Vietnam Journal of Mathematics **39**:2(2011) 159-176

Vietnam Journal
of
MATHEMATICS

On Codes Defined by Binary Relations Part I: Embedding Problem

Do Long Van[1] **and Kieu Van Hung**[2]

[1] *Institute of Mathematics, 18 Hoang Quoc Viet, Hanoi, Vietnam.*

[2] *The Office of Government, 01 Hoang Hoa Tham, Hanoi, Vietnam.*

Received February 25, 2011
Revised July 25, 2011

Abstract. We consider the possibility to embed a finite (regular) code of a given class C of codes in a code maximal in C (not necessarily maximal as a code) which remains finite (regular, respectively). A general embedding schema is proposed for the classes of codes, which can be defined by length-increasing transitive binary relations. As applications, positive solutions for the embedding problem are obtained in a unified way for many classes of codes, well-known as well as new.

2000 Mathematics Subject Classification: 94A45, 68Q45.

Key words: Code, embedding problem, independent set, tree representation.

1. Introduction

Throughout the paper about codes we mean length-variable codes. A simple application of Zorn's Lemma showed that every code is included in a maximal code. For thin codes, regular codes in particular, the maximality is equivalent to the completeness, which concerns with optimal use of transmission alphabet. Thus maximal codes are important in both theoretical and practical points of view. For background of the theory of codes we refer to [1, 8, 17].

Every regular code is included in a maximal code, which is still regular [4]. There exist however finite codes, which cannot be included in any finite maximal code [13, 15]. These facts suggested the question: Is it true, for a given class C of codes, that every finite (regular) code in C can be included in a maximal code in

C (not necessarily maximal as a code) which is still finite (regular, respectively)? We call this the *embedding problem* for the class C of codes.

Until now the answer for the embedding problem is only known for several cases. For prefix codes the answer is positive for the finite case (folklore, see [1]). The embedding procedure is simple: given a finite prefix code X, it suffices to add the lacking leaves to the tree associated with X [1]. The regular case can be solved similarly by using the deterministic finite automaton associated with X [2, 16]. From a well-known result of M. P. Schützenberger (see [1]) it follows that it is impossible to embed a finite code with deciphering delay $d \neq 0$ into a maximal finite code with deciphering delay $d' \neq 0$. For the regular case, the embedding problem for these codes has been solved positively in [3]. There is a finite bifix code, which cannot be included in any finite maximal bifix code [1] whereas every regular bifix code is included in a regular maximal bifix code [24]. This generalizes the construction in [14]. The finite case for infix codes is solved positively in [7] and later by another way in [11] together with the regular case, etc.

H. J. Shyr and G. Thierrin had the idea of defining codes by partial orders. This idea was pursued further in subsequent work by M. Ito, H. Jürgensen, H. J. Shyr and G. Thierrin [6]. A major shift of this idea was achieved in the Ph. D. Thesis of S. S. Yu [22] and in some papers by H. Jürgensen and S. S. Yu [9, 10]. Most of this and some additional material was published in the work [8] by H. Jürgensen and S. Konstantinidis (see also [23]). The main idea is, that classes of codes can be defined as classes of sets of words, which are independent with respect to some relation.

In this paper, we propose a general embedding schema for the classes of codes defined by length-increasing transitive binary relations (Theorem 3.1). Using this, positive solutions are obtained in a unified way for some well-known classes of codes, namely those of prefix codes, suffix codes, p-infix codes, s-infix codes, infix codes, hypercodes, and also for some new classes of codes introduced here, namely those of subinfix codes, p-subinfix codes, s-subinfix codes, sucyperinfix codes, p-sucyperinfix codes, s-sucyperinfix codes, superinfix codes, p-superinfix codes, s-superinfix codes, p-hypercodes, s-hypercodes, sucypercodes and supercodes. This work is motivated by the idea to define codes as independent sets with respect to a binary relation, the way to embed an infix code in a maximal one in [11], and the literal representation for prefix codes [1]. Several among the new kinds of codes introduced here are research subject of another paper [20]. A part of these results has been published without proof in [21] and presented at Conference on Algebraic Informatics, CAI05, 2005.

2. Defining Codes by Binary Relations

In this section, we recall some well-known classes of codes, which can be defined by binary relations (see [6, 17, 18, 19]).

Let A throughout be a non-empty finite alphabet. Let A^* be the free monoid generated by A, that is the set of words over A. The empty word is denoted by 1 and $A^+ = A^* - 1$. The number of occurrences of letters in a word u is the *length* of u, denoted by $|u|$. Any set of words is a *language*. A language X is a *code* over A if for any $n, m \geq 1$ and any $x_1, \ldots, x_n, y_1, \ldots, y_m \in X$, 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 another code over A. Let C be a class of codes over A. A code $X \in C$ is *maximal in* C (not necessarily maximal as a code) if it is not properly contained in another code in C. For further details of the theory of codes we refer to [1, 8, 17].

Given a binary relation $\prec$ on A^*. A subset X in A^* is an *independent set* with respect to the relation $\prec$ if any two elements of X are not in this relation. A class C of codes is *defined by* $\prec$ if these codes are exactly the independent sets with respect to $\prec$. The class C is then denoted by $C_\prec$. When the relation $\prec$ characterizes some property α of words, instead of $\prec$ we write $\prec_\alpha$, and also C_α stands for $C_{\prec_\alpha}$. The relation $\prec$ is said to be *length-increasing* if for any $u, v \in A^* : u \prec v$ implies $|u| < |v|$. We denote by $\preceq$ the reflexive closure of $\prec$, i.e. for any $u, v \in A^*, u \preceq v$ if and only if $u = v$ or $u \prec v$.

A word u is called an *infix* (a *prefix*, a *suffix*) of a word v if there exist words x, y such that $v = xuy$ ($v = uy$, $v = xu$, respectively). The infix (prefix, suffix) is *proper* if $xy \neq 1$ ($y \neq 1$, $x \neq 1$, respectively). A word u is a *subword* of a word 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.

Definition 2.1. Let A be an alphabet and $X \subseteq A^+$.

(i) X is a *prefix* (*suffix*) *code* if no word in X is a proper prefix (suffix, respectively) of another word in X;

(ii) X is a *bifix code* if it is both a prefix code and a suffix code;

(iii) X is an *infix* (a *p-infix*, a *s-infix*) *code* if no word in X is an infix of a proper infix (prefix, suffix, respectively) of another word in X;

(iv) X is a *hypercode* if no word in X is a proper subword of another word in it.

The classes of prefix codes, suffix codes, bifix codes, infix codes, p-infix codes, s-infix codes and hypercodes are denoted respectively by C_p, C_s, C_b, C_i, $C_{p.i}$, $C_{s.i}$ and C_h. It is easy to see that these classes of codes are defined respectively by the relations which satisfy, for any $u, v \in A^*$, the following corresponding conditions:

$u \prec_p v \Leftrightarrow v = ux$, with $x \neq 1$;
$u \prec_s v \Leftrightarrow v = xu$, with $x \neq 1$;
$u \prec_b v \Leftrightarrow (u \prec_p v) \vee (u \prec_s v)$;

$u \prec_i v \Leftrightarrow v = xuy$, with $xy \neq 1$;
$u \prec_{p.i} v \Leftrightarrow v = xuy$, with $y \neq 1$;
$u \prec_{s.i} v \Leftrightarrow v = xuy$, with $x \neq 1$;
$u \prec_h v \Leftrightarrow \exists n \geq 1 : u = u_1 \dots u_n \wedge v = x_0 u_1 x_1 \dots u_n x_n$, with $x_0 \dots x_n \neq 1$.

Prefix codes, suffix codes and bifix codes play a fundamental role in the theory of codes (see [1, 17]). Details about infix codes, p-infix codes and s-infix codes we refer to [6, 17]. Hypercodes, a special kind of infix codes, have some interesting properties, specially, all hypercodes are finite, as it has in fact been proved by Higman (see for instance [17] or [12]). Relationship between these classes of codes can be summarized in the following proposition, where some of the relations have been already proved or they may be easily proved thanks to results by Ito et al. [6].

