Journal of Computer Science and Cybernetics, V.32, N.3 (2016), 273-283
DOI 10.15625/1813-9663/32/3/9350

CHARACTERIZATIONS FOR SEVERAL CLASSES OF ALTERNATIVE CODES

NGO THI HIEN

Hanoi University of Science and Technology; hien.ngothi@hust.edu.vn

Abstract. Alternative codes, an extension of the notion of ordinary codes, have been first introduced and considered by P. T. Huy et al. in 2004. In this paper, we consider some new subclasses of alternative codes. In particular, characteristic properties and a hierarchy of such codes are established. Also, algorithms to test for these classes of codes are proposed.

Keywords. Alternative code, norm alternative code, left norm alternative code, two-sided alternative code, left alternative code, strict alternative code.

Mathematics Subject Classification (2010): 94A45, 68Q45.

1. INTRODUCTION

The theory of length-variable codes has been initiated by M. P. Schützenberger in the 1950s and then developed by many others. This theory has now become a part of theoretical computer science and of formal languages, in particular. A code is a language such that every text encoded by words of the language can be decoded in a unique way or, in other words, every coded message admits only one factorization into code-words. Codes are useful in many areas of application such as information processing, data compression, cryptography, information transmission and so on. For background of the theory of codes we refer to [1, 7, 9].

As mentioned above, the definition of codes is essentially based on unambiguity of the (catenation) product of words. Different modifications of such a product may lead to different extensions of the notion of codes. Such an approach has been proposed by P. T. Huy et al. which deals with the so-called *even alternative codes* and their several subclasses [6, 12]. They demonstrated the importance characterizations (Lemmas 3-5) as well as algorithms to test for these classes (Lemma 6). For simplicity, even alternative codes, weak left alternative codes, weak right alternative codes and alternative codes in [6, 12] are called simply respectively *alternative codes, left alternative codes, right alternative codes* and *strict alternative codes* instead. Recently, in [4, 11, 12], the authors proposed some classes of codes which concern alternative codes. Several interesting characteristic properties and algorithms to test for such codes were established.

In this paper, some new subclasses of alternative codes are introduced and considered. In Section 2 necessary definitions are recalled, and several facts useful in the sequel are shown. Section 3 introduces four new classes of codes which are subclasses of alternative codes and can be defined by alternative factorizations on two non-empty languages. Also some characteristic properties and relationships of these classes are considered. The section ends with a hierarchy of alternative codes and their subclasses. Section 4 establishes four algorithms to test for new classes of codes. The final section discusses the problem investigations and presents a summary of the results obtained. Our work is motivated by the idea to define alternative codes by alternative factorizations on two non-empty languages, and the way to solve the problem for alternative codes as well as their well-known subclasses [6, 11, 12].

2. PRELIMINARIES

Let A throughout be a finite alphabet and A^* the set of all the words over A. The empty word is denoted by ϵ and A^+ stands for $A^* \setminus \{\epsilon\}$. The number of all the occurrences of letters in a word u is the *length* of u, denoted by $|u|$. Any subset of A^* is a *language* over A. A language X is a *code* 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 $i = 1, \ldots, n$. Since $\epsilon.\epsilon = \epsilon$, a code never contains the empty word ϵ.

A word u is called an *infix* (a *prefix*, a *suffix*) of a word v if there exist words x, y such that $v = xuy$ (resp. $v = uy$, $v = xu$). The infix (prefix, suffix) is *proper* if $xy \neq \epsilon$ (resp. $y \neq \epsilon$, $x \neq \epsilon$). The set of non-empty proper prefixes of a word w is denoted Pref(w). We denote by Pref(X) the set of non-empty proper prefixes of words in $X \subseteq A^*$. For non-empty proper suffixes, we use the notations Suff(w) and Suff(X). The *reversal* of a word $w = a_1 a_2 \ldots a_n$, where $a_1, a_2, \ldots, a_n$ are letters, is the word $\widetilde{w} = a_n a_{n-1} \ldots a_1$. Similarly, for $X \subseteq A^*$, we denote $\widetilde{X} = \{\widetilde{x} \mid x \in X\}$.

A language $X \subseteq A^+$ is a *prefix code* (*suffix code*) if no word in X is a proper prefix (resp. proper suffix) of another word in it, and X is a *bifix code* if it is both a prefix code and a suffix code. Prefix codes, suffix codes and bifix codes play a fundamental role in the theory of codes (see [1, 5, 9, 10]).

