Vietnam Journal of Mathematics **33**:2 (2005) 199–206

Vietnam Journal of MATHEMATICS

An Embedding Algorithm for Supercodes and Sucypercodes

Kieu Van Hung and Nguyen Quy Khang

Hanoi Pedagogical University No. 2, Phuc Yen, Vinh Phuc, Vietnam

Received July 21, 2004
Revised October 15, 2004

Abstract. Supercodes and sucypercodes, particular cases of hypercodes, have been introduced and considered by D. L. Van and the first author of this paper. In particular, it has been proved that, for such classes of codes, the embedding problem has positive solution. Our aim in this paper is to propose another embedding algorithm which, in some sense, is simpler than those obtained earlier.

1. Preliminaries

Hypercodes, a special kind of prefix codes (suffix codes), are subject of many research works (see [7, 8] and the papers cited there). They have some interesting properties. In particular, every hypercode over a finite alphabet is finite (see [7]).

Supercodes and sucypercodes, particular cases of hypercodes, have been introduced and considered in [2, 3, 9 - 11]. In particular, supercodes were introduced and studied in depth by D. L. Van [9].

For a given class C of codes, a natural question is whether every code X satisfying some property $\mathfrak{p}$ (usually, the finiteness or the regularity) is included in a code Y maximal in C which still has the property $\mathfrak{p}$. This problem, which we call the *embedding problem* for the class C, attracts a lot of attention. Unfortunately, this problem was solved only for several cases by means of different combinatorial techniques (see [10]).

The embedding problem for supercodes and sucypercodes was solved positively by applying the general embedding schema of Van [9, 10]. Moreover, an effective embedding algorithm for supercodes over two-letter alphabets, was also proposed [9].

In this paper we propose embedding algorithms for these kinds of codes other than those obtained earlier. It is worthy to note that this method allows us to obtain similar embedding algorithms for r_n-supercodes and r_n- sucypercodes.

We now recall some notions, notations and facts, which will be used in the sequel. Let A be a finite alphabet and A^* the set of all the words over A. The empty word is denoted by 1 and A^+ stands for $A^* - 1$. The number of all occurrences of letters in a word u is the *length* of u, denoted by $|u|$.

A language over A is a subset of A^*. A language X is a *code* over A if for all $n, m \geq 1$ and $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$, the condition

$$x_1 x_2 \ldots x_n = y_1 y_2 \ldots y_m,$$

implies $n = m$ and $x_i = y_i$ for $i = 1, \ldots, n$. A code X is *maximal* over A if X is not properly contained in any other code over A. Let C be a class of codes over A and $X \in C$. The code X is *maximal in* C (not necessarily maximal as a code) if X is not properly contained in any other code in C. For further details of the theory of codes we refer to [1, 5, 7].

An *infix* (i.e. *factor*) of a word v is a word u such that $v = xuy$ for some $x, y \in A^*$; the infix is *proper* if $xy \neq 1$. A subset X of A^+ is an *infix code* if no word in X is a proper infix of another word in X.

Let $u, v \in A^*$. We say that a word u is a *subword* of v if, for some $n \geq 1, u = u_1 \ldots u_n, v = x_0 u_1 x_1 \ldots u_n x_n$ with $u_1, \ldots, u_n, x_0, \ldots, x_n \in A^*$. If $x_0 \ldots x_n \neq 1$ then u is called a *proper subword* of v. A subset X of A^+ is a *hypercode* if no word in X is a proper subword of another word in it. The class C_h of hypercodes is evidently a subclass of the class C_i of infix codes. For more details about infix codes and hypercodes, see [4, 6 - 8].

Given $u, v \in A^*$. The word u is called a *permutation* of v if $|u|_a = |v|_a$ for all $a \in A$, where $|u|_a$ denotes the number of occurrences of a in u. And u is a *cyclic permutation* of v if there exist words x, y such that $u = xy$ and $v = yx$. We shall denote by $\pi(v)$ and $\sigma(v)$ the sets of all permutations and cyclic permutations of v, respectively.