Proposition 2.2. *Over any alphabet consisting of at least two letters, the following holds true:*

(i) $C_b \subset C_p, C_b \subset C_s, C_b = C_p \cap C_s, C_h \subset C_i$;
(ii) $C_i \subset C_{p.i}$, $C_i \subset C_{s.i}$, $C_i = C_{p.i} \cap C_{s.i}$, $C_i \subset C_b$, $C_{p.i} \subset C_p$, $C_{s.i} \subset C_s$.

3. A General Embedding Schema

In this section we propose a general embedding schema for the classes of codes defined by length-increasing transitive binary relations which will be used in the sequel.

Let $\prec$ be a binary relation on A^* and $u, v \in A^*$. We say that u *depends* on v if $u \prec v$ or $v \prec u$ holds. Otherwise, u is *independent* of v. These notions can be extended to subsets of words in a standard way. Namely, a word u is dependent on a subset X if it depends on some word in X. Otherwise, u is independent of X. For brevity, we shall adopt the following notation:

$$u \prec X \rightleftharpoons \exists v \in X : u \prec v; \quad X \prec u \rightleftharpoons \exists v \in X : v \prec u.$$

An element u in X is *minimal* in X if there is no word v in X such that $v \prec u$. When X is finite, by $\max X$ we denote the maximal wordlength of X.

Now, for every subset X in A^* we denote by D_X, I_X, L_X and R_X the sets of words dependent on X, independent of X, non-minimal in I_X and minimal in I_X, respectively. In notation:

$$\begin{aligned} D_X &= \{u \in A^* \mid u \prec X \vee X \prec u\}; \\ I_X &= A^* - D_X; \\ L_X &= \{u \in I_X \mid I_X \prec u\}; \\ R_X &= I_X - L_X. \end{aligned}$$

When there is no risk of confusion, for brevity, we simply write D, I, L, R instead.

We can now formulate a fundamental result by means of which one obtains solutions of the embedding problem for many classes of codes in a unified way.

Theorem 3.1. *Let $\prec$ be a length-increasing transitive binary relation on A^* which defines the class $C_\prec$ of codes. Then, for any code X in $C_\prec$, we have:*

(i) *R_X is a maximal code in $C_\prec$ which contains X;*

(ii) *If moreover the relation $\prec$ satisfies the condition*

$$\exists k \geq 1 \forall u, v \in A^+ : (|v| \geq |u| + k) \wedge (u \not\prec v) \Rightarrow \exists w : (|w| \geq |u|) \wedge (w \prec v) \quad (*)$$

then the finiteness of X implies the finiteness of R_X, and $\max R_X \leq \max X + k - 1$.

Proof. (i) First, we prove that $X \subseteq R_X$. There must be $X \cap D_X = \emptyset$, otherwise there would exist u in $X \cap D_X$ and therefore, by the definition of D_X, there exists $v \in X$ such that either $u \prec v$ or $v \prec u$. This is impossible because X is an independent set with respect to $\prec$. So $X \subseteq A^* - D_X = I_X$. Next, we have $X \cap L_X = \emptyset$, otherwise there would exist $u \in X \cap L_X$ and therefore, by the definition of L_X, there is $v \in I_X$ such that $v \prec u$, which contradicts the definition of I_X. Thus $X \subseteq I_X - L_X = R_X$.

Now, we show that $R_X \in C_\prec$, or equivalently, R_X is an independent set with respect to $\prec$. Suppose the contrary that there exist $u, v \in R_X$ such that $u \prec v$. Since $R_X \subseteq I_X$, it follows that $u, v \in I_X$. This implies, by the definition of L_X, $v \in L_X$. Thus $v \in R_X \cap L_X$, a contradiction.

Finally, we prove that R_X is maximal in $C_\prec$. Instead we show that R_X is a maximal independent set with respect to $\prec$. Indeed, suppose this is not the case, there would exist $v_0 \notin R_X$ such that $R_X \cup \{v_0\}$ is still an independent set with respect to $\prec$. Because $X \subseteq R_X$, this implies $v_0 \notin D_X$. So we have $v_0 \in I_X - R_X = L_X$. Now, we show by induction that for any $n \geq 1$ there exist $v_1, \ldots, v_n \in L_X$ such that $v_n \prec \cdots \prec v_1 \prec v_0$. Indeed, by the definition of L_X, there exists $v_1 \in I_X$ such that $v_1 \prec v_0$. If v_1 is in R_X then $R_X \cup \{v_0\}$ is no more an independent set with respect to $\prec$, a contradiction. Therefore $v_1 \in L_X$, i.e. the assertion holds true for $n = 1$. Now, suppose the assertion is true for n, we prove that it is also true for $n + 1$. Because v_n is in L_X, there exists $v_{n+1} \in I_X$ such that $v_{n+1} \prec v_n$. By the transitivity of $\prec$, we have $v_{n+1} \prec v_0$. If v_{n+1} is in R_X then $R_X \cup \{v_0\}$ could not be an independent set with respect to $\prec$. Hence $v_{n+1} \in I_X - R_X = L_X$. Thus, we have proved that there exists an infinite sequence $v_1, v_2, \ldots$ of elements in L_X such $\cdots \prec v_2 \prec v_1 \prec v_0$. Since $\prec$ is length-increasing, we have $|v_{i+1}| < |v_i|$ for all $i \geq 0$. But this is impossible because $|v_0|$ is finite. Thus, R_X must be a maximal independent set with respect to $\prec$ and hence a maximal code in $C_\prec$ which contains X.

(ii) Suppose X is a finite code in $C_\prec$ and $n = \max X$. We will prove that for any $v \in I_X$ with $|v| \geq n + k$ there exists $w \in I_X$ such that $w \prec v$. Indeed, let u be a word of maximal length in X. Then $|v| \geq |u| + k$. Since every element

of I_X is independent of X, it follows that $u \not\prec v$. By the condition $(*)$, there exists w such that $|w| \geq |u|$ and $w \prec v$. The word w must be in I_X, otherwise, the transitivity of $\prec$ implies the existence of some $x \in X$ such that $x \prec v$, a contradiction. Thus the maximal wordlength of R_X cannot exceed $n+k-1$ that required to prove. ■

4. New Classes of Codes

We introduce, in this section, some new classes of codes whose embedding problem will be considered in the sequel. All such classes, as we shall see later (Proposition 4.5), are subclasses of prefix codes or suffix codes.

Given $u, v \in A^+$. Let u be a subword of v, $u = u_1 \dots u_n, v = x_0u_1x_1 \dots u_nx_n$. As $u \neq 1$, we may assume $u_i \neq 1$ for all i. Then, we call u a *right-proper subword* of v if $x_1 \dots x_n \neq 1$. Dually, if $x_0 \dots x_{n-1} \neq 1$ then u is a *left-proper subword* of v. A word u is a *permutation* of a word v if u and v are commutatively equivalent (see [1]), i.e. if $|u|_a = |v|_a$ for all $a \in A$, where $|u|_a$ denotes the number of occurrences of a in u. We say that u is a *cyclic permutation* of v if u and v are conjugate, i.e. if there exist x, y such that $u = xy$ and $v = yx$. For any word u, $\pi(u)$ and $\sigma(u)$ denote the sets of all permutations and all cyclic permutations of u, respectively. These operations π and σ are extended to languages in a standard way.

Definition 4.1. Let A be an alphabet and $X \subseteq A^+$.

(i) X is a *subinfix* (*p-subinfix, s-subinfix*) *code* if no word in X is a **sub**word of a proper **infix** (**p**refix, **s**uffix, respectively) of another word in X;

(ii) X is a *p-hypercode* (*s-hypercode*) if no word in X is a right-proper (left-proper, respectively) subword of another word in X;

(iii) X is a *superinfix* (*p-superinfix, s-superinfix*) *code* if no word in X is a **su**bword of a **per**mutation of a proper **infix** (**p**refix, **s**uffix, respectively) of another word in X;

(iv) X is a *sucyperinfix* (*p-sucyperinfix, s-sucyperinfix*) *code* if no word in X is a **su**bword of a **cy**clic **per**mutation of a proper **infix** (**p**refix, **s**uffix, respectively) of another word in X;

(v) X is a *supercode* (*sucypercode*) if no word in X is a proper **su**bword of a **per**mutation (**cy**clic **per**mutation, respectively) of another word in it.

