ be an arbitrary subset of $A^*$. Consider the set $M(X)$ of the words not containing any word of $X$ as factor and at the same time not being a factor of $X$. In symbols:

$$M(X) = A^* \setminus (A^*XA^* \cup F(X)).$$

Let $M_0(X)$ be the *infix root* of $M(X)$

$$M_0(X) = M(X) \setminus (A^*M(X)A^+ \cup A^+M(X)A^*)$$

consisting of those words of $M(X)$ that have no proper factors therein. Certainly, every word of $M(X)$ always contains at least a factor in $M_0(X)$ and we call rank of a word $w$, in notation rank($w$), the number of occurences of words of $M_0(X)$, as factors, in $w$.

We need some more technical notations vital to the characterization. Let now $X$ be a solid code; denote by $D(X)$ the set of the words in $M(X)$ that do not overlap $X$ that is

$$\begin{aligned} D(X) &= M(X) \setminus (A^*P(X) \cup S(X)A^*) \\ &= A^* \setminus (A^*XA^* \cup F(X) \cup A^*P(X) \cup S(X)A^*). \end{aligned}$$

The meaning of $D(X)$ is that: a solid code is maximal if and only if this set is empty.

We further define $G(X)$ as the set consisting of the words of $M(X)$ satifying the following conditions:

(a) If $(x, m, y)$, $m \in M_0(X)$, is its leftmost occurence of $M_0(X)$ with $m = la$, $l \in A^*$, $a \in A$ then $xl \in P(X)$;

(b) If $(z, m', t)$ is its rightmost occurence of $M_0(X)$, $m' \in M_0(X)$ with $m' = br$, $r \in A^*$, $b \in A$ then $rt \in S(X)$.

We see at once, as $X$ is a solid code, that $G(X) \subseteq D(X)$. Now denote by $D_1(X)$ the subset of $D(X)$ consisting of those words of rank 1, i.e., those containing only one occurence of $M_0(X)$, which is

$$G_1(X) = \{xmy : m = la = br; xl \in P(X), ry \in S(X); r, l \in A^*, a, b \in A\}.$$

Equipped with the notations above, we come to formulate the next criterion.

**Theorem 3.1.** *For a solid code $X$, the following assertions are equivalent:* (i) *$X$ is a maximal solid code, i.e., $D(X) = \emptyset$;* (ii) $G(X) = \emptyset$; (iii) $G_1(X) = \emptyset$.

*Proof.* Trivially, (ii) implies (iii). Because of $G_1(X) \subseteq G(X) \subseteq D(X)$, (i) implies (ii) and (iii). Now we prove $G_1(X) \neq \emptyset$ whenever $G(X) \neq \emptyset$ which shows that (iii) implies (ii).

Let $w$ be a word of $G(X)$ that contains the least number of occurences of $M_0(X)$, or just the same, $w$ is of the minimum rank. We show that the rank of $w$ is 1. If this is not so then suppose that $(x_1, m_1, y_1)$ is the leftmost and $(x_2, m_2, y_2)$ is the next left most occurence of $M_0(X)$ in $w$. By assumption, $m_1 = l_1a_1$, $l_1 \in A^*$, $a_1 \in A$ and $x_1l_1 \in P(X)$. Let $w_1$ be now the factor spanning the two occurences of $m_1$ and $m_2$ above and let us write as $w_1 = m_1s = tm_2$ for some $s, t \in A^+$. If we further write out $m_1 = b_1r_1, m_2 = l_2a_2$ then $w_1 = b_1r_1s = tl_2a_2$. If we now write $s = s'a_2$ for $s' \in A^*$ then the word $r_1s'$, a factor of $X$, cannot be in $P(X)$ because if it were so, $(xb_1)^{-1}w = r_1y$ would be then in $G(X)$, but with rank smaller than that of $w$. Therefore $r_1s'z \in S(X)$ for some $z \in A^*$, hence $xm_1s'z \in G_1(X)$: $G_1(X)$ is non-empty.