Definition 1.1. *A subset X of A^+ is a supercode (sucypercode) over A if no word in X is a proper* **su***bword of a* **per***mutation (***cy***clic* **per***mutation, resp.) of another word in it. Denote by C_{sp} and C_{scp} the classes of all supercodes and sucypercodes over A, respectively.*

Thus, every supercode is a sucypercode and every sucypercode is a hypercode. Hence, all supercodes and sucypercodes are finite (see [10]).

Example 1.2.
(i) Every uniform code over A which is a subset of A^k, $k \geq 1$, is a supercode and a sucypercode over A.
(ii) Consider the subset $X = \{ab, b^2a\}$ over $A = \{a, b\}$. Since ab is not a proper subword of b^2a, X is a hypercode. But X is not a sucypercode, because ab is a proper subword of ab^2, a cyclic permutation of b^2a.
(iii) The $Y = \{abab, a^2b^3\}$ over $A = \{a, b\}$ is a sucypercode, because $abab$ is not a proper subword of any word in $\sigma(a^2b^3) = \{a^2b^3, ba^2b^2, b^2a^2b, b^3a^2, ab^3a\}$. As

$abab$ is a proper subword of the permutation $abab^2$ of a^2b^3, we have Y is not a supercode.

For any set X we denote by $\mathcal{P}(X)$ the family of all subsets of X. Recall that a *substitution* is a mapping f from B into $\mathcal{P}(C^*)$, where B and C are alphabets. If $f(b)$ is regular for all $b \in B$ then f is called a *regular substitution*. When $f(b)$ is a singleton for all $b \in B$ it induces a *homomorphism* from B^* into C^*. Let # be a new letter not being in A. Put $A_\# = A \cup \{\#\}$. Let us consider the regular substitutions S_1, S_2 and the homomorphism h defined as follows

$$S_1 : A \to \mathcal{P}(A_\#^*), \text{ where } S_1(a) = \{a, \#\} \text{ for all } a \in A;$$
$$S_2 : A_\# \to \mathcal{P}(A^*), \text{ with } S_2(\#) = A^+ \text{ and } S_2(a) = \{a\} \text{ for all } a \in A;$$
$$h : A_\#^* \to A^*, \text{ with } h(\#) = 1 \text{ and } h(a) = a \text{ for all } a \in A.$$

Actually, the substitution S_1 is used to mark the occurrences of letters to be deleted from a word. The homomorphism h realizes the deletion by replacing # by empty word. The inverse homomorphism h^{-1} "chooses" in a word the positions where the words of A^+ inserted, while S_2 realizes the insertions by replacing # by A^+.

Denote by $A^{[n]}$ the set of all the words in A^* whose length is less than or equal to n. For every subset X of A^*, we denote $XA^- = X(A^+)^{-1} = \{w \in A^* \mid wy \in X, y \in A^+\}$, $A^-X = (A^+)^{-1}X = \{w \in A^* \mid yw \in X, y \in A^+\}$ and $A^-XA^- = (A^+)^{-1}X(A^+)^{-1}$. The following result has been proved in [10] (see also [2]).

Theorem 1.3. *The embedding problem has positive answer in the finite case for every class C_α of codes, $\alpha \in \{i, h, scp, sp\}$. More precisely, every finite code X in C_α, with $\max X = n$, is included in a code Y which is maximal in C_α and remains finite with $\max Y = \max X$. Namely, Y can be computed by the following formulas according to the case.*

(i) *For infix codes*

$$Y = Z - (ZA^+ \cup A^+Z \cup A^+ZA^+) \cap A^{[n]},$$

where $Z = A^{[n]} - F - (XA^+ \cup A^+X \cup A^+XA^+) \cap A^{[n]}$ and $F = XA^- \cup A^-X \cup A^-XA^-$.