This definition itself explains the way we named the new kinds of codes. It is easy to see that these classes of codes have, as defining relations, the following, respectively:

$$u \prec_{si} v \Leftrightarrow \exists w \in A^* : w \prec_i v \wedge u \preceq_h w;$$
$$u \prec_{p.si} v \Leftrightarrow \exists w \in A^* : w \prec_p v \wedge u \preceq_h w;$$
$$u \prec_{s.si} v \Leftrightarrow \exists w \in A^* : w \prec_s v \wedge u \preceq_h w;$$

$u \prec_{p.h} v \Leftrightarrow \exists n \geq 1 : u = u_1...u_n \wedge v = x_0u_1x_1...u_nx_n$, with $x_1...x_n \neq 1$;
$u \prec_{s.h} v \Leftrightarrow \exists n \geq 1 : u = u_1...u_n \wedge v = x_0u_1x_1...u_nx_n$, with $x_0...x_{n-1} \neq 1$;
$u \prec_{spi} v \Leftrightarrow (\exists v' : v' \prec_i v)(\exists v'' \in \pi(v')) : u \preceq_h v''$;
$u \prec_{p.spi} v \Leftrightarrow (\exists v' : v' \prec_p v)(\exists v'' \in \pi(v')) : u \preceq_h v''$;
$u \prec_{s.spi} v \Leftrightarrow (\exists v' : v' \prec_s v)(\exists v'' \in \pi(v')) : u \preceq_h v''$;
$u \prec_{scpi} v \Leftrightarrow (\exists v' : v' \prec_i v)(\exists v'' \in \sigma(v')) : u \preceq_h v''$;
$u \prec_{p.scpi} v \Leftrightarrow (\exists v' : v' \prec_p v)(\exists v'' \in \sigma(v')) : u \preceq_h v''$;
$u \prec_{s.scpi} v \Leftrightarrow (\exists v' : v' \prec_s v)(\exists v'' \in \sigma(v')) : u \preceq_h v''$;
$u \prec_{sp} v \Leftrightarrow \exists v' \in \pi(v) : u \prec_h v'$;
$u \prec_{scp} v \Leftrightarrow \exists v' \in \sigma(v) : u \prec_h v'$.

Example 4.2. Denote by R the reverse operation for languages. Consider the subsets $X_1 = \{aba, bab^2a\}$, $X_1^R = \{aba, ab^2ab\}$, $X_2 = ab^*a$, $X_3 = \{a, ba\}$, $X_3^R = \{a, ab\}$, $X_4 = \{ab, b^3a\}$, $X_4^R = \{ba, ab^3\}$, $X_5 = \{abab, a^2b^3\}$ and $X_6 = \{a^2, ab, b^2a\}$ over the alphabet $A = \{a, b\}$. It is easy to check that the following holds true:

$$X_1 \in C_{p.si} - C_{si}, X_1^R \in C_{s.si} - C_{si}, X_2 \in C_{si} \cap C_{spi} \cap C_{scpi};$$
$$X_3 \in C_{p.h} - C_h, X_3^R \in C_{s.h} - C_h, X_6 \in C_h - C_{scp};$$
$$X_4 \in C_{p.spi} - C_{scpi}, X_4^R \in C_{s.spi} - C_{scpi}, X_5 \in C_{scpi} \cap C_{scp} - C_{spi}.$$

Although, as we shall see below (Proposition 4.5), the class of p-hypercodes (s-hypercodes) strictly contains the class of hypercodes, the former codes, however, are still finite.

Proposition 4.3. *All p-hypercodes and s-hypercodes over a finite alphabet are finite.*

Proof. Let X be a p-hypercode over A, $|A| = n$. First we show that every chain $u_1 \prec_h u_2 \prec_h \ldots$ of elements in X with respect to $\prec_h$ has the length no more than n. Indeed, if it is not the case then there exist i, j with $1 \leq i < j$ such that u_i and u_j commence with a same letter $a \in A$. We must have $u_i = au_i'$, $u_j = au_j'$. On the other hand, because u_i and u_j are not in the relation $\prec_{p.h}$, we have $u_j = xu_i$ with $x \neq 1$. Thus $u_j = au_j' = xu_i = ax'au_i'$ with $x'a \neq 1$, i. e. $u_i \prec_{p.h} u_j$, a contradiction. Next, we denote by X' the set of all the elements of X which are maximal with respect to $\prec_h$. Clearly, X' is an independent set with respect to $\prec_h$, i. e. a hypercode and therefore finite. For every $x \in A^*$ we put $Sub(x) = \{y \in X \mid y \preceq_h x\}$. Obviously, $Sub(x)$ is finite. By the above, $\forall y \in X\ \exists x \in X'$ such that $y \preceq_h x$. All these together imply $X = \bigcup_{x \in X'} Sub(x)$ which means that X is finite. For s-hypercodes, the proof is similar. ∎

The following fact will be useful in the sequel (see [8]).

Lemma 4.4. *Let $\prec_1$ and $\prec_2$ be binary relations on A^*. Then $C_{\prec_1 \cup \prec_2} = C_{\prec_1} \cap C_{\prec_2}$.*

The relationship between the classes of codes under consideration can be summarized as below.

Proposition 4.5. *Over any alphabet consisting of at least two letters, the following holds true:*

(i) $C_{si} \subset C_{p.si}$, $C_{si} \subset C_{s.si}$, $C_{si} = C_{p.si} \cap C_{s.si}$, $C_{si} \subset C_i$, $C_{p.si} \subset C_{p.i}$, $C_{s.si} \subset C_{s.i}$;

(ii) $C_{scpi} \subset C_{p.scpi}$, $C_{scpi} \subset C_{s.scpi}$, $C_{scpi} = C_{p.scpi} \cap C_{s.scpi}$, $C_{scpi} \subset C_{si}$, $C_{p.scpi} \subset C_{p.si}$, $C_{s.scpi} \subset C_{s.si}$;

(iii) $C_{spi} \subset C_{p.spi}$, $C_{spi} \subset C_{s.spi}$, $C_{spi} = C_{p.spi} \cap C_{s.spi}$, $C_{spi} \subset C_{scpi}$, $C_{p.spi} \subset C_{p.scpi}$, $C_{s.spi} \subset C_{s.scpi}$;

(iv) $C_h \subset C_{p.h}$, $C_h \subset C_{s.h}$, $C_h \subset C_{si}$, $C_{p.h} \subset C_{p.si}$, $C_{s.h} \subset C_{s.si}$, $C_h = C_{p.h} \cap C_{s.h} = C_{p.h} \cap C_{s.si} = C_{p.si} \cap C_{s.h}$;

(v) $C_{sp} \subset C_{scp} \subset C_h, C_{scp} \subset C_{scpi}, C_{sp} \subset C_{spi}$.

Proof. We prove only the item (i). For the remaining items the argument is similar. The inclusions $C_{si} \subseteq C_{p.si}$ and $C_{si} \subseteq C_{s.si}$ hold because proper prefixes and proper suffixes of a word are particular cases of proper infixes of the word. The sets X_1 and X_1^R in Example 4.2 show that the inclusions are strict. By definition of the relations $\prec_{p.si}$, $\prec_{s.si}$ and $\prec_{si}$ we have $\prec_{si} = \prec_{p.si} \cup \prec_{s.si}$. Hence, by Lemma 4.4, $C_{si} = C_{p.si} \cap C_{s.si}$. Next, since infixes of a word are particular cases of subwords of it, the inclusions $C_{si} \subseteq C_i$, $C_{p.si} \subseteq C_{p.i}$ and $C_{s.si} \subseteq C_{s.i}$ are true. The following example proves the strictness of the inclusions. Consider $X = \{aba, bab^2ab\}$ over $A = \{a, b\}$. The word $u = aba$ is not a proper infix of $v = bab^2ab$. But u is a subword of bab^2a, a proper prefix of v, and also a subword of ab^2ab, a proper suffix of v. Thus X is an infix code not being either a p-subinfix code nor a s-subinfix code. ∎

Remark 4.6. It is easy to check that the set $X = \{ab^2, ba^3b\}$ is both a hypercode and a superinfix code, i.e. $C_h \cap C_{spi} \neq \emptyset$. However, C_h and C_{spi} are not comparable by inclusion. Indeed, evidently the infinite superinfix code ab^*a cannot be a hypercode. On the other hand, the set $Y = \{bab, ab^3a\}$, which is easily verified to be a hypercode, is not a superinfix code because bab is a subword of a permutation of the proper infix ab^3 of ab^3a.

