ELSEVIER

Theoretical Computer Science 262 (2001) 333–347

Theoretical Computer Science

www.elsevier.com/locate/tcs

Finite maximal solid codes

Nguyen Huong Lam

Hanoi Institute of Mathematics, P.O. Box 631, Bo Ho, 10000 Hanoi, Viet Nam

Received 30 June 1998; revised 5 April 2000; accepted 22 June 2000
Communicated by M. Nivat

Abstract

Solid codes, a special class of bifix codes, were introduced recently in the connection with formal languages. However, they have a much earlier history and more important motivation in information transmission dating back to the 1960s. In this paper, they are studied as an independent subject in the theory of variable-length codes. It is shown that every finite solid code is contained in a finite maximal one; based on further analysis of the structure of finite maximal solid codes, an algorithm is proposed to construct all of them starting from the most simple and evident ones.

Keywords: Solid code; Completion; Finite maximal solid code

1. Introduction

The concept of solid code is introduced as an auxiliary means by Shyr and Yu [12] in the context of characterizing disjunctive domains, an object of purely language-theoretic nature. Originally, solid codes are defined as codes satisfying a uniqueness condition concerning a certain kind of factorizations of words which implies, immediately, that a solid code is a code in general. Further, Jürgensen and Yu [7] studied solid codes in detail, revealing some basic combinatorial properties, closure and non-closure properties of the class of solid codes. They raised the question of characterizing the maximal solid codes and embedding finite solid codes in finite maximal solid codes. They gave an equivalent combinatorial definition of solid codes, which we find more convenient for our purposes and we will reformulate in the sequel.

Solid codes as such have a communication-theoretic origin in the work of Levenshtein and Romanov dating back to the mid-1960s where solid codes are called codes without overlaps. In [8] the maximal size of solid codes of a constant word length is

E-mail address: nhlam@hanimath.ac.vn (N.H. Lam).

0304-3975/01/$ - see front matter
PII: S0304-3975(00)00277-2

bounded above and below by exponential functions in length. In [10] it is shown that the class of finite solid codes coincides with the class of languages accepted by a certain type of deterministic finite tranducers, the state-invariant decoders without look-ahead (see also [6] for further details). Solid codes have not only amazing mathematical implications but remarkable error-correcting capabilities in very noisy channels. During transmission, encoded messages may be distorted due to environmental conditions or faults in the channel or may be subject to noise and tampering: insertion or deletion of symbols by a hostile party. Solid codes have a remarkable resistance to insertion and deletion by their synchronization capabilities: Every correctly transmitted code word can also be decoded correctly and without delay, which is not true for most kinds of codes. This potential issue of the application makes them an object of intense attention and motivates some further work on their communication-theoretic aspects (see [6] for details).

In this paper we follow the traditional line in the general theory of codes, free of error-correcting considerations [1]. First, it is a common question to embed a code of a given class in a maximal one in this class, called a *completion* of the previous code in the class (the terminology stems from the theory of codes in which for a finite code – a regular code more generally – the maximality is equivalent to completeness [1]). For instance, recall the result of Ehrenfeucht and Rozenberg that every regular code can be embedded in a regular maximal one [4] or the result of Bruyère, Wang and Zhang on completing codes with bounded deciphering delay [2], or the recent result of Zhang and Shen on completing regular bifix codes [13]. As for solid codes, Jürgensen and Konstantinidis have conjectured in [6] – with some reservation – that the embedding question for finite solid codes has a positive answer.

In this paper, we confirm the conjecture of Jürgensen and Konstantinidis by adding one more embedding construction to prove that each finite solid code always has a finite completion, that is, a finite maximal solid code containing the given one. As a matter of fact, our construction works primarily for the class of regular solid codes to yield regular maximal solid codes and, when restricted to finite solid codes, it yields finite maximal ones. It must be said that not for every finite code the embedding within the class is possible. Even for finite bifix codes a finite completion is not always possible; for example, the bifix code $\{a^n, b^m\}$ with the distinct positive integers m, n and letters a, b is not included in any finite maximal bifix code. The reader may consult [1, 9] for the very interesting theory of bifix codes. Thus, the outcome for finite solid codes, as a subclass of the class of bifix codes, is more favorable.

As the second principal result, we propose an algorithm giving all finite maximal solid codes. The situation is much reminiscient of the procedure of Césari [3, 1] giving all finite maximal bifix codes by successive internal transformations from the uniform codes.

Thus, combined with the completion procedure, the algorithm allows one to obtain all *finite* solid codes by taking subsets of the finite maximal ones, an outcome that cannot be envisaged for bifix codes.

Now we come to the formal and exact presentation of notions and notations.

2. Solid codes

Let A be a finite alphabet of at least two elements, A^* the free monoid over A, the elements of which are *words*, and the identity of which, the *empty* word, is denoted by 1. The elements of A are called *letters*; the number of letters in a word w is the length of w, denoted $|w|$. We use the notation $A^+ = A^* - \{1\}$.

For any two sets S and T we use the notation $S - T$ and $S + T$ to denote their difference and union, respectively. For any subsets X and Y of A^*, we denote $XY = \{xy\colon\ x \in X, y \in Y\}$; $X^n = XX \cdots X$ (n times) for a positive integer n; and $X^* = \{1\} + X + X^2 + \cdots = \{1\} + \sum_{n \geqslant 1} X^n$, the Kleene closure of X.