(ii) *For hypercodes*

$$Y = Z - S_2(h^{-1}(Z) \cap (A_\#^*\{\#\}A_\#^*) \cap A_\#^{[n]}) \cap A^{[n]},$$

where $Z = A^{[n]} - h(S_1(X) \cap (A_\#^*\{\#\}A_\#^*)) - S_2(h^{-1}(X) \cap (A_\#^*\{\#\}A_\#^*) \cap A_\#^{[n]}) \cap A^{[n]}$.

(iii) *For sucypercodes*

$$Y = Z - \sigma(S_2(h^{-1}(Z) \cap (A_\#^*\{\#\}A_\#^*) \cap A_\#^{[n]}) \cap A^{[n]}),$$

where $Z = A^{[n]} - h(S_1(\sigma(X)) \cap (A_\#^*\{\#\}A_\#^*)) - \sigma(S_2(h^{-1}(X) \cap (A_\#^*\{\#\}A_\#^*) \cap A_\#^{[n]}) \cap A^{[n]})$.

(iv) *For supercodes*

$$Y = Z - \pi(S_2(h^{-1}(Z) \cap (A^*_\#\{\#\}A^*_\#) \cap A^{[n]}_\#) \cap A^{[n]}),$$

where $Z = A^{[n]} - h(S_1(\pi(X)) \cap (A^*_\#\{\#\}A^*_\#)) - \pi(S_2(h^{-1}(X) \cap (A^*_\#\{\#\}A^*_\#) \cap A^{[n]}_\#) \cap A^{[n]})$.

2. Embedding Algorithms

We propose in this section embedding algorithms for supercodes and sucypercodes. These algorithms use only the permutation π or the cyclic permutation σ at the last step. Particularly, an effective algorithm for supercodes over two-letter alphabets is established.

Let A be a finite, totally ordered alphabet, and let $\sim$ be an equivalence relation on A^*. For every $[w]$ of $A^*/\sim$, we denote by w_0 the lexicographically minimal word of $[w]$. On A^*, we introduce two equivalence relations $\sim_\pi$ and $\sim_\sigma$ defined by

$$u \sim_\pi v \Leftrightarrow \forall a \in A : |u|_a = |v|_a,$$
$$u \sim_\sigma v \Leftrightarrow \exists x, y \in A^* : u = xy, v = yx.$$

We denote by $A^*_\pi = \{w_0 \in [w] \mid [w] \in A^*/\sim_\pi\}$ and $A^*_\sigma = \{w_0 \in [w] \mid [w] \in A^*/\sim_\sigma\}$.

Let $\rho \in \{\pi, \sigma\}$. A subset X of A^*_ρ is called an *infix code* (a *hypercode*) *on* A^*_ρ if it is an infix code (resp., a hypercode) over A. Denote by $C_{i|A^*_\rho}$ and $C_{h|A^*_\rho}$ the sets of all infix codes and hypercodes on A^*_ρ, respectively.

Lemma 2.1. *If* $|A| = 2$ *then* $C_{h|A^*_\pi} = C_{i|A^*_\pi}$.

Proof. Since $C_{h|A^*_\pi} \subseteq C_{i|A^*_\pi}$ is trivial, it suffices to show that $C_{i|A^*_\pi} \subseteq C_{h|A^*_\pi}$. Suppose the contrary that there exists $X \in C_{i|A^*_\pi}$ but $X \notin C_{h|A^*_\pi}$. Let $A = \{a, b\}$. Then, for all w in A^*_π, w has the form $w = a^m b^n$ with $m, n \geq 0$. Since $X \notin C_{h|A^*_\pi}$, it follows that there exist $u, v \in X$ such that $u \prec_h v$. Therefore, $u = a^m b^n$, $v = a^k b^\ell$ with $0 \leq m \leq k$, $0 \leq n \leq \ell$ and $m + n < k + \ell$. Hence $u \prec_i v$, which contradicts $X \in C_{i|A^*_\pi}$. Thus, $C_{i|A^*_\pi} \subseteq C_{h|A^*_\pi}$. ■

From the fact that every hypercode is finite and from Lemma 2.1, it follows that all the infix codes on A^*_π with $|A| = 2$, are finite.