By virtue of Propositions 2.2, 4.5 and Remark 4.6, the relative positions of the classes of codes under consideration can be illustrated in the Figure 1, where the arrow $\rightarrow$ stands for a strict inclusion. It is worthy to note that if we restrict ourselves to considering only one-letter alphabets then all the classes of codes represented in the Figure 1 coincide.

5. Embedding Problem for Regular Case

In this section we apply Theorem 3.1 to show that the embedding problem for the classes of codes introduced above has a positive solution in the regular case. For proving this we need some lemmas.

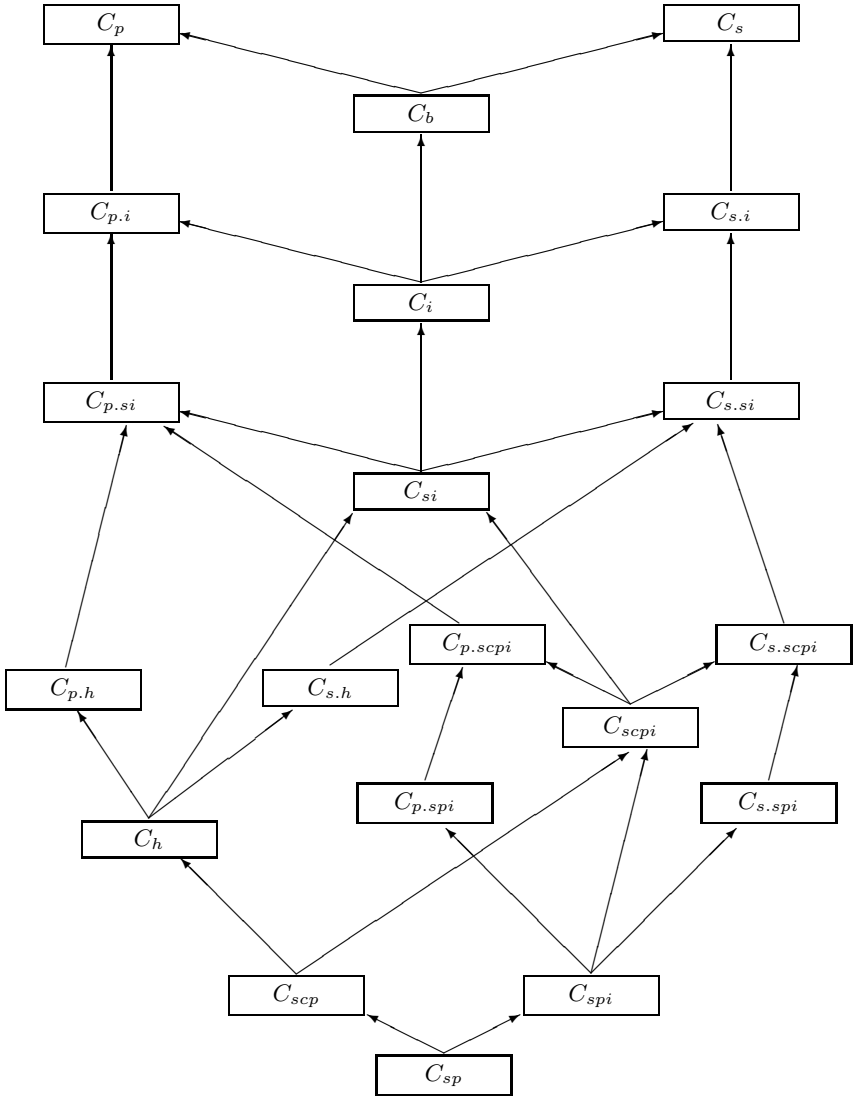

Fig. 1 Relative positions of the classes of codes

Lemma 5.1. *For any $u, v \in A^*$ we have $\exists v' \in \sigma(v) : u \preceq_h v'$ if and only if $\exists u' \in \sigma(u) : u' \preceq_h v$.*

Proof. For $u = 1$, the assertion is true trivially. Suppose $u \neq 1$ and $v' \in \sigma(v)$ such that $u \preceq_h v'$. Then, on one hand, $v = xy$, $v' = yx$ for some $x, y \in A^*$. On the other hand, $u = a_1 \dots a_n$ and $v' = x_0 a_1 x_1 \dots a_n x_n$ with $n \geq 1$, $a_1, \dots, a_n \in A$, $x_0, x_1, \dots, x_n \in A^*$. Since $v' = yx$ and $v' = x_0 a_1 x_1 \dots a_n x_n$,

it follows that there exists $k, 0 \leq k \leq n$, such that $y = x_0a_1x_1 \dots a_kx'_k$, $x = x''_ka_{k+1}x_{k+1} \dots a_nx_n$ with $x_k = x'_kx''_k$, $x'_k, x''_k \in A^*$. Therefore, $v = xy = x''_ka_{k+1}x_{k+1} \dots a_nx_nx_0a_1x_1 \dots a_kx'_k$. Hence, the word $u' = a_{k+1} \dots a_na_1 \dots a_k \in \sigma(u)$ and $u' \preceq_h v$. Conversely, suppose $u' \in \sigma(u)$ such that $u' \preceq_h v$. If $u' = u$ then $u \preceq_h v'$ with $v' = v$. Assume $u' \neq u$ and $u = a_1 \dots a_n$ with $n \geq 1$, $a_1, \dots, a_n \in A$. From $u' \in \sigma(u)$, $u' \neq u$ it follows that $u' = a_{k+1} \dots a_na_1 \dots a_k$ for some k, $1 \leq k < n$. Since $u' \preceq_h v$, this implies $v = x_0a_{k+1}x_1 \dots a_nx_{n-k}a_1x_{n-k+1} \dots a_kx_n$ with $x_0, x_1, \dots, x_n \in A^*$. Let us take $v' = a_1x_{n-k+1} \dots a_kx_nx_0a_{k+1}x_1 \dots a_nx_{n-k}$, we have $v' \in \sigma(v)$ and $u \preceq_h v'$. ■

From now on, we denote by Ω the set $\{p, s, p.i, s.i, i, p.si, s.si, si, p.scpi, s.scpi, scpi, p.spi, s.spi, spi, p.h, s.h, h, scp, sp\}$ and $\Omega' = \Omega - \{p.h, s.h, h, scp, sp\}$.

Lemma 5.2. *The relations $\prec_\alpha$, $\alpha \in \Omega$ are transitive and length-increasing.*

Proof. The fact that all the mentioned above relations are length-increasing is immediate from their definitions. The transitivity of these relations, except for $\prec_{p.scpi}$, $\prec_{s.scpi}$, $\prec_{scpi}$ and $\prec_{scp}$, is straightforward. We verify, for example, the transitivity of $\prec_{scpi}$. Let us have $u \prec_{scpi} v \prec_{scpi} w$. By definition, $\exists v' : v' \prec_i v, \exists v'' \in \sigma(v') : u \preceq_h v''$ and $\exists w' : w' \prec_i w, \exists w'' \in \sigma(w') : v \preceq_h w''$. This implies, by Lemma 5.1, that $\exists u' \in \sigma(u) : u' \preceq_h v'$. From $v' \prec_i v$, hence $v' \prec_h v$, and from the transitivity of $\prec_h$, it follows that $u' \prec_h v$. As $u' \prec_h v \preceq_h w''$, we have $u' \prec_h w''$. By Lemma 5.1, $\exists u'' \in \sigma(u') = \sigma(u) : u'' \preceq_h w'$. Again by Lemma 5.1, $\exists w''' \in \sigma(w') : u \prec_h w'''$. So $u \prec_{scpi} w$. ■

Note that the relation $\prec_b$ is not transitive. For this reason the embedding problem for bifix codes cannot be solved by means of Theorem 3.1.