We shall achieve the proof by showing $D(X) \neq \emptyset$ implies $G(X) \neq \emptyset$,i.e., (ii) implies (i). Let $w \in D(X)$ and $(x, m, y)$ be the leftmost occurence of $M_0(X)$ in $w$. We write $m = la$ with $l \in A^*$ and $a \in A$. Since $w$ does not overlap $X$, the word $xl$, which is a factor of $X$, is not a suffix of $X$. Therefore, for some $z \in A^*$, $zxl$ is a prefix of $X$. By a similar argument, we can prove the analogous

statement concerning the last (rightmost) occurence of $M_0(X)$ in $w$, with some $t$ that $zwt \in G(X)$. The proof is complete.

We are interested in the candidates with which we complete $X$ to a maximal comma-free code. They are among those described in the next definition.

**Definition 3.2.** *A word $w$ is said to be $X$-free if it is not a factor of $X$ and its square contains no factor in $X$.*

Evidently every $X$-free word is an element of $M(X)$ and, given the solidity of $X$, $X \cup \{w\}$ is a comma-free code only if $w$ is $X$-free. Thus, in order to complete $X$ to a maximal comma-free code, we have to search for a complememt among $X$-free words.

Suppose now that $X$ is a maximal solid code, we shall describe the $X$-free word more explicitly. Suppose that $w$ is $X$-free and $(x_1, m_1, y_1), \ldots, (x_k, m_k, y_k)$ are all successive occurences of $M_0(X)$ in $w$: $m_1, \ldots, m_k \in M_0, k > 0$. Again, we write $m_i = b_i r_i = l_i a_i$ with $b_i, a_i \in A$, $r_i, l_i \in A^*$ for $i = 1, 2, \ldots, k$ and we call the words $x_1 l_1$ and $r_k y_k$ the *border-pieces* of $w$. For two occurences $(x_i, m_i, y_i)$ and $(x_{i+1}, m_{i+1}, y_{i+1})$, $1 \leq i < k$, we name the word $(x_i b_i)^{-1}(x_{i+1} l_{i+1})$ the *inter-piece* of the two occurences. Note that all of the pieces, inter- and border-, are factors of $X$.

We make now a classification of $X$-free words. Observe that $n$-power of a $X$-free word is also $X$-free, for all $n$; in particular, a square of $X$-free word is $X$-free.

**Theorem 3.3.** *For any maximal solid code $X$, each $X$-free word is either a prefix of $S(X)^*$ or a suffix of $P(X)^*$. Moreover, if the word is of rank more than* 1*, one and only one possibility above takes place.*

*Proof.* First we prove that for any $X$-free word $x$, each inter-piece is either a suffix or a prefix of $X$. In fact, assume that $(x_1, m_1, y_1)$ and $(x_2, m_2, y_2)$ are two consecutive occurences of $M_0(X)$ in $x$ with the inter-piece $(x_1 b_1)^{-1}(x_2 l_2)$ which is neither prefix nor suffix of $X$, where $b_1 r_1 = m_1, l_2 a_2 = m_2$ for $b_1, a_2 \in A$, $r_1, l_2 \in A^*$. There should exist then some words $z_1$ and $z_2$ so that $z_1((x_1 b_1)^{-1}(x_2 l_2))$ is a prefix of $X$ and $((x_1 b_1)^{-1}(x_2 l_2))z_2$ is a suffix of $X$. Now the word (remember that a square of an $X$-free word is also $X$-free)

$$z_1((x_1 b_1)^{-1}(x_2 l_2))a_2 y_2 x_1 b_1((x_1 b_1)^{-1}(x_2 l_2))z_2$$

is in $G(X)$: a contradiction with the maximality of $X$ (the emptyness of $G(X)$).