We now consider two maps $\lambda_\pi : A^* \to A^*_\pi$, $\lambda_\pi(w) = w_0$ and $\lambda_\sigma : A^* \to A^*_\sigma$, $\lambda_\sigma(w) = w_0$. The following result establishes relationship between supercodes and sucypercodes with the images of them with respect to the maps λ_π and λ_σ.

Theorem 2.2. *For any* $X \subseteq A^+$, *we have the following assertions*

(i) $X \in C_{sp} \Leftrightarrow \lambda_\pi(X) \in C_{h|A^*_\pi}$. *Particularly, if* $|A| = 2$ *then* $X \in C_{sp} \Leftrightarrow \lambda_\pi(X) \in C_{i|A^*_\pi}$.

(ii) $X \in C_{scp} \Leftrightarrow \lambda_\sigma(X) \in C_{h|A^*_\sigma}$.

Proof. We treat only the item (i). For the item (ii) the argument is similar. Let $X \in C_{sp}$ but $\lambda_\pi(X) \notin C_{h|A_\pi^*}$. Then, there exist $u_0, v_0 \in \lambda_\pi(X)$ such that $u_0 \prec_h v_0$. Since $u_0, v_0 \in \lambda_\pi(X)$, there are $u, v \in X$ satisfying $u \in \pi(u_0), v \in \pi(v_0)$. Hence, from $u_0 \prec_h v_0$ it follows that $u \prec_{sp} v$, which contradicts the fact that $X \in C_{sp}$. Thus, $\lambda_\pi(X) \in C_{h|A_\pi^*}$. Conversely, suppose that $\lambda_\pi(X) \in C_{h|A_\pi^*}$. If $X \notin C_{sp}$, i.e. $\exists u, v \in X$: $u \prec_{sp} v$, then $\lambda_\pi(u) \prec_h \lambda_\pi(v)$, a contradiction. So, $X \in C_{sp}$.

If $|A| = 2$ then, by Lemma 2.1, $C_{h|A_\pi^*} = C_{i|A_\pi^*}$. Therefore, by the above, $X \in C_{sp} \Leftrightarrow \lambda_\pi(X) \in C_{h|A_\pi^*} \Leftrightarrow \lambda_\pi(X) \in C_{i|A_\pi^*}$. ∎

An infix code (a hypercode) X on A_π^* (resp., A_σ^*) is *maximal on* A_π^* (resp., A_σ^*) if it is not properly contained in any one on A_π^* (resp., A_σ^*). The following assertion establishes relationship between maximal hypercodes on A_π^* (resp., A_σ^*) and maximal supercodes (resp., sucypercodes) over A.

Theorem 2.3. *For any $X \subseteq A^+$, we have the following*

(i) *If X is a maximal hypercode on A_π^* then $\pi(X)$ is a maximal supercode over A. In particular, if $|A| = 2$ and X is a maximal infix code on A_π^* then $\pi(X)$ is a maximal supercode over A.*

(ii) *If X is a maximal hypercode on A_σ^* then $\sigma(X)$ is a maximal sucypercode over A.*

Proof. We prove only the item (i). For the remaining item the argument is similar. Let X be a maximal hypercode on A_π^*. By definition, $\pi(X)$ is a supercode over A. If $\pi(X)$ is not a maximal supercode over A then there exist $u, v \in \pi(X)$ such that $u \prec_{sp} v$. Then $\lambda_\pi(u), \lambda_\pi(v) \in X$ and $\lambda_\pi(u) \prec_h \lambda_\pi(v)$, a contradiction. Thus, $\pi(X)$ must be a maximal supercode over A.