Lemma 5.3. *For any X in A^* we have:*

(i) *$\sigma(X)$ is regular if X is regular;*

(ii) *If X is a maximal superinfix (p-superinfix, s-superinfix) code then X is also a maximal sucyperinfix (p-sucyperinfix, s-sucyperinfix, respectively) code.*

Proof. (i) This fact is well-known (see [5], Exercise 4.2.11, page 148). (ii) These are immediate consequences of Theorem 3.4(ii) and Corollary 3.5 in [20]. ■

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 in A. Put $A_\# = A \cup \{\#\}$. Consider the regular substitutions S_1, S_2 and the homomorphism h defined as follows:

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

$$h : A_{\#}^* \to A^*, \text{ with } h(\#) = 1 \text{ and } h(a) = a \text{ for all } a \in A.$$

As we see later, 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 the empty word. The inverse homomorphism h^{-1} "chooses" in a word the positions where the words of A^+ may be inserted, while S_2 realizes the insertions by replacing $\#$ by A^+. Note that regular languages are closed under regular substitutions, homomorphisms and inverse homomorphisms (see [5]).

Lemma 5.4. *Given $\alpha \in \Omega$ and $X \in C_\alpha$. Then R_X can be computed by the following expressions according to the case:*

(i) *Case of prefix codes: $R = I - IA^+$, where $I = A^* - XA^- - XA^+$ and A^- stands for $(A^+)^{-1}$.*

(ii) *Case of suffix codes: $R = I - A^+I$, where $I = A^* - A^-X - A^+X$.*

(iii) *Case of p-infix codes: $R = I - (IA^+ + A^+IA^+)$, where $I = A^* - XA^- - A^-XA^- - XA^+ - A^+XA^+$.*

(iv) *Case of s-infix codes: $R = I - (A^+I + A^+IA^+)$, where $I = A^* - A^-X - A^-XA^- - A^+X - A^+XA^+$.*

(v) *Case of infix codes: $R = I - (IA^+ + A^+I + A^+IA^+)$, where $I = A^* - XA^- + A^-X + A^-XA^- - XA^+ - A^+X - A^+XA^+$.*

(vi) *Case of p-subinfix codes: $R = I - S_2(h^{-1}(I) \cap A_{\#}^*\{\#\})$, where $I = A^* - h(S_1(X) \cap A_{\#}^*\{\#\}) - S_2(h^{-1}(X) \cap A_{\#}^*\{\#\})$.*

(vii) *Case of s-subinfix codes: $R = I - S_2(h^{-1}(I) \cap \{\#\}A_{\#}^*)$, where $I = A^* - h(S_1(X) \cap \{\#\}A_{\#}^*) - S_2(h^{-1}(X) \cap \{\#\}A_{\#}^*)$.*

(viii) *Case of subinfix codes: $R = I - S_2(h^{-1}(I) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\}))$, where $I = A^* - h(S_1(X) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\})) - S_2(h^{-1}(X) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\}))$.*

(ix) *Case of p-sucyperinfix codes: $R = I - S_2(h^{-1}(\sigma(I)) \cap A_{\#}^*\{\#\})$, where $I = A^* - \sigma(h(S_1(X) \cap A_{\#}^*\{\#\})) - S_2(h^{-1}(\sigma(X)) \cap A_{\#}^*\{\#\})$.*

(x) *Case of s-sucyperinfix codes: $R = I - S_2(h^{-1}(\sigma(I)) \cap \{\#\}A_{\#}^*)$, where $I = A^* - \sigma(h(S_1(X) \cap \{\#\}A_{\#}^*)) - S_2(h^{-1}(\sigma(X)) \cap \{\#\}A_{\#}^*)$.*

(xi) *Case of sucyperinfix codes: $R = I - S_2(h^{-1}(\sigma(I)) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\}))$, where $I = A^* - \sigma(h(S_1(X) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\}))) - S_2(h^{-1}(\sigma(X)) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\}))$.*

(xii) *Case of p-superinfix codes: $R = I - S_2(h^{-1}(\pi(I)) \cap A_{\#}^*\{\#\})$, where $I = A^* - \pi(h(S_1(X) \cap A_{\#}^*\{\#\})) - S_2(h^{-1}(\pi(X)) \cap A_{\#}^*\{\#\})$.*

(xiii) *Case of s-superinfix codes: $R = I - S_2(h^{-1}(\pi(I)) \cap \{\#\}A_{\#}^*)$, where $I = A^* - \pi(h(S_1(X) \cap \{\#\}A_{\#}^*)) - S_2(h^{-1}(\pi(X)) \cap \{\#\}A_{\#}^*)$.*

(xiv) *Case of superinfix codes: $R = I - S_2(h^{-1}(\pi(I)) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\}))$, where $I = A^* - \pi(h(S_1(X) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\}))) - S_2(h^{-1}(\pi(X)) \cap (\{\#\}A_{\#}^* \cup A_{\#}^*\{\#\}))$.*

(xv) *Case of p-hypercodes:* $R = I - S_2(h^{-1}(I) \cap (A_\#^*\{\#\}A_\#^* - \{\#\}^+A^+))$, *where* $I = A^* - h(S_1(X) \cap (A_\#^*\{\#\}A_\#^* - \{\#\}^+A^+)) - S_2(h^{-1}(X) \cap (A_\#^*\{\#\}A_\#^* - \{\#\}^+A^+))$.

(xvi) *Case of s-hypercodes:* $R = I - S_2(h^{-1}(I) \cap (A_\#^*\{\#\}A_\#^* - A^+\{\#\}^+))$, *where* $I = A^* - h(S_1(X) \cap (A_\#^*\{\#\}A_\#^* - A^+\{\#\}^+)) - S_2(h^{-1}(X) \cap (A_\#^*\{\#\}A_\#^* - A^+\{\#\}^+))$.

(xvii) *Case of hypercodes:* $R = I - S_2(h^{-1}(I) \cap A_\#^*\{\#\}A_\#^*)$, *where* $I = A^* - h(S_1(X) \cap A_\#^*\{\#\}A_\#^*) - S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^*)$.

(xviii) *Case of sucypercodes:* $R = I - \sigma(S_2(h^{-1}(I) \cap A_\#^*\{\#\}A_\#^*))$, *where* $I = A^* - h(S_1(\sigma(X)) \cap A_\#^*\{\#\}A_\#^*) - \sigma(S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^*))$.

(xix) *Case of supercodes:* $R = I - \pi(S_2(h^{-1}(I) \cap A_\#^*\{\#\}A_\#^*))$, *where* $I = A^* - h(S_1(\pi(X)) \cap A_\#^*\{\#\}A_\#^*) - \pi(S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^*))$.

Proof. We treat only the case of sucypercodes. For the other cases the argument is similar. The proof follows from the following computations, which are not difficult to be verified.

$\{u \in A^* \mid u \prec_{scp} X\} = \{u \in A^* \mid u \prec_h \sigma(X)\} = h(S_1(\sigma(X)) \cap A_\#^*\{\#\}A_\#^*)$;

$\{u \in A^* \mid X \prec_{scp} u\} = \sigma(\{u \in A^* \mid X \prec_h u\}) = \sigma(S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^*))$;

$D = \{u \in A^* \mid u \prec_{scp} X \vee X \prec_{scp} u\}$

$= h(S_1(\sigma(X)) \cap A_\#^*\{\#\}A_\#^*) + \sigma(S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^*))$;

$I = A^* - h(S_1(\sigma(X)) \cap A_\#^*\{\#\}A_\#^*) - \sigma(S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^*))$;

$L = \{u \in I \mid I \prec_{scp} u\} = I \cap \sigma(S_2(h^{-1}(I) \cap A_\#^*\{\#\}A_\#^*))$;

$R = I - L = I - \sigma(S_2(h^{-1}(I) \cap A_\#^*\{\#\}A_\#^*))$. ∎

Theorem 5.5. *For any $\alpha \in \Omega'$, every regular code in C_α is contained in a maximal code in C_α which is still regular.*

Proof. By Lemma 5.2, the relations $\prec_\alpha$, $\alpha \in \Omega'$, are transitive and length-increasing. Given a regular code X in C_α. By Theorem 3.1(i), R_X is a maximal code in C_α which contains X. Now we distinguish two cases:

Case 1. $\alpha \in \Omega' - \{spi, p.spi, s.spi\}$. By Lemma 5.3(i), R_X, which can be computed by corresponding expressions in Lemma 5.4, is still regular because it can be obtained from X by finitely many applications of the operations which all preserve the regularity.