Now, in virtue of the obvious inequality $\mathrm{rank}(uv) \geq \mathrm{rank}(u) + \mathrm{rank}(v)$ and its consequence, $\mathrm{rank}(x^n) \geq n\,\mathrm{rank}(x)$, we can choose $n$ so that $x^n$ has sufficiently many occurences in $M_0$: $(x_1, m_1, y_1), \ldots, (x_k, m_k, y_k)$, $k$ is large. Take two consecutive occurences $(x_i, m_i, y_i)$ and $(x_{i+1}, m_{i+1}, y_{i+1})$ fairly distant from the beginning and the end of $x^n$, for example, with $|x_i| \geq |x|$ and $|y_{i+1}| \geq |x|$, hence $x$ is a prefix of $x_i$ and a suffix of $y_{i+1}$. We have, as before, $m_i = b_i r_i = l_i a_i$, $m_{i+1} = b_{i+1} r_{i+1} = l_{i+1} a_{i+1}$ $(i < k)$. We consider the following cases.
(a) The inter-piece $(x_i b_i)^{-1}(x_{i+1} l_{i+1})$ is a suffix of $X$. We see at once then that every piece, border- and inter-piece alike, on the left of $(x_i b_i)^{-1}(x_{i+1} l_{i+1})$

is also a suffix of $X$, for if $(x_jb_j)^{-1}(x_{j+1}l_{j+1})$, $j \leq i$, is not a suffix of $X$ then $z(x_jb_j)^{-1}(x_{j+1}l_{j+1})$ is a proper prefix of $X$ for some $z \in A^*$ and the word

$$z((x_jb_j)^{-1}(x_ib_i))((x_ib_i)^{-1}(x_{i+1}l_{i+1}))$$

belongs to $G(X)$. That is impossible by the maximality of the solid code $X$. The same argument for showing that the border-piece $x_1b_1 \in S(X)$.

Thus, by induction, all of the words

$$x_1l_1, (x_1b_1)((x_1b_1)^{-1}(x_2l_2)), \ldots, (x_ib_i)((x_ib_i)^{-1}(x_{i+1}l_{i+1}))$$

are equally in $S(X)^*$. As $x$ is a prefix of $x_i$, hence, of $(x_ib_i)((x_ib_i)^{-1}(x_{i+1}l_{i+1}))$, that is, of $S(X)^*$.

(b) The inter-piece $(x_ib_i)^{-1}(x_{i+1}l_{i+1})$ is a prefix of $X$. This case is handled completely analogously as above, only in the symmetric way to see that $x$ is now a suffix of $P(X)^*$.

To finish the proof, let $x$ have rank at least 2 and $(x_1, m_1, y_1)$ and $(x_2, m_2, y_2)$ be two arbitrary successive occurences of $M_0(X)$ in $x$, with $m_1 = b_1r_1$, $m_2 = l_2a_2$, $b_1, a_2 \in A$, $r_1, l_2 \in A^*$. Suppose for instance that the inter-piece $p = (x_1b_1)^{-1}(x_2l_2)$ is a prefix of $X$, we shall prove that $x$ cannot be a prefix of $S^*(X)$. Actually, if it is not so, we can write as $x = s_1 \ldots s_k s'$ with $s_1, \ldots, s_k \in S(X)$, $k \geq 0$ and $s'$ a prefix of $S(X)$. Let $i$ be the largest index satisfying $|s_1 \ldots s_i| \leq |x_1b_1| \leq |x_2|$. Clearly, $i < k$, since otherwise, $s'$ contains, say, $m_2$. Further, as $|s_1 \ldots s_i| > |x_1b_1|$ we get $|s_1 \ldots s_is_{i+1}| < |x_2m_2|$, otherwise, $s_{i+1}$ contains $m_2$. This inequality indicates that $s_{i+1}$ (a suffix of $X$) overlaps the inter-piece $p$ (a prefix of $X$) on the left of $p$ that contradicts solidity of $X$. The proof is complete.

Next, we need a presentation of $X$-free words in a more special kind of prefixes and suffixes.

**Definition 3.4.** *A proper prefix of $X$ is called primary prefix provided it has no proper suffix which is a proper prefix of $X$, or equivalently, if it is not a product of other non-empty proper prefixes of $X$, i.e., it belongs to $P(X) \setminus A^+P(X) = P(X) \setminus P(X)^+P(X)$. Similarly, a proper suffix of $X$ is called primary suffix provided it has no proper prefix which is a proper suffix of $X$, or equivalently, if it is not a product of other non-empty proper suffixes of $X$, i.e., it belongs to $S(X) \setminus S(X)A^+ = S(X) \setminus S(X)S(X)^+$.*

**Proposition 3.5.** *Every proper prefix (suffix, resp.) decomposes uniquely into a product of primary prefixes (suffixes, resp.).*

*Proof.* Straightforward by induction on length. Note that proper prefix of length 1 is primary.