For the case $|A| = 2$, the assertion follows immediately from the above and Lemma 2.1. ∎

Denote by $A_\rho^{[n]}$, $\rho \in \{\pi, \sigma\}$, the set of all the words in A_ρ^* whose length is less than or equal to n. For every X of A_π^*, we denote $XA_\pi^- = X(A_\pi^+)^{-1}$, $A_\pi^- X = (A_\pi^+)^{-1}X$ and $A_\pi^- X A_\pi^- = (A_\pi^+)^{-1}X(A_\pi^+)^{-1}$. As a consequence of Theorem 1.3 we have

Theorem 2.4. *The following assertions are true*

(i) *Let $A = \{a, b\}$ and let $X \in C_{i|A_\pi^*}$ with $\max X = n$. Then, there exists a maximal infix code Y on A_π^* with $\max X = \max Y$ which can be computed by the formulas*

$$Y = Z - (Zb^+ \cup a^+Z \cup a^+Zb^+) \cap A_\pi^{[n]},$$

where $Z = A_\pi^{[n]} - F - (Xb^+ \cup a^+X \cup a^+Xb^+) \cap A_\pi^{[n]}$ and $F = XA_\pi^- \cup A_\pi^- X \cup A_\pi^- X A_\pi^-$.

(ii) *Let $\rho \in \{\pi, \sigma\}$ and let $X \in C_{h|A_\rho^*}$ with $\max X = n$. Then, there exists a maximal hypercode Y on A_ρ^* with $\max X = \max Y$ which can be computed by the formulas*

$$Y = Z - S_2(h^{-1}(Z) \cap (A_{\#}^*\{\#\}A_{\#}^*) \cap A_{\#}^{[n]}) \cap A_{\rho}^{[n]},$$

where $Z = A_\rho^{[n]} - h(S_1(X) \cap (A_{\#}^*\{\#\}A_{\#}^*)) \cap A_\rho^{[n]} - S_2(h^{-1}(X) \cap (A_{\#}^*\{\#\}A_{\#}^*) \cap A_{\#}^{[n]}) \cap A_\rho^{[n]}$.

Proof. It follows immediately from Theorem 1.3(i) and (ii) with the notice that $A_\pi^* = a^*b^*$, where $A = \{a, b\}$. ■

By virtue of Theorems 2.2, 2.3 and 2.4, embedding algorithms for supercodes and sucypercodes can be presented as follows.

Algorithm SP

Input: A supercode X over A with $\max X = n$.
Output: A maximal supercode Y over A containing X, with $\max Y = n$.

1. Finding $X' = \lambda_\pi(X)$. By Theorem 2.2(i), X' is a hypercode on A_π^*. In particular, X' is an infix code on A_π^*, if $|A| = 2$.
2. We compute a maximal infix code (hypercode) Y' on A_π^* which contains X' by the formulas in Theorem 2.4(i) or (ii). Then, by Theorem 2.3(i), $Y = \pi(Y')$ is a maximal supercode over A. The set Y contains X because $X \subseteq \pi(X') \subseteq \pi(Y') = Y$.

Algorithm SCP

Input: A sucypercode X over A with $\max X = n$.
Output: A maximal sucypercode Y over A containing X, with $\max Y = n$.

1. Finding $X' = \lambda_\sigma(X)$. By Theorem 2.2(ii), X' is a hypercode on A_σ^*.
2. We compute a maximal hypercode Y' on A_σ^* which contains X' by the formulas in Theorem 2.4(ii). Then, by Theorem 2.3(ii), $Y = \sigma(Y')$ is a maximal sucypercode over A. The set Y contains X because $X \subseteq \sigma(X') \subseteq \sigma(Y') = Y$.

3. Examples

In this section, we consider some examples by applying the above embedding algorithms.

Example 3.1. Consider the supercode $X = \{a^2b^2ab^2, a^3ba^2b, b^4ab^3\}$ over the alphabet $A = \{a, b\}$ with $\max X = 8$. By Algorithm SP, we may compute a maximal supercode Y over A which contains X as follows

1. We have $X' = \lambda_\pi(X) = \{a^3b^4, a^5b^2, ab^7\}$ is an infix code on $A_\pi^* = a^*b^*$.

2. Since $\max X' = 8$, we can compute a maximal infix code Y' on A_π^* which contains X' by the formulas in Theorem 2.4(i) with $n = 8$. We shall do it now step by step.