For $X, Y \subseteq A^*$, the *product* of X and Y is the set $XY = \{xy \mid x \in X, y \in Y\}$. The product is said to be *unambiguous* if for each $z \in XY$ there is exactly one pair $(x, y) \in X \times Y$ such that $z = xy$. We also use the notations

$$X^0 = \{\epsilon\}, \quad X^{n+1} = X^n X \ (n \geq 0);$$
$$(XY)^+ = (XY) \cup (XY)^2 \cup (XY)^3 \cup \ldots$$

For $w \in A^*$, we define

$$w^{-1}X = \{u \in A^* \mid wu \in X\}, \ Xw^{-1} = \{u \in A^* \mid uw \in X\}.$$

This notation is extended as usual to sets by

$$X^{-1}Y = \bigcup_{x \in X} x^{-1}Y, \ XY^{-1} = \bigcup_{y \in Y} Xy^{-1}.$$

From now on, unless otherwise specified, we always consider X and Y are non-empty subsets of A^+. We say that $u_1 u_2 \ldots u_n$, $n \geq 2$, is an *alternative factorization on* (X, Y) if $u_i \in X$ implies $u_{i+1} \in Y$ and $u_i \in Y$ implies $u_{i+1} \in X$ for all $i = 1, 2, \ldots, n-1$. Two alternative factorizations $u_1 u_2 \ldots u_n$ and $v_1 v_2 \ldots v_m$ on (X, Y) are said to be *left similar* (*right similar*) if they begin (resp. end) with words in the same set X or Y, namely $u_1, v_1 \in X$ or $u_1, v_1 \in Y$ (resp. $u_n, v_m \in X$ or $u_n, v_m \in Y$). Two alternative factorizations on (X, Y) are called *similar* if they are both left similar and right similar.

Definition 1. A pair (X, Y) is called an *alternative code* (a *left alternative code*, a *right alternative code*) over A if no word in A^+ admits two different similar (resp. left similar, right similar) alternative factorizations on (X, Y), and (X, Y) is a *strict alternative code* over A if every word in A^+ admits at most one alternative factorization on (X, Y).

The classes of alternative codes, left alternative codes, right alternative codes and strict alternative codes are denoted respectively by C_A, C_{lA}, C_{rA} and C_{sA}. Over any alphabet consisting of at least two letters, the following holds true:

$$C_{sA} \subset C_{lA} \cap C_{rA}, C_{lA} \cup C_{rA} \subset C_A.$$

For more details of alternative codes and their subclasses we refer to [4, 6, 11, 12].

Now we formulate, in the form of lemmas, several facts which will be useful in the sequel.

Lemma 1 (Sardinas-Patterson's criterion [8], see also [1, 2, 3]). *Let X be a subset of A^+, and let*

$$U_1 = X^{-1}X \setminus \{\epsilon\}, \quad U_{n+1} = X^{-1}U_n \cup U_n^{-1}X \text{ for } n \geq 1.$$

Then, X is a code if and only if none of the sets U_n defined above contains the empty word ϵ.

A simple characterization of the unambiguity of a product of two languages is given in the following lemma.

Lemma 2 ([6], see also [12]). *A product XY is unambiguous if and only if $X^{-1}X \cap YY^{-1} \setminus \{\epsilon\} = \emptyset$.*

The following results claim some basic characterizations for alternative codes and their subclasses.

Lemma 3 ([6], see also [12]). *A pair (X, Y) is an alternative code if and only if XY is a code and the product XY is unambiguous.*

Lemma 4 ([12]). *A pair (X, Y) is a left (right) alternative code if and only if it is an alternative code and $(XY)^+X^{-1} \cap (XY)^+ = \emptyset$ (resp. $Y^{-1}(XY)^+ \cap (XY)^+ = \emptyset$).*

Lemma 5 ([6], see also [12]). *A pair (X, Y) is a strict alternative code if and only if the four following conditions are satisfied:*

(i) $(X, Y) \in C_A$;

(ii) $(XY)^+X^{-1} \cap (XY)^+ = \emptyset$;

(iii) $Y^{-1}(XY)^+ \cap (XY)^+ = \emptyset$;

(iv) $(XY)^+ \cap (YX)^+ = \emptyset$.

Moreover the conditions (i), (ii), (iii) and (iv) are independent.