A word v is a factor of the word u if there exist two words x, y such that $u = xvy$; the factor v is a prefix if $x = 1$, a suffix if $y = 1$. A factor is *proper* if it is not the empty word and not the whole word itself, that is, when x or y or both are not empty. Proper prefixes and proper suffixes are defined accordingly.

For a subset X of A^* we denote by $F(X)$, $P(X)$ and $S(X)$ the collections of proper factors, proper prefixes and proper suffixes of the words in X, respectively. X is a prefix code if no word in it is a proper prefix of the others, that is $X \cap P(X) = \emptyset$ and symmetrically, is a suffix code if $X \cap S(X) = \emptyset$. It is a bifix code if it is both a prefix code and a suffix code.

Let u and v be words. We say that u *overlaps* v, more specifically, *on suffix*, if $S(u) \cap P(v) \neq \emptyset$, in other words, if there exist non-empty words x, y and w such that $u = xw$ and $v = wy$; moreover, u *overlaps* v *on prefix* if v overlaps u on suffix. We say that two words overlap if one overlaps the other. We sometimes call the factor w an *overlap* of u and v and then say that u and v have an overlap w, or else u has an overlap w with v or vice versa.

We now present the characterization due to Jürgensen and Yu [7] which we use as definition.

Definition 2.1. A subset X of A^* is said to be *a solid code* if no word of it is a factor of the others and no two words, not neccessarily distinct, overlap; equivalently, $X \cap F(X) = \emptyset$ and $P(X) \cap S(X) = \emptyset$.

A solid code X is *maximal* if it ceases to be a solid code when a word from $A^* - X$ is added. A straightforward characterization: X is a maximal solid code if and only if every word outside X either has an overlap with X, or is a factor of some word of X or has a factor in X or overlaps itself. It is easy to see that a solid code is a bifix code and that every subset of a solid code is also a solid code. By Zorn's lemma, every solid code is contained in a maximal solid code. Here are some maximal solid codes. Let $A = \{a, b\}$.

Example 2.2. The sets $\{ab^n\}$ are maximal solid codes for each $n \geqslant 1$. That the code $\{ab^n\}$ is solid is evident. Next, if a word has no overlaps with ab^n then it must belong

to aA^*b, and if in addition to this, when $n \geqslant 2$, it has no factors ab^n then it has a suffix in $\{ab, ab^2, \ldots, ab^{n-1}\}$ meaning that it must be a factor of ab^n.

The following examples are from [7], the maximal solidity of them is verified directly without great difficulty.

Example 2.3. Non-regular solid code. The set $\{aba^ib^2 : i = 2, 4, 8, \ldots\} + \{aba^iba^jb^2 : i = 1, 2, 3, \ldots;\ j \neq 2, 4, 8, \ldots\}$ is an infinite, non-regular, maximal solid code.

Example 2.4. Infinite regular solid code. Let A be a finite aphabet with at least three letters and $A = X + Y + Z$ be an arbitrary partition of A. Then the set XY^*Z is a maximal solid code, infinite and regular.

The solid codes in the following assertion deserve special attention as they are the cornerstones for all finite maximal solid codes to be built on.

Proposition 2.5. *For every partition $A = I + K + J$ with non-empty I and J of the alphabet A, the subset $IJ + K$ is a finite maximal solid code.*

Proof. If a word $u \in A^*$ is in I^* or J^* or JA^*I or A^*KA^* then we are done: u has either a factor in K or overlaps with IJ. The alternative that remains guarantees that u has a factor in IJ. This shows the maximal solidity of the code. □

We now prove several properties of finite maximal solid codes. For a solid code X we define

$$I(X) = \{a \in A : aA^+ \cap X \neq \emptyset\},$$

the collection of the first letters of the words of length at least 2; symmetrically,

$$J(X) = \{a \in A : A^+a \cap X \neq \emptyset\}$$

as the collection of the last letter of the words of length at least 2 in X; and

$$K(X) = A \cap X$$

is the set of words of X that are letters. We write I, J and K instead when X is understood. Solidity implies that the sets I, J, K are pairwise disjoint, to avoid ovelaps. For finite maximal solid codes they even cover the entire alphabet. Note that the set K might be empty, and I is non-empty if and only if J is.

Proposition 2.6 (Jürgensen and Yu [7]). *Let X be a finite maximal solid code then $I(X)$, $J(X)$, $K(X)$ form a partition of the alphabet, that is the union $A = I(X) + J(X) + K(X)$ is disjoint.*

We derive a necessary condition for a finite solid code to be maximal. The result is typical for this kind of maximality.

Proposition 2.7 (Jürgensen and Yu [7]). *For a finite maximal solid code X and for all letters $a \in I(X)$, $b \in J(X)$ there is a unique pair (m,n) of positive integers such that $X \cap a^*b^* = \{a^m b^n\}$.*

Proof. The uniqueness of the pair follows by the solidity of X because for any two distinct words of a^*b^* inevitably one overlaps the other or one is a factor of the other one.