Case 2. $\alpha \in \{spi, p.spi, s.spi\}$. Suppose $\alpha = spi$. Then X is a superinfix code and therefore also a sucyperinfix code. Denote by ${R_X}^{spi}$ and ${R_X}^{scpi}$ the sets R_X with X to be considered as a superinfix code and a sucyperinfix code respectively. These sets can be computed by the expressions (xiv) and (xi) in Lemma 5.4 respectively. It is not difficult to check that ${R_X}^{spi} \subseteq {R_X}^{scpi}$. On one hand, by Theorem 3.1(i) and Lemma 5.3(ii), ${R_X}^{spi}$ is a maximal sucyperinfix code containing X. On the other hand, as shown in *Case 1*, ${R_X}^{scpi}$ is a regular maximal sucyperinfix code containing X. It follows that ${R_X}^{spi} = {R_X}^{scpi}$. Thus,

$R_X{}^{spi}$ is a regular maximal superinfix code containing X. For $\alpha = p.spi$ or $\alpha = s.spi$ the arguments are similar. ■

6. Embedding Problem for Finite Case

Our aim in this section is to solve the embedding problem for the above mentioned classes of codes in the finite case. Namely, we will exhibit algorithms to construct, for every finite code X in a class C_α, $\alpha \in \Omega$, a finite maximal code in the same class which contains X.

Theorem 6.1. *For any $\alpha \in \Omega$, every finite code X in C_α is contained in a finite maximal code Y in C_α with $\max Y = \max X$.*

Proof. By Lemma 5.2, all the relations $\prec_\alpha$, $\alpha \in \Omega$, are transitive and length-increasing. These relations satisfy the condition $(*)$ of Theorem 3.1 with $k = 1$. Indeed, let $u \in A^*$ with $|u| = n \geq 1$ and let $v \in A^*$ such that $u \not\prec_\alpha v$ and $|v| \geq n + 1$. We show that there exists always w of the length $|v| - 1 \geq |u|$ such that $w \prec_\alpha v$. Indeed, note that for every $\alpha \in \Omega$ we have either $\prec_p \subseteq \prec_\alpha$ or $\prec_s \subseteq \prec_\alpha$, say the former holds true. There exists always w with $|w| = |v|-1 \geq |u|$ such that $w \prec_p v$, and therefore $w \prec_\alpha v$. Next, given a finite code X in C_α. By Theorem 3.1(i) and (ii), R_X is a finite maximal code in C_α which contains X and $\max R_X \leq \max X - 1 + 1 = \max X$, hence $\max R_X = \max X$. Setting $Y = R_X$ we obtain the required to prove. ■

Denote by $A^{[n]}$ the set of all the words in A^* whose length is less than or equal to n. As an immediate consequence of Lemma 5.4 and Theorem 6.1 we have:

Corollary 6.2. *Given $\alpha \in \Omega$ and $X \in C_\alpha$ with $\max X = n$. Then the maximal code R_X in C_α which contains X can be computed by the following "restricted" expressions according to the case:*

(i) *Case of prefix codes: $R = I - IA^+ \cap A^{[n]}$, where $I = A^{[n]} - XA^- - XA^+ \cap A^{[n]}$.*

(ii) *Case of suffix codes: $R = I - A^+I \cap A^{[n]}$, where $I = A^{[n]} - A^-X - A^+X \cap A^{[n]}$.*

(iii) *Case of p-infix codes: $R = I - (IA^+ + A^+IA^+) \cap A^{[n]}$, where $I = A^{[n]} - XA^- - A^-XA^- - (XA^+ + A^+XA^+) \cap A^{[n]}$.*

(iv) *Case of s-infix codes : $R = I - (A^+I + A^+IA^+) \cap A^{[n]}$, where $I = A^{[n]} - A^-X - A^-XA^- - (A^+X + A^+XA^+) \cap A^{[n]}$.*

(v) *Case of infix codes: $R = I - (IA^+ + A^+I + A^+IA^+) \cap A^{[n]}$, where $I = A^{[n]} - XA^- + A^-X + A^-XA^- - (XA^+ + A^+X + A^+XA^+) \cap A^{[n]}$.*

(vi) *Case of p-subinfix codes: $R = I - S_2(h^{-1}(I) \cap A_\#^{[n-1]}\{\#\}) \cap A^{[n]}$, where $I = A^{[n]} - h(S_1(X) \cap A_\#^*\{\#\}) - S_2(h^{-1}(X) \cap A_\#^{[n-1]}\{\#\}) \cap A^{[n]}$.*

(vii) *Case of s-subinfix codes: $R = I - S_2(h^{-1}(I) \cap \{\#\}A_\#^{[n-1]}) \cap A^{[n]}$, where $I = A^{[n]} - h(S_1(X) \cap \{\#\}A_\#^*) - S_2(h^{-1}(X) \cap \{\#\}A_\#^{[n-1]}) \cap A^{[n]}$.*

(viii) *Case of subinfix codes:* $R = I - S_2(h^{-1}(I) \cap (\{\#\}A_\#^{[n-1]} \cup A_\#^{[n-1]}\{\#\})) \cap A^{[n]}$, *where* $I = A^{[n]} - h(S_1(X) \cap (\{\#\}A_\#^* \cup A_\#^*\{\#\})) - S_2(h^{-1}(X) \cap (\{\#\}A_\#^{[n-1]} \cup A_\#^{[n-1]}\{\#\})) \cap A^{[n]}$.

(ix) *Case of p-sucyperinfix codes:* $R = I - S_2(h^{-1}(\sigma(I)) \cap A_\#^{[n-1]}\{\#\}) \cap A^{[n]}$, *where* $I = A^{[n]} - \sigma(h(S_1(X) \cap A_\#^*\{\#\})) - S_2(h^{-1}(\sigma(X)) \cap A_\#^{[n-1]}\{\#\}) \cap A^{[n]}$.

(x) *Case of s-sucyperinfix codes:* $R = I - S_2(h^{-1}(\sigma(I)) \cap \{\#\}A_\#^{[n-1]}) \cap A^{[n]}$, *where* $I = A^{[n]} - \sigma(h(S_1(X) \cap \{\#\}A_\#^*)) - S_2(h^{-1}(\sigma(X)) \cap \{\#\}A_\#^{[n-1]}) \cap A^{[n]}$.

(xi) *Case of sucyperinfix codes:* $R = I - S_2(h^{-1}(\sigma(I)) \cap (\{\#\}A_\#^{[n-1]} \cup A_\#^{[n-1]}\{\#\})) \cap A^{[n]}$, *where* $I = A^{[n]} - \sigma(h(S_1(X) \cap (\{\#\}A_\#^* \cup A_\#^*\{\#\}))) - S_2(h^{-1}(\sigma(X)) \cap (\{\#\}A_\#^{[n-1]} \cup A_\#^{[n-1]}\{\#\})) \cap A^{[n]}$.

(xii) *Case of p-superinfix codes:* $R = I - S_2(h^{-1}(\pi(I)) \cap A_\#^{[n-1]}\{\#\}) \cap A^{[n]}$, *where* $I = A^{[n]} - \pi(h(S_1(X) \cap A_\#^*\{\#\})) - S_2(h^{-1}(\pi(X)) \cap A_\#^{[n-1]}\{\#\}) \cap A^{[n]}$.