The next statement is only the translation of Theorem 3.3 into the new terms.

**Corollary 3.6.** *For any maximal solid code $X$, each $X$-free word is either a prefix of a product of primary suffixes of $X$ or a suffix of a product of primary*

*prefixes of $X$. Morever, if the word is of rank more than* 1*, one and only one of the possibilities takes place.*

**Definition 3.7.** *A primary prefix (suffix, resp.) is called a maximal primary prefix (suffix, resp.) if it is not a proper factor of any other primary prefix (suffix, resp.).*

*Remark.* Not every solid code disposes maximal primary prefixes or suffixes. For example, the solid code $XY^*Z$, where $A = X \cup Y \cup Z$ is a partition with $Y \neq \emptyset$. But every finite solid code definitely has at least one maximal primary prefix and one maximal primary suffix, since the number of both primary prefixes and suffixes is finite.

Now we prove an important property of maximal primary prefixes and suffixes.

**Theorem 3.9.** *Let $s$ ($p$, resp.) be a maximal primary suffix (prefix, resp.) of a maximal solid code $X$ and $u$ be a word of $sS(X)^*$ ($P(X)^*p$, resp.) but not a factor of $X$. Then for an $X$-free word $v$ both $uv$ and $vu$ do not contain any factor in $X$ if and only if $v \in S(X)^*$ ($v \in P(X)^*$, resp.).*

*Proof.* If $v \in S(X)^*$ then $uv, vu \in S(X)^*$ and both of them obviously do not contain $X$ by its solidity. Conversely, assume that $uv$ and $vu$ do not contain any word of $X$ as a factor, which implies that $vu^2$ is $X$-free. Since $u^2$ belongs to $S(X)^*$, it has no factor in $X$ and since $u$ is not a factor of $X$, $\mathrm{rank}(u) \geq 1$, therefore, $\mathrm{rank}(u^2) \geq 2$. Since $u^2$ is in $S(X)^*$ already, the possibility that $vu^2$ is a suffix, hence so is $u^2$, of $P(X)^*$ is excluded by Theorem 3.3. So we get that $vu^2$ is a prefix of $S(X)^*$.

We now write out $u = ss_1 \dots s_k$, for $s_1, \dots, s_k \in S(X)$, $k \geq 0$ and $vu^2 = r_1 \dots r_{n-1} r_n$, where $n > 0$, $r_1, \dots, r_{n-1}$ primary suffixes of $X$ and $r_n$ a prefix of $S(X)$ and

$$vss_1 \dots s_k ss_1 \dots s_k = r_1 \dots r_{n-1} r_n.$$

Let $i$ be the greatest integer such that $|r_1 r_2 \dots r_i| \leq |v|$. Clearly $i < n-1$, otherwise, $u^2$ would be a factor of $r_n$, hence, a factor of $X$ despite the assumption. Therefore $i+1 \leq n-1$ and the suffix $r_{i+1}$ is primary. Suppose that we have the strict inequality $|r_1 \dots r_i| < |v|$. Then, concerning $r_{i+1}$ we have two possibilities.
(a) $|r_1 \dots r_i r_{i+1}| < |vs|$ that means $s$ has a non-empty suffix of $r_{i+1}$ which is indeed a suffix of $X$, as a proper prefix. This is against the primarity of $r_{i+1}$.
(b) $|r_1 \dots r_i r_{i+1}| \geq |vs|$: that means $s$ is a proper factor of $r_{i+1}$. This is against the maximality of the primary $s$.

Both cases lead to a contradiction, so we must have $|r_1 \dots r_i| = |v|$, therefore, $r_1 \dots r_i = v \in S(X)^*$ and this completes the proof.