For the existence, consider a word $a^p b^q$ with p,q greater than the maximum length of words of X. By the maximality of X and by the assumption about p,q and by the fact that $a \notin J$, $b \notin I$ it follows that $a^p b^q$ has a factor in X and such a factor has the form $a^m b^n$ as was to be proved. □

We present now an useful remark due to H. Jürgensen which says that concerning maximal solid codes we can disregard the self-overlapping requirement.

Remark 2.8. A necessary and sufficient condition for a solid code X to be maximal is that every word outside X either is a factor of X or has a factor in X or overlaps X.

Proof. If $I = J = \emptyset$ then X is the underlying alphabet and the remark is true. Otherwise, if I and J are both non-empty we use fact that, for every word u and arbitrary two distinct letters a and b, the word $a^{|u|}ub^{|u|}$ does not overlap itself and, moreover, if we choose $a \in I$ and $b \in J$ then $a^{|u|}$ and $b^{|u|}$ both contain no factors in X. □

This argument is used also in the proof of the next proposition.

Finally, the following characterization of finite maximal codes will be of use in the sequel.

Proposition 2.9. *A necessary and sufficient condition for a finite solid code X to be a maximal solid code is that for every word $w \in A^*$ and for every letters $a \in I(X)$ and $b \in J(X)$ and for any integers m,n not less than the maximum length of X, the word $a^m w b^n$ has a factor in X.*

Proof. The sufficiency is easy. Let w be an arbitrary word and x be a factor w in X. Then x cannot be a factor a^m or b^n as $a \notin J + K$ and $b \notin I + K$. Therefore x overlaps w or is a factor of w, or has w as a factor. Thus for $w \notin X$, X is not a solid code. This shows that X is a maximal solid code.

Conversely, assume that X is a maximal solid code. With m,n is not less than the maximum length of X, $a^m w b^n$ cannot be a factor of any word in X nor overlap any word in X. As X is maximal, by the Remark 2.8, $a^m w b^n$ has a factor in X. The proof is complete. □

For further aspects of and background on solid codes we refer to [5–7, 11, 12].

3. Finite completion of a finite solid code

Every solid code is included in a maximal solid code – a routine application of Zorn's lemma allows one to state this. Moreover, by a more sophisticated manipulation, finite solid codes are shown to be contained in finite maximal solid codes. This section is devoted to establish this result.

Consider the set R of the words of A^* which are not factors of words in X, no suffix of which is in P, but every proper prefix of which has a suffix in P; formally

$$R = (A^* - A^*P - F) \cap P^*A.$$

Note that $X \subseteq R$ by the solidity of X and by the fact that $X \subseteq PA \subseteq P^*A$. Here is an essential property of the set R.

Proposition 3.1. *No pair of words, not necessarily distinct, of R overlap. If a word is a factor of another one, the first one is a suffix of the latter one.*

Proof. Suppose that the words r_1, r_2 of R overlap. We have $r_1 = xw$, $r_2 = wy$ for x, y, $w \in A^+$. The overlapping factor w is a proper prefix of r_2 as y is non-empty, hence w has a suffix in P which implies that r_1, having w as a suffix, also has a suffix in P. This contradicts the fact that $r_1 \in R$.

For the second claim, suppose that r_1 is a factor of r_2: $r_2 = xr_1y$ for some $x, y \in A^*$. If y is not empty, xr_1 is a proper prefix of r_2, hence xr_1 has a suffix p in P. Then r_1 is either a suffix of p, which implies that $r_1 \in F$, or p is a suffix of r_1, that is $r_1 \in A^*P$ contradicting the fact that $r_1 \in R \subseteq A^* - A^*P - F$. Thus, we have $y = 1$ and r_1 is a suffix of r_2. This completes the proof. □

Define now the set

$$Q = R - A^+R$$

of those words of R having no proper prefixes in R. It is noteworthy that, by definition, every word in R has a suffix in Q. If X is a *regular* subset then R, and hence Q, is a regular set as well.

Theorem 3.2. *Let X be a solid code then X is a subset of Q and Q is a maximal solid code. If, moreover, X is regular then Q is also regular, that is every regular solid code is included in a regular maximal solid code.*

Proof. First, as remarked $X \subseteq R$. By definition no word in R is a factor of a word in X. Thus, a fortiori, X has no proper suffix in R. This means that $X \subseteq R - A^+R = Q$.

Moreover, by Proposition 3.1, no pair of words in Q, as a subset of R, overlaps; also, by definition, no word of Q is a factor of the others. This means that Q is a solid code.

Finally, we prove that Q is maximal. It suffices to prove that an arbitrary word w which is not a factor of X and which does not have overlaps with any word in X, must contain a factor in Q or have overlaps with Q.

Let w_1 be a shortest non-empty, not neccessarily proper, prefix of w with no suffix in P. Such a prefix always exists since the word w itself is without suffixes in P. That is $w_1 \in (A^* - A^*P) \cap P^*A$. If, in addition to this, $w_1 \notin F$ then obviously $w_1 \in R$ and w_1 has a suffix $s \in Q$ which is a factor w and we are done. □

Otherwise, if $w_1 \in F$ then there is a word $u_1 \in A^+$ such that $u_1w_1 \in P$ and u_1 is non-empty since w_1 has no suffix in P. Now let w_2 be the shortest non-empty prefix of w such that u_1w_2 has no suffix in P, that is, $u_1w_2 \notin A^*P$. This prefix w_2 exists because u_1w has no suffix in P and moreover $|w_2| > |w_1|$ since $u_1w_1 \in P$ and $w_1 \in P^*A$.