(xiii) *Case of s-superinfix codes:* $R = I - S_2(h^{-1}(\pi(I)) \cap \{\#\}A_\#^{[n-1]}) \cap A^{[n]}$, *where* $I = A^{[n]} - \pi(h(S_1(X) \cap \{\#\}A_\#^*)) - S_2(h^{-1}(\pi(X)) \cap \{\#\}A_\#^{[n-1]}) \cap A^{[n]}$.

(xiv) *Case of superinfix codes:* $R = I - S_2(h^{-1}(\pi(I)) \cap (\{\#\}A_\#^{[n-1]} \cup A_\#^{[n-1]}\{\#\})) \cap A^{[n]}$, *where* $I = A^{[n]} - \pi(h(S_1(X) \cap (\{\#\}A_\#^* \cup A_\#^*\{\#\}))) - S_2(h^{-1}(\pi(X)) \cap (\{\#\}A_\#^{[n-1]} \cup A_\#^{[n-1]}\{\#\})) \cap A^{[n]}$.

(xv) *Case of p-hypercodes:* $R = I - S_2(h^{-1}(I) \cap (A_\#^*\{\#\}A_\#^* - \{\#\}^+A^+) \cap A_\#^{[n]}) \cap A^{[n]}$, *where* $I = A^{[n]} - h(S_1(X) \cap (A_\#^*\{\#\}A_\#^* - \{\#\}^+A^+)) - S_2(h^{-1}(X) \cap (A_\#^*\{\#\}A_\#^* - \{\#\}^+A^+) \cap A_\#^{[n]}) \cap A^{[n]}$.

(xvi) *Case of s-hypercodes:* $R = I - S_2(h^{-1}(I) \cap (A_\#^*\{\#\}A_\#^* - A^+\{\#\}^+) \cap A_\#^{[n]}) \cap A^{[n]}$, *where* $I = A^{[n]} - h(S_1(X) \cap (A_\#^*\{\#\}A_\#^* - A^+\{\#\}^+)) - S_2(h^{-1}(X) \cap (A_\#^*\{\#\}A_\#^* - A^+\{\#\}^+) \cap A_\#^{[n]}) \cap A^{[n]}$.

(xvii) *Case of hypercodes:* $R = I - S_2(h^{-1}(I) \cap A_\#^*\{\#\}A_\#^* \cap A_\#^{[n]}) \cap A^{[n]}$, *where* $I = A^{[n]} - h(S_1(X) \cap A_\#^*\{\#\}A_\#^*) - S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^* \cap A_\#^{[n]}) \cap A^{[n]}$.

(xviii) *Case of sucypercodes:* $R = I - \sigma(S_2(h^{-1}(I) \cap A_\#^*\{\#\}A_\#^* \cap A_\#^{[n]}) \cap A^{[n]})$, *where* $I = A^{[n]} - h(S_1(\sigma(X)) \cap A_\#^*\{\#\}A_\#^*) - \sigma(S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^* \cap A_\#^{[n]}) \cap A^{[n]})$.

(xix) *Case of supercodes:* $R = I - \pi(S_2(h^{-1}(I) \cap A_\#^*\{\#\}A_\#^* \cap A_\#^{[n]}) \cap A^{[n]})$, *where* $I = A^{[n]} - h(S_1(\pi(X)) \cap A_\#^*\{\#\}A_\#^*) - \pi(S_2(h^{-1}(X) \cap A_\#^*\{\#\}A_\#^* \cap A_\#^{[n]}) \cap A^{[n]})$.

Example 6.3. Consider the p-subinfix code $X = \{a^2, ba^2\}$ over $A = \{a, b\}$. Since $\max X = 3$, by Corollary 6.2(vi), R_X can be computed by the formula:

$$R = I - S_2(h^{-1}(I) \cap A_\#^{[2]}\{\#\}) \cap A^{[3]},$$

where $I = A^{[3]} - h(S_1(X) \cap A_{\#}^*\{\#\}) - S_2(h^{-1}(X) \cap A_{\#}^{[2]}\{\#\}) \cap A^{[3]}$. We may now compute R_X step by step as follows.

$S_1(X) \cap A_{\#}^*\{\#\} = \{a\#, \#^2, ba\#, b\#^2, \#a\#, \#^3\}$;
$h(S_1(X) \cap A_{\#}^*\{\#\}) = \{1, a, b, ba\}$;
$h^{-1}(X) \cap A_{\#}^{[2]}\{\#\} = \{a^2\#\}$;
$S_2(h^{-1}(X) \cap A_{\#}^{[2]}\{\#\}) \cap A^{[3]} = a^2A = \{a^3, a^2b\}$;
$I = A^{[3]} - \{1, a, b, ba\} - \{a^3, a^2b\} = \{a^2, ab, b^2, aba, ab^2, ba^2, bab, b^2a, b^3\}$;
$h^{-1}(I) \cap A_{\#}^{[2]}\{\#\} = \{a^2\#, ab\#, b^2\#\}$;
$S_2(h^{-1}(I) \cap A_{\#}^{[2]}\{\#\}) \cap A^{[3]} = a^2A + abA + b^2A = \{a^3, a^2b, aba, ab^2, b^2a, b^3\}$;
$R = I - \{a^3, a^2b, aba, ab^2, b^2a, b^3\} = \{a^2, ab, b^2, ba^2, bab\}$.

Example 6.4. For the sucypercode $X = \{acb, a^2b^2, cabc\}$ over $A = \{a, b, c\}$, in a similar way, using formulas in Corollary 6.2(xviii) with $n = 4$ instead, we obtain the maximal sucypercode $R = \{a^3, a^2c, aca, acb, bac, b^3, b^2c, bcb, ca^2, cba, cb^2, c^3, a^2b^2, abab, ab^2a, abc^2, ba^2b, baba, b^2a^2, bc^2a, cabc, c^2ab\}$, which contains X.

7. Tree Representations

Sometimes a graph representation of codes defined by binary relations seems to be useful. As in the case of prefix codes, it facilitates the construction of examples of codes and in many cases of maximal codes containing a given code. Moreover, as we see below, it provides more intuition in understanding and proving facts.

First, with every transitive binary relation $\prec$ on A^* we associate an infinite oriented graph as follows. The alphabet A is totally ordered, and words of equal length are ordered lexicographically. Each word represents a node of the graph. Words of small length are to the left of words of greater length, and words of equal length are positioned vertically according to lexical ordering. For any nodes u, v, there is an edge $u \to v$ if and only if $u \prec v$ and there is no w such that $u \prec w \prec v$. Throughout we restrict ourselves to length-increasing relations only. Thus, the corresponding graph is a tree in some large sense, called the *tree of A^* with respect to $\prec$*, denoted by $\mathcal{T}(A^*, \prec)$, or simply $\mathcal{T}$ when there is no risk of confusion. In the case of the relation $\prec_p$ this is nothing but the *literal representation* of A^* [1]. From now on, we refer indifferently to a node in $\mathcal{T}$ and the word it represents. For example, one may speak of the length of a node which means the length of the word it represents, etc.

To a given subset X of A^* we associate a subtree of $\mathcal{T}(A^*, \prec)$ as follows. We keep just the nodes representing the words of X and of $\{u \mid u \prec v, v \in X\}$, and all related edges. The tree obtained in this way is the *tree of X with respect to $\prec$*, denoted by $\mathcal{T}(X, \prec)$ (see Figure 2).

A set X of nodes is *node-independent* if there is no path from one node to another. The set X is *maximal* if it is included properly in no node-independent

set. In other words, a node-independent set X is maximal if it becomes no more a node-independent set by adding a new node. The following fact establishes relationship between the codes defined by $\prec$, that is the independent sets with respect to $\prec$, and the node-independent sets of the tree $\mathcal{T}(A^*, \prec)$.