$X'A_\pi^- = \{1, a, a^2, ab, a^3, ab^2, a^4, a^3b, ab^3, a^5, a^3b^2, ab^4, a^5b, ba^5, a^3b^3, ab^6\};$

$A_\pi^- X' = \{1, b, b^2, ab^2, b^3, a^2b^2, b^4, a^3b^2, ab^4, b^5, a^4b^2, a^2b^4, b^6, b^7\}$;

$A_\pi^- X' A_\pi^- = \{1, a, b, a^2, ab, b^2, a^3, a^2b, ab^2, b^3, a^4, a^3b, a^2b^2, ab^3, b^4, a^4b, a^2b^3, b^5, b^6\}$;

$F = X'A_\pi^- \cup A_\pi^- X' \cup A_\pi^- X' A_\pi^- = \{1, a, b, a^2, ab, b^2, a^3, a^2b, ab^2, b^3, a^4, a^3b, a^2b^2, ab^3, b^4, a^5, a^4b, a^3b^2, a^2b^3, ab^4, b^5, a^5b, a^4b^2, a^3b^3, a^2b^4, ba^5, b^6, ab^6, b^7\}$;

$(X'b^+ \cup a^+X' \cup a^+X'b^+) \cap A_\pi^{[8]} = \{a^6b^2, a^5b^3, a^4b^4, a^3b^5\}$;

$Z = A_\pi^{[8]} - F - \{a^6b^2, a^5b^3, a^4b^4, a^3b^5\} = \{a^6, a^7, a^6b, a^5b^2, a^4b^3, a^3b^4, a^2b^5, a^8, a^7b, a^2b^6, ab^7, b^8\}$;

$(Zb^+ \cup a^+Z \cup a^+Zb^+) \cap A_\pi^{[8]} = \{a^7, a^6b, a^8, a^7b, a^6b^2, a^5b^3, a^4b^4, a^3b^5, a^2b^6\}$;

$Y' = \{a^6, a^5b^2, a^4b^3, a^3b^4, a^2b^5, ab^7, b^8\}$.

So, $Y = \pi(\{a^6, a^5b^2, a^4b^3, a^3b^4, a^2b^6, ab^7, b^8\})$ is a maximal supercode over A containing X.

Example 3.2. Let us consider the language $X = \{acb, a^2b^2, cabc\}$ over the alphabet $A = \{a, b, c\}$. It is not difficult to check that this language is a sucypercode, not being a supercode. By Algorithm SCP, we can compute a maximal sucypercode Y over A containing X as follows

1. We have $X' = \lambda_\sigma(X) = \{acb, a^2b^2, abc^2\}$ which is a hypercode on A_σ^*.

2. Since $\max X' = 4$, we may compute a maximal hypercode Y' on A_σ^* which contains X' by the formulas in Theorem 2.4(ii) as follows

$S_1(X') \cap (A_\#^*\{\#\}A_\#^*) = \{\#cb, a\#b, ac\#, \#^2b, \#c\#, a\#^2, \#^3, \#ab^2, a\#b^2, a^2\#b, a^2b\#, \#^2b^2, \#a\#b, a\#^2b, \#ab\#, a\#b\#, a^2\#^2, \#^3b, \#^2b\#, \#a\#^2, a\#^3, \#^4, \#bc^2, a\#c^2, ab\#c, abc\#, \#^2c^2, \#b\#c, a\#^2c, \#bc\#, a\#c\#, ab\#^2, \#^3c, \#^2c\#, \#b\#^2\}$;

$h(S_1(X') \cap (A_\#^*\{\#\}A_\#^*)) \cap A_\sigma^{[4]} = \{1, a, b, c, a^2, ab, ac, b^2, bc, c^2, a^2b, ab^2, abc, ac^2, bc^2\}$;

$h^{-1}(X') \cap (A_\#^*\{\#\}A_\#^*) \cap A_\#^{[4]} = \{\#acb, acb\#, ac\#b, a\#cb\}$;