If $u_1w_2 \in F$, we repeat the argument to get the prefices $w_3, w_4, \ldots$ of w and the words $u_2, u_3, \ldots$ such that $u_2w_2 \in P, u_3w_3 \in P, \ldots$ and $u_2w_3 \in F, u_3w_4 \in F, \ldots$ with $\ldots > |w_4| > |w_3| > |w_2| > |w_1|$. Since the lengths of prefixes w_i are strictly increasing and bounded above by $|w|$, the argument cannot be repeated infinitely. It must terminate in the l-th step with

$$u_{l-1} \ldots u_1 w_l \notin F(X)$$

and

$$u_{l-1} \ldots u_1 w_l \notin A^*P, \quad u_{l-1} \ldots u_1 w_l \in P^*A.$$

Thus $u_{l-1} \ldots u_1 w_l \in R$, hence it has a suffix $s \in Q$. Either s is a suffix of w_l, therefore a suffix of w, or w_l is a suffix of s which implies that s overlaps w. In both cases w has a factor in Q or an overlap with Q as was to be proved.

We show further that with the very same construction Q will be finite whenever X is finite.

Theorem 3.3. *If X is a finite solid code, then Q is a finite maximal solid code. Precisely, let X be of maximal word length n then $Q = A$ if $n = 1$ or, if $n > 1$, Q is of maximal word length at most $2n - 2$.*

Proof. If $n = 1$, or $X \subseteq A$, we have $P = \emptyset$, $F = \emptyset$, therefore $P^* = \{1\}$, $Q = (A^* - \emptyset - \emptyset) \cap A = A$.

Now let $n > 1$. Suppose that there exists $w \in Q$ with $|w| > 2n - 2$. We write $w = u_1u_2 \ldots u_ka$, where $k \geqslant 1$, $u_1, \ldots, u_k \in P$, $a \in A$. Let l be the least index such that

$$|u_lu_{l+1} \ldots u_ka| > 2n - 2. \qquad (*)$$

Indeed $l \geqslant 1$ and

$$|u_{l+1} \ldots u_ka| \leqslant 2n - 2.$$

If $u_{l+1} \ldots u_ka \in F$ then $|u_{l+1} \ldots u_ka| \leqslant n - 1$, which implies by $(*)$ that $|u_l| > 2n - 2 - (n - 1) = n - 1$. This is a contradiction as u_l is a proper prefix of some word in X

of length at most n. Thus we have $u_{l+1}\dots u_k a \notin F$. Besides, $u_{l+1}\dots u_k a$ is not in A^*P for it is a suffix of $w \in Q$, hence $u_{l+1}\dots u_k a$ is in R as it is actually in A^*P. Thus $u_{l+1}\dots u_k a$ has a suffix in Q, that is, w has a proper suffix in Q which is again a contradiction. The theorem is proved. □

We demonstrate the completion by two examples. In the first one, Q is calculated according to the formula, while in the latter one by enumeration.

Example 3.4. Let $A=\{a,b\}$, $X=\{a^2b^2\}$. We have

$$P=\{a,a^2,a^2b\}, \quad F=\{a,a^2,a^2b,b,b^2,ab^2,ab\}$$

$$\begin{aligned} R &= (A^* - A^*P - F)\cap P^*A \\ &= (A^* - A^*\{a,a^2,a^2b\} - \{a,a^2,a^2b,b,b^2,ab^2,ab\})\cap \{a,a^2,a^2b\}^*A \\ &= (A^*bab + A^*b^2 + \{b\} + \{ab\} - \{a,a^2,a^2b,b,b^2,ab^2,ab\})\cap \{a,a^2b\}^*b \\ &= ((A^*ba + A^+b - \{ab\})\cap \{a,a^2b\}^*)b \\ &= ((A^*a^2ba + A^*a^2b)\cap \{a,a^2b\}^*)b. \end{aligned}$$

Finally,

$$Q = R - A^+R = (\{a^2ba\} + \{a^2b\})b = \{a^2bab, a^2b^2\}.$$

Example 3.5. Let $A=\{a,b,c\}$ and $X=\{ab\}$. We have $n=2$; so it is sufficient to search among the words of length not exeeding $2\times 2-2=2$ as candidates to include in Q. They are ab, ac, cb, c. There are only three possibilities $Q=\{ab,ac\}$, $\{ab,cb\}$ or $\{ab,c\}$. Only $\{ab,c\}$ is a maximal solid code.

4. Construction giving all finite maximal solid codes

4.1. Transformation

The construction to be presented involves successive transformations of a solid code that gradually lead to the desired one. Each transformation step consists essentially of removing, from a given solid code, an arbitrary word and adding subsequently an appropriate set of words having the chosen word as a proper prefix (or suffix, in the symmetric version).

Let X be a solid code and, as before, let P be the set of non-empty proper prefixes of X. Let M denote the set of the words all the suffixes of which are not in $P+X$ but all the proper prefixes of which have a suffix in P, that is

$$M(X) = P^*A - A^*(P+X).$$

It is clear that, if non-empty, M is a prefix code and every word having no suffix in P has a suffix in M. It is not difficult to see that no word in M overlaps a word in X or has a factor in X since by definition, its longest proper prefix is in P^* and none of its suffixes is in P. Moreover, if X is a maximal solid code each word in M should be a proper factor of a word in X: $M(X) \subseteq F(X)$.