All of the premises for completing procedure are ready and we come to the next section for the completion construction itself.

## 4. Completion

As said before, we can complete every finite solid code to a finite maximal one, thus, for the completing task it is enough to consider the finite maximal solid code. Let $X$ be a finite maximal solid code and let the set of primary suffixes (prefixes, resp.) be denoted as $S_p$ ($P_p$, resp.). We know by Corollary 3.4 that every $X$-free word is a prefix of $S_p^*$ or suffix of $P_p^*$, but the converse does not hold. Nevertheless, it is easy to see that every word in $S(X)^*$ or $P(X)^*$ not a factor of $X$ is $X$-free. Our tactics to complete $X$ is to choose a complement in $S_p^*$ (equally in $P_p^*$, if we see fit, with the symmetric treatment) and, more than that, we require that the complement contain at least one word in $sS_p^*$ for a *maximal* primary factor $s$. Once this is done, when the comma-freeness and maximality of the completion are concerned, the prefixes of $S(X)^*$ that are not in $S(X)^*$ and the suffixes of $P(X)^*$ are out of question, by Theorem 3.9. Consequently, we need to take into account only the words of $S_p^*$ and we could make use of a *coding morphism* to switch to a new alphabet $B$ of the same cardinalty as the set $S_p$. Although the inverse morphic image of a comma-free code is a comma-free code, the (direct) image of a comma-free code unfortunately is not, but we can remedy this situation by pushing further the requirement that *all* words of the complement are in $sS_p^*$ for some maximal primary suffix $s$ exploiting a property of maximal primary suffixes that will be done in the next assertion.

**Theorem 4.1.** *Let $B$ be an alphabet of the same cardinality as $S_p$ ($P_p$, resp.), $c$ a letter of $B$ and $s$ ($p$, resp.) an arbitrary but fixed maximal primary suffix of $X$ (maximal primary prefix of $X$, resp.). Then for the coding morphism $\theta$ from $B^*$ onto $S_p^*$ ($P_p^*$, resp.) such that $\theta(B) = S_p$ ($\theta(B) = P_p$, resp.) and $\theta(c) = s$ ($\theta(c) = p$, resp.) and for any finite maximal comma-free code $Y$ on $B$ such that $Y \subseteq cB^*$ ($Y \subseteq B^*c$, resp.) and the words and the left borders (right borders, resp.) of $Y$ are all long enough that their morphic images under $\theta$ are no shorter than the maximal length of $X$, the set $X \cup \theta(Y)$ is a finite maximal comma-free code on $A$, i.e. a finite completion of $X$.*

*Proof.* We prove the theorem for the suffix case, the other case requires just the same reasoning on the mirror image. We first observe that if a primary suffix $s$ is a factor of a product $s_1 s_2 \dots s_k$ of primary suffixes $s_i$, $i = 1, 2, \dots, k$, then $s$ must be a factor, not prefix, of some $s_i$, by primarity, and $s = s_i$ if, besides, $s$ is a maximal primary suffix. Consequently, if the product of primary suffixes $ss'_1 \dots s'_l$ is a factor of $s_1 s_2 \dots s_k$,i.e., $s_1 s_2 \dots s_k = uss'_1 \dots s'_l v$ then $s = s_i$, $s'_1 = s_{i+1}, \dots, s'_l = s_{l+i}$ and $u = s_1 \dots s_{i-1}, v = s_{l+1} \dots s_k$ for some $i$ with $i + l \leq k$, or equivalently, $\theta^{-1}(s_1 s_2 \dots s_k) = u'\theta^{-1}(ss'_1 \dots s'_l)v'$ for $u' = \theta^{-1}(s_1) \dots \theta^{-1}(s_{i-1}) \in B^*, v' = \theta^{-1}(s_{l+1}) \dots \theta^{-1}(s_k) \in B^*$.