$S_2(h^{-1}(X') \cap (A_\#^*\{\#\}A_\#^*) \cap A_\#^{[4]}) \cap A_\sigma^{[4]} = \{a^2cb, acb^2, acbc, ac^2b, abcb\}$;

$Z = \{a^3, a^2c, acb, b^3, b^2c, c^3, a^4, a^3b, a^3c, a^2b^2, a^2bc, a^2c^2, abab, abac, ab^3, ab^2c, abc^2, acac, ac^3, b^4, b^3c, b^2c^2, bcbc, bc^3, c^4\}$;

$h^{-1}(Z) \cap (A_\#^*\{\#\}A_\#^*) \cap A_\#^{[4]} = \{\#a^3, a^3\#, a^2\#a, a\#a^2, \#a^2c, a^2c\#, a^2\#c, a\#ac, \#acb, acb\#, ac\#b, a\#cb, \#b^3, b^3\#, b^2\#b, b\#b^2, \#b^2c, b^2c\#, b^2\#c, b\#bc, \#c^3, c^3\#, c^2\#c, c\#c^2\}$;

$S_2(h^{-1}(Z) \cap (A_\#^*\{\#\}A_\#^*) \cap A_\#^{[4]}) \cap A_\sigma^{[4]} = \{a^4, a^3b, a^3c, a^2cb, a^2c^2, a^2bc, abac, acac, acb^2, acbc, ac^2b, abcb, ab^3, b^4, b^3c, ab^2c, b^2c^2, bcbc, ac^3, bc^3, c^4\}$;

$Y' = \{a^3, a^2c, acb, b^3, b^2c, c^3, a^2b^2, abab, abc^2\}$.

Thus, $Y = \sigma(\{a^3, a^2c, acb, b^3, b^2c, c^3, a^2b^2, abab, abc^2\})$ is a maximal sucypercode over A which contains X.

Acknowledgement. The authors would like to thank his colleagues in the seminar Mathematical Foundation of Computer Science at Hanoi Institute of Mathematics for their useful discussions and attention to the work. Especially, the authors are indebted to Profs. Do Long Van and Phan Trung Huy for their kind help.

References

1. J. Berstel and D. Perrin, *Theory of Codes*, Academic Press, New York, 1985.
2. K. V. Hung, P. T. Huy, and D. L. Van, On some classes of codes defined by binary relations, *Acta Math. Vietnam.* **29** (2) (2004) 163–176.
3. K. V. Hung, P. T. Huy, and D. L. Van, Codes concerning roots of words, *Vietnam J. Math.* **32** (2004) 345–359.
4. M. Ito, H. Jürgensen, H. Shyr, and G. Thierrin, Outfix and infix codes and related classes of languages, *J. Computer and System Sciences* **43** (1991) 484–508.
5. H. Jürgensen and S. Konstatinidis, Codes, G. Rozenberg and A. Salomaa (Eds.), *Handbook of Formal Languages*, Springer, Berlin, 1997, 511–607.
6. N. H. Lam, Finite maximal infix codes, *Semigroup Forum* **61** (2000) 346–356.
7. H. Shyr, *Free Monoids and Languages*, Hon Min Book Company, Taichung, 1991.
8. H. Shyr and G. Thierrin, Hypercodes, *Information and Control* **24** (1974) 45–54.
9. D. L. Van, On a class of hypercodes, in M. Ito, T. Imaoka (Eds.), *Words, Languages and Combinatorics III* (Proceedings of the 3rd International Colloquium, Kyoto, 2000), World Scientific, 2003, 171-183.
10. D. L. Van and K. V. Hung, An approach to the embedding problem for codes defined by binary relations, *J. Automata, Languages and Combinatorics*, 2004, submitted (21 pages).
11. D. L. Van and K. V. Hung, Characterizations of some classes of codes defined by binary relations, *J. Automata, Languages and Combinatorics*, 2004, submitted (16 pages).