When $X \subseteq A$ the proper prefixes are absent, $P = \emptyset$, therefore $M = A - A^*X = A - X$, which is empty only if $X = A$. When $X \not\subseteq A$ the set $J(X) \neq \emptyset$ and it is obvious that $J(X) \subseteq M(X)$. Thus $M(X) = \emptyset$ if and only if $X = A$.

Now provided that $M(X)$ is non-empty, that is $X \neq A$, we define the transformation p of X for an arbitrary word $x \in X$ as

$$p(X,x) = X - \{x\} + xM(X).$$

An essential property of p is that it preserves solidity.

Proposition 4.1. *If X is a solid code, $p(X,x)$ is also a solid code for every $x \in X$.*

Proof. As noted, M has no overlaps with X; therefore xM has no overlaps with $X - \{x\}$. As M has no factor in X, xM has no factor in $X - \{x\}$. These two facts also imply that xM is overlap-free. Moreover, no word of xM is a factor of the others, as M is a prefix code and no word of xM is a factor of $X - \{x\}$ as X is a solid code. Thus $p(X,x)$ is a solid code. □

We show further that p preserves maximality and finiteness.

Proposition 4.2. *If X is a maximal (finite maximal) solid code then $p(X,x)$ is also a maximal (finite maximal resp.) solid code.*

Proof. By the maximality of X, for every word $u \in A^*$, if u has no factor in $X - \{x\}$ or u is not a factor of $X - \{x\}$, or u has no overlaps with $X - \{x\}$ then there remain the following possibilities:

(i) u is a factor of x, which is a factor of xM, hence a factor of $p(X,x)$;

(ii) u overlaps x on suffix. Then obviously u overlaps every word in xM on suffix;

(iii) u overlaps x on prefix, that is, we have the equalities $u = wy, x = zw$. Then u is a suffix of $xy = zwy = zu$. If $xy \in A^*P$ then either u is a factor of P, hence of X which implies that u is a factor of x (u is not a factor of $X - \{x\}$ by assumption) and we return to (i), or u has a suffix in P which implies that u overlaps X, hence u overlaps x on suffix (u does not overlap $X - \{x\}$ by assumption) and we return to (ii). If, otherwise, $xy \notin A^*P$ then xy must have a prefix, say p, in M. We write $xy = zu = ps$, $s \in A^*$. Since $M \cap A^*(P + X) = \emptyset$ we have $|p| > |x|$ and then $|s| < |y| < |u|$ meaning that $u = ts$ for $t \in A^+$ and then $p = zt$. Consequently, $xp = xzt$ showing that u overlaps $xp \in xM$ on prefix.

(iv) Finally x is a factor of u. We write $u = yxz$ for some $y, z \in A^*$. If $z = 1$ then u has the overlap x with xM. If $z \in A^*P$ then u has (on suffix) an overlap $p \in P$ with

X, hence with x (by the assumption about u again!) and thus with xM. If $z \neq 1$ and $z \notin A^*P$ then z has a prefix in M, it follows that u has a factor in xM.

Together, points (i)–(iv) show that $p(X,x)$ is a maximal solid code. The claim about finiteness is evident. As we have noticed $M(X) \subseteq F(X)$ when X is a maximal solid code and it is finite if in addition X is finite, hence $p(X,x)$ finite. The theorem is proved. □

Remark 4.3. The transformation p is defined relative to prefixes; we might define, in a symmetric way, the transformation relative to suffixes as well. Namely, let

$$M'(X) = AS^* - (S + X)A^*$$

and if $M'(X) \neq \emptyset$, for $x \in X$

$$s(X,x) = X - \{x\} + M'(X)x.$$

By symmetry $M' \neq \emptyset$ if and only if $X \not\subseteq A$, or equivalently, $M \neq \emptyset$. Thus, both transformations p and s are applicable for all solid codes except subsets of A.

We can formulate the symmetric version of Propositions 4.1 and 4.2 on maximal solidity and finiteness for the transformation s. We have no need to simulate their proofs since the mirror image $X^\sim$ of a solid (maximal solid) code is a solid (maximal solid) code and $s(X,x)^\sim$ is nothing else but $p(X^\sim, x^\sim)$.

Example 4.4. Let $X = IJ$ with $A = I + J$ a bipartite partition of the alphabet, which is a finite maximal code (Proposition 2.5). We have $P = I$ and $M = I^*A - A^*(IJ + I) = J$. Thus for $a \in I, b \in J$ we have

$$p(X, ab) = (I - \{a\})J + I(J - \{b\}) + abJ.$$

In particular, let $A = \{a,b\}$ be a binary alphabet and $X = \{ab\}$. Then $M = \{b\}$ and $p(X,ab) = \{ab^2\}$. Put $Y = \{ab^2\}$, we can compute further $M(Y) = \{b\}$ and $p(Y, ab^2) = \{ab^3\}, \ldots$. But if we compute

$$\begin{aligned} M'(Y) &= AS(Y)^* - (S(Y) + Y)A^* \\ &= A\{b, b^2\}^* - \{b, b^2, ab^2\}A^* \\ &= \{a,b\}b^* - (bA^* + ab^2A^*) \\ &= \{a, ab\}. \end{aligned}$$

We have then

$$\begin{aligned} s(Y, ab^2) &= M'(Y)\{b^2\} \\ &= \{a, ab\}\{ab^2\} \\ &= \{a^2b^2, abab^2\}. \end{aligned}$$

This is the code in Example 3.4, up to mirror images and interchanging a and b!