Now we verify the comma-freeness directly by definition. Suppose on the contrary that $X \cup \theta(Y)$ is not comma-free. Note that by assumption $X \cup \theta(Y)$ is infix. Since $X$ is solid and all words in $\theta(Y)$, as well as their product, are $X$-free (they are in $S_p^* \subseteq S^*$) then only the the following cases need a treament. In all cases below $u, v \in A^+$, $x \in X$ and $w_1, w_2, w_3 \in \theta(Y)$.
(a) $uxv = w_1 w_2$. This case is impossible by the fact that $w_1 \in S(X)^*$ and $X$ is

solid.
(b) $uw_1v = xw_2$. This case shows that $|x|$ is larger than the length of the left border of $\theta(Y)$ in concern. By the observation in the first passage, this border is the image of a left border in the overlapping of $\theta^{-1}(w_1)$ and $\theta^{-1}(w_2)$, it cannot be shorter than $|x|$ by assumption, since $\theta$ does not erase any symbol: a contradiction.
(c) $uw_1v = w_2x$. This case is obviously impossible by $w_1 \in S(X)^*$ and by solidity of $X$.
(d) $w_1w_2 = uw_3v$. If we write

$$w_1 = \theta(cx_1 \dots x_k), \quad w_2 = \theta(cy_1 \dots y_l), \quad w_3 = \theta(cz_1 \dots z_m)$$

for $x_1, \dots, x_k, y_1, \dots, y_k, z_1, \dots z_k \in B$ and $cx_1 \dots x_k, cy_1 \dots y_k, cz_1 \dots z_k \in Y$ and put

$$\theta(x_1)=s'_1,\dots,\theta(x_k)=s'_k, \theta(y_1)=s''_1, \dots, \theta(y_l)=s''_l, \theta(z_1)=s_1, \dots, \theta(z_m)=s_m,$$

we see that $ss_1 \dots s_m$ is a factor of $ss'_1 \dots s'_k s'_k s s''_1 \dots s''_l$. By the observation above, the product $ss_1 \dots s_m$ is actually a "subproduct" of $ss'_1 \dots s'_k s'_k s s''_1 \dots s''_l$ with the factors $s$, $s_1, \dots$ and $s_k$ are equal to the $k+1$ successive factors of $ss'_1 \dots s'_k s'_k s s''_1 \dots s''_l$. This means that shifted back to $B$, the coding alphabet, $cx_1 \dots x_k cy_1 \dots y_l \in B^+ cz_1 \dots z_m B^+$, or $w_1w_2 \in B^+ w_3 B^+$ which is impossible in view of comma-freeness of $Y$.

Finally, we prove the maximality of $X \cup \theta(Y)$. Let $v$ be an arbitrary word of $A^*$, not in $\theta(Y)$. If $v$ is a suffix of $P(X)^*$ or a prefix of $S(X)^*$ but not in $S^*$ then by Theorem 3.9, for any word $u$ in $\theta(Y)$ either $uv$ or $vu$ has a factor in $X$. Alternatively, suppose that $v \in S(X)^*$ then $v = \theta(y)$ for a word $y \in B^*$ not in $Y$. Now that $Y$ is a maximal comma-free on $B$, there exist two words $y_1, y_2 \in Y$ such that $\{y_1, y_2, y\}$ is not a comma-free code, consequently, $\{\theta(y_1), \theta(y_2), \theta(y)\} = \{\theta(y_1), \theta(y_2), v\}$ is not comma-free that means $X \cup \theta(Y)$ is maximal comma-free. The proof is complete.

Let now $B$ be an alphabet with $|B| \geq 2$ and $c \in B$. For a word $u \in B^*$ we use $|u|_c$ to indicate the number of occurences of $c$ in $u$ and put for arbitrary but fixed integers $n > 0, t > 0$

$$\begin{aligned} Y_1 &= \{cu : |u| = n+1, |u|_c = 0\}; \\ Y_2 &= \{cu : 0 \leq |u| \leq n, |u|_c = 0, u \neq v\}^t cv; \\ Y_3 &= \cup_{j=1}^{t-1} cv\{cu : 0 \leq |u| \leq n, |u|_c = 0, u \neq v\}^j cv, \end{aligned}$$

where $u$ runs through $B^*$. The following proposition shows that the comma-free code $Y$ with properties required in the preceding theorem does really exist.