Proposition 7.1. *Let $\prec$ be a length-increasing transitive binary relation on A^* which defines a class $C_\prec$ of codes. Let $\mathcal{T}(A^*, \prec)$ be the tree of A^* with respect to $\prec$. Then, for any $X \subseteq A^*$, X is a (maximal) code in $C_\prec$ if and only if the corresponding nodes in $\mathcal{T}(A^*, \prec)$ constitute a (maximal, respectively) node-independent set.*

Proof. We first prove that for any $u, v \in A^*$, $u \prec v$ if and only if there exists in $\mathcal{T}$ a path $p : u = u_0 \to u_1 \to \cdots \to u_k = v$ from u to v, $k \geq 1$. Indeed, let $u \prec v$ and $|v| = n$. If $n = 1$ then $v = a$ for some $a \in A$, and $u = 1$. Then, we have $p : u = 1 \to a = v$. Let now $n > 1$ and suppose the claim true for all the words v' of length less than n. If there is no word w such that $u \prec w \prec v$ then, by definition, there exists an edge $u \to v$ which we may take as the path p. Otherwise, let w be the longest word such that $u \prec w \prec v$. Then, there exists an edge $w \to v$. Since $|w| < n$, by induction hypothesis, there exists some path $p' : u \to^* w$, from u to w. It suffices to put $p : u \to^* w \to v$. Conversely, let $p : u = u_0 \to u_1 \to \cdots \to u_k = v$ be a path of length k in $\mathcal{T}$, $k \geq 1$. With $k = 1$ then p is an edge, namely $u \to v$, and by definition, $u \prec v$, Let $k > 1$ and assume the claim true for all $k' < k$. Put $p' : u = u_0 \to \cdots \to u_{k-1}$. On one hand, by induction hypothesis, $u \prec u_{k-1}$. On the other hand, $u_{k-1} \to v$ implies $u_{k-1} \prec v$. The transitivity of $\prec$ implies $u \prec v$.

Now, using the above fact, it is easy to see: X is in $C_\prec \Leftrightarrow$ X is an independent set with respect to $\prec \Leftrightarrow$ any two words $u, v \in A$ are not in the relation $\prec \Leftrightarrow$ for any $u, v \in X$, there is no path from one to other in $\mathcal{T}$ $\Leftrightarrow$ X is a node-independent set of $\mathcal{T}$. Thus, the claim "X is a code in $C_\prec$ if and only if the corresponding nodes in $\mathcal{T}(A^*, \prec)$ constitute a node-independent set" holds true. The remaining claim follows immediately from the above claim and definitions. ∎

Remark 7.2. Proposition 7.1 can be used to obtain another proof [18], more intuitive, of item (i) in Theorem 3.1.

As seen in the examples above (Section 6), even for a small code X, computing R_X is not simple in practice. It is however much easier when using a tree representation of A^* with respect to $\prec$ as shown in the following example.

Example 7.3. Consider again the p-subinfix code $X = \{a^2, ba^2\}$. As $\max X = 3$, for finding R_X we need only consider the tree $\mathcal{T}(A^3, \prec_{p.si})$ of the full uniform code A^3 with respect to $\prec_{p.si}$. By virtue of Theorem 3.1 and Proposition 7.1, R_X can be obtained by applying the following algorithm: First, mark the nodes represented by the words in X (namely: aa, baa). Then, delete all the nodes depending on X (namely: $1, a, b, ba, aaa, aab$). Next, among the rest (namely:

$ab, bb, aba, abb, bab, bba, bbb$) keep just the minimal nodes (namely: ab, bb, bab), which together with X constitute R_X, i.e. $R_X = \{a^2, ab, b^2, ba^2, bab\}$ (see Figure 2).

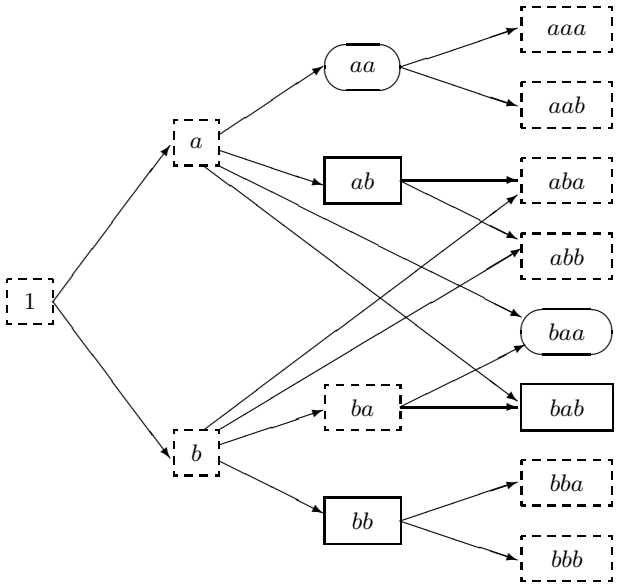

Fig. 2 Computing $R_X, X = \{a^2, ba^2\}$, by using the tree $\mathcal{T}(A^3, \prec_{p.si})$

Acknowlegements. The authors express their sincere thanks to the referees whose valuable remarks and comments helped them very much in correcting and improving the paper.

References

1. J. Berstel, D. Perrin, *Theory of Codes*, Academic Press, New York, 1985.
2. V. Bruyère, M. Latteux, Variable-length maximal codes, *Theoret. Comput. Sci.*, **98** (1992), 321–337.
3. V. Bruyère, L. Wang, L. Zhang, On completion of codes with finite deciphering delay, *Eur. J. Comb.*, **11** (1990), 513–521.
4. A. Ehrenfeucht, G. Rozenberg, Each regular code is included in a maximal regular code, *RAIRO Theoret. Inform. Appl.*, **20** (1986), 89–96.
5. J. E. Hopcroft, R. Motwani, J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, Addison-Wesley, Reading, Mass., 2001.
6. M. Ito, H. Jürgensen, H. J. Shyr, G. Thierrin, Outfix and infix codes and related classes of languages, *J. Comput. Syst. Sci.*, **43** (1991), 484–508.
7. M. Ito, G. Thierrin, Congruences, infix and cohesive prefix codes, *Theoret. Comput. Sci.*, **136** (1994), 471–485.

8. H. Jürgensen, S. Konstatinidis, Codes, In: G. Rozenberg, A. Salomaa (eds.), *Handbook of Formal Languages*, Springer, Berlin, 1997, 511–607.
9. H. Jürgensen, S. S. Yu, Solid codes, *J. Inform. Process. Cybernet., EIK* **26** (1990), 563–574.
10. H. Jürgensen, S. S. Yu, Relations on free monoides, their independent sets, and codes, *Int. J. Comput. Math.*, **40** (1991), 17–46.
11. N. H. Lam, Finite maximal infix codes, *Semig. Forum*, **61** (2000), 346–356.
12. M. Lothaire, *Combinatorics on Words*, Encyclopedia of Math. and its Appl., Addison-Wesley Publishing Company, 1983.
13. A. A. Markov, An example of an independent system of words which cannot be included in a finite complete system, *Math. Z.*, **1** (1967), 87–90 (in Russian).
14. D. Perrin, Completing biprefix codes, *Theoret. Comput. Sci.*, **28** (1984), 329–336.
15. A. Restivo, On codes having no finite completion, *Discrete Math.*, **17** (1977), 309–316.
16. A. Restivo, S. Salemi, T. Sportelli, Completing codes, *RAIRO Theoret. Inform. Appl.*, **23** (1989), 135–147.
17. H. J. Shyr, *Free Monoids and Languages*, Hon Min Book Company, Taichung, 1991.
18. D. L. Van, Embedding problem for codes defined by binary relations, *Preprint 98/A22*, Inst. of Math., Hanoi, 1998.
19. D. L. Van, On a class of hypercodes, In: M. Ito, T. Imaoka (eds.), *Words, Languages and Combinatorics III*. World Scientific, 2003, 171–183.
20. D. L. Van, K. V. Hung, On codes defined by binary relations, Part II: Vector characterizations and maximality (submitted).
21. D. L. Van, K. V. Hung, P. T. Huy, Codes and length-increasing binary relations, *Lecture Notes in Computer Science*, **3772** (2005), 29–48.
22. S. S. Yu, *Codes and n-ary Relations*, Ph. D. Thesis, The Univ. of Western Ontario, 1990.
23. S. S. Yu, *Languages and Codes*, Tsang Hai Book Publishing Company, Taichung, Taiwan, 2005.
24. L. Zhang, Z. Shen, Completion of recognizable bifix codes, *Theoret. Comput. Sci.*, **145** (1995), 345–355.