A natural way to obtain new maximal solid codes is to apply, when possible, successively the transformations p or s to the given ones, in particular to the simplest ones of the form $IJ + K$. We will show that, restricted to the finite maximal case, this procedure generates *all* finite maximal solid codes. The rest of this section is devoted to this task.

4.2. Adequacy

To prove that the transformations p ans s are adequade to generate all finite solid codes, we define certain transformations inverse to p and s. For every finite maximal solid code, not the underlying alphabet, at least one of these inverses will be applicable and successive applications will lead finally to a code of the form $IJ + K$. By restoring in the reverse order the respective inverses p or s we get the sequence of application of p or s that will turn $IJ + K$ into the given maximal solid code.

Let X be a solid code. A prefix (resp. suffix) of X is *primary* if it is proper and it has no proper suffix (resp. prefix) in $P(X)$ (resp. in $S(X)$). Primary prefixes and primary suffixes are called primary factors.

We say that a primary factor is *maximal* provided it is not a proper factor of other primary factors. Not every solid code has maximal primary factors but when it is finite and the set of primary factors (being finite) is not empty then it definitely possesses a maximal primary factor and every primary factor is completed to a maximal primary factor! This fact we show to hold for finite maximal solid codes.

Lemma 4.5. *For a finite maximal solid code X containing $a^m b^n$ with the letters a, b and the positive integers m, n, one and only one of the two following conditions is fulfilled*: (i) *$a^j b^{n-1}$ is a prefix of X for some $0 < j < m$;* (ii) *$a^{m-1} b^i$ is a suffix in X for some $0 < i < n$.*

Proof. First, the two properties cannot hold simultaneously; otherwise, every word with suffix $a^{m-1} b^i$ overlaps one with prefix $a^j b^{n-1}$.

To show the existence, consider, for arbitrary $a \in I(X)$ and $b \in J(X)$, a word

$$w = a^{m-1} b^{n-1} a^{m-1} b^{n-1} \cdots a^{m-1} b^{n-1}$$

long enough not to be a factor of X. The following situations need to be considered: There is $x \in X$ so that

(i) w overlaps x on prefix: $w = ur, x = su$, $r, s, u \in A^+$. Since $a \notin J$, u must terminate on a letter b, so u, and therefore x, has a suffix $a^{m-1} b^j$ for some $0 < j < n$.

(ii) w overlaps x on suffix. A similar argument shows that x has a prefix $a^i b^{n-1}$ for some $0 < i < m$.

(iii) The remaining possibility that x is a factor of w is ruled out. In fact, x has then the form $a^j b^{n-1} v a^{m-1} b^i$ for $0 < j < n, 0 < i < m$ and $v \in A^*$, which is a contradiction since x overlaps itself. The lemma is proved. □

Proposition 4.6. *For every finite maximal solid code the set of primary factors is non-empty*; *therefore*, *maximal primary factors always exist.*

Proof. Let X be a finite maximal solid code containing the word $a^m b^n$ for $a \in I, b \in J$, $m, n > 0$. If $m = 1$ then $X = \{ab^n\}$ is a maximal solid code (Example 2.2(ii)) for which $a, ab, \ldots, ab^{n-1}$ are all primary prefixes. Similarly for the case of $n = 1$. If $m, n > 1$ it is easy to see that either $a^m b^{n-1}$ is a primary prefix or $a^{m-1} b^n$ is a primary suffix depending on which of (i) or (ii) in Lemma 4.1 holds. The proof is complete. □

The primary factors have the following basic property.

Proposition 4.7. *Let X be a solid code and p be a maximal primary factor which is a primary prefix of X. Then, if p is a factor of a word x in X, it is none but a prefix of x. Exactly, for every maximal primary prefix p: $A^* pA^* \cap X = pA^+ \cap X$ and $A^+ pA^* \cap X = \emptyset$.*

Proof. We write $x = upw$ for $u, w \in A^*$ and we show that $u = 1$. Suppose that $u \neq 1$ then $pw \in S$. Obviously w is a non-empty word to avoid overlapping in X. Therefore $pA^+ \cap S \neq \emptyset$. Let v be a shortest word satisfying $pv \in S$. Then pv must be a primary suffix since, for every proper prefix f of pv, the inequality $|p| < |f|$ implies $f \notin S$ by the minimality of $|v|$, and $|f| \leqslant |p|$ implies $f \notin S$ by the solidity of X. But this fact contradicts the assumption that p is a maximal primary factor. So we have $u = 1$ as required. □

Let X' be an arbitrary solid code possessing a maximal primary factor, say a prefix p. We define the transformation p^* as

$$p^*(X', p) = X' - (pA^+ \cap X') + \{p\}.$$

We expect $p^*(X', p)$ to be a solid code, finite solid code and maximal solid code whenever X' is such a code. This is indeed the case.