**Proposition 4.2.** *The set $Y = Y_1 \cup Y_2 \cup Y_3$ is a finite maximal comma-free code on $B$ satisfying the requirements of Theorem 4.1 when $n$ and $t$ are sufficiently large.*

*Proof.* First we verify that $Y$ is comma-free. The fact that $Y$ is infix is straightforward to see. Next, a non-empty prefix of $Y$ always begins with the letter $c$, if it is also a proper suffix, it must be of the form $cv$ or $cu_1cu_2 \dots cu_kcv$ for $k > 0$ and for some words $u_1, u_2, \dots, u_k$ all of length less or equal to $n$ containing none

occurence of $c$ and different from $v$. It follows that the common element of $P(X)$ and $S(X)$ now reduces to the only one $cv$, hence $(S(Y) \cap P(Y))^2 \cap Y = \emptyset$ and the comma-freeness follows.

To prove maximality, let $w$ be an arbitrary $Y$-free word. If $w$ does not begin with $c$ then taking any word $y$ in $Y_2$ or $Y_3$ we see that $wy$ contains a word in $Y_1$.

Now suppose that $w$ has the first letter $c$, that is, $w$ has the presentation $w = cu_1 \dots cu_k$ for $k > 0$ and $u_1, \dots, u_k \in (B \setminus \{c\})^*$. If $|u_i| > n$ for some $i$ then $cu_i$ has a prefix in $Y_1$; otherwise, if all $u_1, \dots, u_k$ have length less than or equal to $n$ and they are diffrent from $v$ then taking any word $y$ in $cv(B \setminus \{c\}) \subseteq Y_1$ we see that $w^s y$ contains a factor in $Y_2$ when $s$ is sufficiently large. Now, if all $u_1, \dots, u_k$ are equal to $v$ then $w = (cv)^k$ which is either imprimitive or a factor of $Y$.

Finally, the last alternative, there exist among $u_i$'s one equal to and one different from $v$. It is easy to see that $w^2$ then has at least two occurences of $v$ and if any two consecutive of them are separated by no less than $t$ occurences of $cu_i$'s, $w^2$ contains a word in $Y_2$ and if any two consecutive occurences are separated by less than $t$ of but no less than one of such occurences, they surely do exist, $w^2$ then contains a word in $Y_3$. All issues show that $Y$ is maximal.

It is straightforward to check that the comma-free codes $Y$ with their parameters satisfies the assumption of Theorem 4.1: it begins with the same letter $c$, their words have length at least $n + 2$ and the left borders all have length greater than $\min(n + 1, t)$ with $n, t$ arbitrarily large. The proof is complete.

Note that Theorem 4.1 provides completion just in case the maximal solid code has at least two primary prefixes or suffixes; what about the solid codes that have only one primary prefix and one primary suffix? It turns out that we need not bother about them: there are only two with this property, they are $\{ab\}$ and $\{ba\}$ and the alphabet is then binary $A = \{a, b\}$.

These finite maximal codes (Example 3.5) are at the same time maximal comma-free codes! We give a concise explanation. As a litteral prefix is primary, every word of $X$ has the form $a^n b$ for, say, the same letter $a$ and some letter $b \neq a$. Symmetrically, relative to suffixes, every word of $X$ has the form $al^m$ for the same letter $l \neq a$. Consequently, the alphabet is binary $A = \{a, b\}$ and $X$ is the unique $\{ab\}$ in this case. The other case $\{ba\}$ is obtained when the first letter of the word is $b$. We show details of performing completion by a simple example.

*Example 4.3.* Let $A = \{a, b\}$ and $X = \{a^2 b\}$. The solid code $X$ is maximal. The primary suffixes are $b, ab$ and $ab$ is a unique maximal primary suffix. So $B = \{c, d\}$ and $\theta(c) = ab, \theta(d) = b$. As $a^2 b$ is of length 3 and the selected maximal primary suffix is of length larger than 1, it is enough to take $n = 1$ and $t = 2$. Now $v = d$ (a unique choice) and

$Y_1 = \{cd^2\}$;

$Y_2 = \{cu : 0 \leq |u| \leq 1, u \neq v, |u|_c = 0\}^2 cd = \{c\}^2 cd = c^3 d$;

$Y_3 = \cup_{j=1}^{1} cd\{cu : 0 \leq |u| \leq 1, u \neq v, |u|_c = 0\}^j cd = cd\{c\}^2 cd = \{cdc^2 d\}$.