Theorem 4.8. *Let X' be a (finite, finite maximal) solid code with a maximal primary prefix p. If p is not a letter then $p^*(X', p)$ is also a (finite, finite maximal) solid code.*

Proof. Put $X = p^*(X', p) = X' - (pA^+ \cap X') + \{p\}$. Indeed p is in X and is not a factor of $X' - (pA^+ \cap X')$ by the preceding proposition. Moreover, p overlaps $X' - (pA^+ \cap X')$ neither on suffix since p is a primary prefix nor on prefix since X' is solid. Thus X is a solid code. The fact that X is finite when X' is finite is evident.

Suppose now that X' is a finite maximal solid code. The existence of a primary factor ensures that $X' \neq A$, hence $I(X') \neq \emptyset$ and $J(X') \neq \emptyset$. For an arbitrary word $w \in A^*$ consider the word $a^n w b^n$ with n arbitrarily large and $a \in I(X')$, $b \in J(X')$. By Proposition 2.9, $a^n w b^n$ admits a factor $x \in X'$. We distinguish three cases.

(i) $x = a^i u$, where u is a prefix of w and $i > 0$. If $x \notin pA^+$ then $x \in X$ that means w overlaps $x \in X$ on prefix. If $x \in pA^+$, as p is a primary prefix which is not a letter, it follows that $p \notin a^*$, hence p overlaps u and w, on suffix.

(ii) $x = vb^i$ for some $i > 0$ and suffix v of w. In this case w overlaps $X' - pA^+$ on suffix (if $x \notin pA^+$) or overlaps p on suffix or admits p as factor (if $x \in pA^+$).

(iii) x is a factor of w. Then w has a factor in $X' - pA^+ \subseteq X$ or in pA^+, hence x has p as a factor.

Summarizing, these possibilities show that X is a maximal solid code. The theorem is proved. □

The transformation p^* is inverse to p in the following sense.

Theorem 4.9. *Let X' be a finite maximal solid code with a maximal primary prefix p. If p is not a letter then $p(p^*(X', p), p) = X'$.*

Proof. Denote $X = p^*(X', p) = X' - (pA^+ \cap X') + \{p\}$. Then indeed $p \in X$ and X is a finite maximal solid code.

We show that $pM(X) = pA^+ \cap X'$ which implies $p(X, p) = X + pM(X) - \{p\} = X' - (pA^+ \cap X') + \{p\} + (pA^+ \cap X') - \{p\} = X'$, which is what we have to prove. Put $D = \{u : pu \in X'\}$. We first show that $D \subseteq M(X)$. Since X' is solid, every $u \in D$ has no suffix in $P(X')$, in particular, in $P(X) + X = \{p\} + P(\{p\}) + P(X' - pA^+) \subseteq P(X')$ that is $u \in A^* - A^*(P(X) + X)$. On the other hand, let $u = va$ for $a \in A$, $v \in A^*$. Then $pv \in P(X')$. If $v = 1$ we have $u = a \in A \subseteq P(X)^* A$. If $v \neq 1$, pv is not a primary prefix by the maximal primarity of p, hence $pv \in A^+ P(X')$; this yields $v \in A^* P(X')$ by the primarity of p. The same argument shows that $v \in P(X')^+$ and $u \in P(X')^+ A$. Now v, as a factor which is not a prefix of $pu \in X'$, has no occurrence of p as factor. By Proposition 4.7, we get $v \in (P(X') - A^* pA^*)^+ = P(X' - pA^+) + P(\{p\})^+ = P(X)^+$. We conclude that $u \in P(X)^* A - A^*(P(X) + X) = M(X)$, that is $D \subseteq M$. Thus $X' \cap pA^+ = pD \subseteq pM(X)$.

Conversely, we prove $pM(X) \subseteq X'$. The fact that every word u of M has no factor in $X = \{p\} + (X' - \{pA^+\})$ and no suffixes in $P(X) = P(\{p\}) + P(X' - pA^+)$ and that p, being a word of X and a primary prefix of X', has no proper suffix in $P(X') + X'$, imply that pu has no proper suffix in $P(X')$, or which amounts to the same, pu does not overlap X' on suffix and does not have a factor in X'. On the other hand, as $P(X) \subseteq P(X')$, the word pu, being in $pP(X)^* A \subseteq P(X')^+ A$ evidently does not overlap X' on prefix. Therefore by the maximal solidity of X' it is a factor of X': $wpuv = x \in X'$, $w, v \in A^*$. By virtue of Proposition 4.7 $w = 1$. If $v \neq 1$ then $pu \in P(X')$; but the fact that pu has no proper suffix in $P(X')$ shows that pu is a primary prefix, which indeed is a contradiction with p being a maximal primary prefix of X'. Thus we have $v = 1$ and $pu \in X'$ yielding $pM \subseteq X' \cap pA^+$. This concludes the proof. □

Remark 4.10. If the choosen primary factor of X' happens to be a suffix, say s, then we can prove, of course, the symmetric version of the Proposition 4.3 saying that

$A^*sA^* \cap X' = A^+s \cap X'$, as well as define the transformation

$$s^*(X', s) = X' - (A^+s \cap X') + \{s\}$$

for which the symmetric versions of Theorems 4.8 and 4.9 holds.