Now $Y = Y_1 \cup Y_2 \cup Y_3 = \{cd^2, c^3 d, cdc^2 d\}$ and $\{a^2 b\}$ has the finite comma-free completion

$$\{a^2b\} \cup \{\theta(cd^2), \theta(c^3d), \theta(cdc^2d)\} = \{a^2b; ab^3, ababab^2, ab^2abab^2\}.$$

We can directly verify by definition that it is a maximal comma-free code. We close this work with a remark that we have just succeeded in solving in general the completion problem for the class of finite comma-free codes which we hope to publish in a forthcoming paper.

*Acknowledgement.* Last but not least, I am grateful to the reviewer for his meticulous reading, suggestions and corrections.

## References

1. J. Berstel and D. Perrin, *Theory of Codes*, Academic Press, Orlando, 1985.
2. F. H. C. Crick, J. S. Griffith, and L. E. Orgel, Codes without commas, *Proc. Nat. Acad. Sci. USA* **43** (1957) 416–421.
3. A. Ehrenfeucht and G. Rozenberg, Each regular code is included in a regular maximal code, *RAIRO Informatique Théorique* **20** (1986) 89–96.
4. S. W. Golomb, *Proceedings of the Symposium on Mathematical Problems in the Biological Sciences*, Amer. Math. Soc., 1961.
5. S. W. Golomb and B. Gordon, Codes with bounded synchronization delay, *Information and Control* **8** (1965) 355–372.
6. S. W. Golomb, B. Gordon, and L. R. Welch, Comma-free codes, *Canad. J. Math.* **10** (1958) 202–209.
7. S. W. Golomb, L. R. Welch, and M. Delbrück, Construction and properties of comma-free codes, *Medd. Dan. Vid. Selsk.* **23** (1958).
8. B. H. Jiggs, Recent results in comma-free codes, *Canad. J. Math.* **15** (1963) 178–187.
9. H. Jürgensen and S. Konstantinidis, *Codes*, in: G. Rozenberg, A. Salomaa (eds.), *Hanbook of Formal Languages*, Vol. 1, Springer–Verlag, Berlin, 1997, pp. 511–607.
10. H. Jürgensen, M. Katsura, and S. Konstantinidis, Maximal solid codes, *J. Automata, Combinatorics and Languages* **6** (2001) 25–50.
11. H. Jürgensen and S. S. Yu, Solid codes, *J. Information Processing and Cybernetics, EIK* **26** (1990) 563–574.
12. N. H. Lam, Finite maximal solid codes, *Theoretical Computer Science* **262** (2001) 333–347.
13. N. H. Lam, Completing comma-free codes, *Theoretical Computer Science* (to appear).
14. Al. A. Markov, An example of an independent system of words which cannot be included in a finite copmplete system, *Matematicheskie Zametki* **1** (1967) 87–90.
15. D. Perrin, Completing biprefix codes, *Theoretical Computer Science* **28** (1984) 329–336.
16. A. Restivo, On codes having no finite completions, *Discrete Math.* **17** (1977) 306–316.
17. H. J. Shyr, *Free Monoids and Languages*, Lecture Notes Hon Min Book Company, Taichung, 1991.
18. L. Stryer, *Biochemistry*, Freeman, San Fransisco, 1975.

19. J. D. Watson and F. H. C. Crick, A structure for deoxyribose nucleic acid, *Nature* **171** (1953) 737.
20. M. Yčas, *The Biological Code*, North-Holland, Amsterdam, 1969.
21. S. S. Yu, A characterization of intercodes, *Inter. J. Computer Math.* **36** (1990) 39–45.
22. L. Zhang and Z. Shen, Completion of recognizable bifix codes, *Theoretical Computer Science* **145** (1995) 345–355.