Now, consider an arbitrary finite maximal solid code X', to which we apply the transformation p^* or s^* if it possesses non-literal primary factors to get a new finite maximal solid code. We repeat the procedure as long as there has been non-literal primary factors for each step. As those words affected by the transformation result in a shorter word, after a finite number of steps the resulting finite maximal solid code inevitably has all its maximal primary factors literal and the procedure halts. What are such codes? The answer is very much in sight.

Proposition 4.11. *A finite maximal solid code has all primary factors literal if and only if it is of the form $IJ + K$ for a partition (I,J,K) of the alphabet.*

Proof. The "if" direction is straightforward: every maximal solid code $IJ + K$ has $P = I$ and $S = J$ comprising all of its primary factors. Indeed they are all letters.

For the converse, let the finite maximal solid code X have all primary factors in A. For every $a \in I$ and $b \in J$ there exist the integers $m, n > 0$ such that $a^m b^n \in X$ (Proposition 2.7). If $m > 1, n > 1$ there exists $x \in X$ with a prefix $a^j b^{n-1}$, $0 < j < m$ (Lemma 4.5). Therefore the proper suffix $a^j b^n$ is evidently primary, since all words of X do not overlap $a^j b^n$, to avoid overlapping x. But $a^j b^n$ is not literal, a contradiction. So we have at least one of m, n equal to 1, let $m = 1$. If still $n \neq 1$, it is easy to see that the proper prefixes $ab, \ldots, ab^{n-1}$ are all primary which are not literal, again a contradition. Thus we have $m = n = 1$ and $ab \in X$ which shows that $IJ + K \subseteq X$. As both of them are maximal solid codes, $X = IJ + K$, the theorem is proved. □

Finally, we are in a position to state the concluding result.

Theorem 4.12. *The transfomations p and s are sufficient to generate all finite maximal solid codes by successive application starting from the codes $IJ + K$ for arbitrary partitions (I,J,K) of the underlying alphabet and neither of them alone is sufficient.*

Proof. Let X' be a finite maximal solid code and $t_n^*, t_{n-1}^*, \ldots, t_1^*$, where $t_i = p$ or s, be a sequence of the transformations the successive application of which leads from X' to a code X of the form $IJ + K$:

$$X = t_1^*(t_2^*(\ldots t_n^*(X')\ldots)).$$

Now the application of the transformations t_i in the reverse order $t_1, t_2, \ldots, t_n$, by Theorem 4.5, will bring X to X':

$$X' = t_n(t_{n-1}(\ldots t_1(X)\ldots)).$$

For the latter claim, consider concrete examples. Let $A=\{a,b\}$ and $X'=\{ab^n\}, n\geqslant 2$. The codes $\{ab\}$ and $\{ba\}$ are the only finite maximal solid codes of the form $IJ+K$. The one-element $\{ab^n\}$ with $n>1$ is in no way to be reached from $\{ab\}$ or $\{ba\}$ by the application of s alone, since s produces the result by juxtaposing the set M' to the left of the given solid code. Symmetrically, the code $X^{\sim}=\{a^nb\}, n\geqslant 2$ shows the insufficiency of p alone. The theorem is proved. □

Acknowledgements

I am grateful to an anonymous referee for very careful reading and positive comments, and to H. Jürgensen for remarks that improved the presentation. I appreciate very much the help of M. Ito, S. Konstantinidis and S.S. Yu who provided valuable information. I thank N.X. My for bringing solid codes to my attention.

References

[1] J. Berstel, D. Perrin, Theory of Codes, Academic Press, Orlando, 1985.
[2] V. Bruyère, L. Wang, L. Zhang, On Completion of Codes with Finite Deciphering Delay, European J. Combin. 11 (1990) 513–521.
[3] Y. Césari, Sur un algorithme donnant les codes bipréfixes finis, Math. Systems Theory 6 (1982) 221–225.
[4] A. Ehrenfeucht, G. Rozenberg, Each Regular Code Is Included in a Regular Maximal Code, RAIRO Informatique Théorique 20 (1986) 89–96.
[5] H. Jürgensen, M. Katsura, S. Konstantinidis, Maximal solid codes, J. Automata Combin. Languages, to appear.
[6] H. Jürgensen, S. Konstantinidis, Codes, in: G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Vol. 1, Springer, Berlin, 1997, pp. 511–607.
[7] H. Jürgensen, S.S. Yu, Solid codes, J. Inform. Process. Cybernet. EIK 26 (1990) 563–574.
[8] V.I. Levenshtein, Maximum number of words in codes without overlaps, Problemy Peredachi Informatsii 6 (4) (1970) 88–90 (in Russian) (English translation: Problems Inform. Transmission 6 (4) (1973) 355–357).
[9] D. Perrin, Completing biprefix codes, Theoret. Comput. Sci. 28 (1984) 329–336.
[10] O.T. Romanov, Invariant decoding automata without look-ahead, Problemy Kibernetiki 17 (1966) 233–236 (in Russian).
[11] H.J. Shyr, Free Monoids and Languages, Lecture Notes, Hon Min Book Company, Taichung, 1991.
[12] H.J. Shyr, S.S. Yu, Solid codes and disjunctive domains, Semigroup Forum 41 (1990) 23–37.
[13] L. Zhang, Z. Shen, Completion of recognizable bifix codes, Theoret. Comput. Sci. 145 (1995) 345–355.