Lemma 6 ([12], see also [11]). *If X and Y are regular languages, then there exists an algorithm to decide*

(i) *XY is a code in $O(8^{m+n})$ worst-case time;*

(ii) $X^{-1}X \cap YY^{-1} \setminus \{\epsilon\} = \emptyset$ *in* $O(m^2n^2)$ *worst-case time;*

(iii) $(XY)^+X^{-1} \cap (XY)^+ = \emptyset$ *in* $O(8^{m+n})$ *worst-case time;*

(iv) $Y^{-1}(XY)^+ \cap (XY)^+ = \emptyset$ *in* $O(8^{m+n})$ *worst-case time;*

(v) $(XY)^+ \cap (YX)^+ = \emptyset$ *in* $O(16^{m+n})$ *worst-case time;*

where m *and* n *are respectively the number of states in the deterministic finite automata recognizing* X *and* Y*.*

3. SOME CHARACTERIZATIONS

We introduce in this section some new subclasses of alternative codes. Some characterizations for alternative codes and their subclasses are established. Let us begin with a new class of codes which is a subclass of left alternative codes and right alternative codes. We say that two alternative factorizations on (X, Y) are *weak similar* if they are left similar or right similar.

Definition 2. A pair (X, Y) is called a *two-sided alternative code* over A if no word in A^+ admits two different weak similar alternative factorizations on (X, Y). In other words, (X, Y) is a two-sided alternative code if it is both a left alternative code and a right alternative code.

Let us take an example.

Example 1. Consider the sets $X = \{aab, baa\}$ and $Y = \{aa\}$ over $A = \{a, b\}$. It is easy to check that (X, Y) is both a left alternative code and a right alternative code, and hence it is a two-sided alternative code. However, (X, Y) is not a strict alternative code because the word *aabaa* has two distinct factorizations,

$$aabaa = (aab)(aa) = (aa)(baa).$$

As an immediate consequence of Definition 2 and Lemma 4, we obtain the following characterization for two-sided alternative codes.

Theorem 1. *A pair* (X, Y) *is a two-sided alternative code if and only if the three following conditions are satisfied:*

(i) $(X, Y) \in C_A$*;*

(ii) $(XY)^+X^{-1} \cap (XY)^+ = \emptyset$*;*

(iii) $Y^{-1}(XY)^+ \cap (XY)^+ = \emptyset$*.*

Moreover the conditions (i), (ii) and (iii) are independent.

Two alternative factorizations $u_1u_2 \ldots u_n$ and $v_1v_2 \ldots v_m$ on (X, Y) are called *dissimilar* if they both begin and end with words in different sets, namely both $u_1 \in X, v_1 \in Y$ or $u_1 \in Y, v_1 \in X$ and $u_n \in X, v_m \in Y$ or $u_n \in Y, v_m \in X$. A pair (X, Y) is said to be *norm form* if no word in A^+ admits two different dissimilar alternative factorizations on (X, Y).

Definition 3. A pair (X, Y) is called a *norm alternative code* (*left norm alternative code*, *right norm alternative code*) over A if it is an alternative code (resp. a left alternative code, a right alternative code) which has norm form.

The classes of two-sided alternative codes, left norm alternative codes, right norm alternative codes and norm alternative codes are denoted respectively by C_{tA}, C_{lnA}, C_{rnA} and C_{nA}.

Example 2. Consider the sets $X_1 = \{b^n a \mid n \geq 1\}$, $Y_1 = \{a, aba\}$, $X_2 = \{a, ba\}$, $Y_2 = \{b, ab\}$ over $A = \{a, b\}$. Then, we have $X_1Y_1 = \{b^n aa, b^n aaba \mid n \geq 1\}$ and $X_2Y_2 = \{ab, aab, bab, baab\}$. By Lemma 1, X_1Y_1 is a code because $U_1 = \{ba\}$, $U_2 = \{a, aba\}$, $U_3 = \emptyset$. Since X_1Y_1 is a code and X_1 is a prefix code, by Proposition 2 (we shall see below), (X_1, Y_1) is an alternative code. On the other hand, it is easy to check that no word in A^+ admits two different dissimilar alternative factorizations on (X_1, Y_1), that means (X_1, Y_1) is norm form. Thus, (X_1, Y_1) is a norm alternative code. However, (X_1, Y_1) is not a left norm alternative code because the word $baaba$ has two different left similar alternative factorizations on (X_1, Y_1), $baaba = (ba)(a)(ba) = (ba)(aba)$.

Next, by Proposition 2, (X_2, Y_2) is an alternative code because X_2 and X_2Y_2 are prefix codes. It is not difficult to check that no word in A^+ admits two different left similar and two different dissimilar alternative factorizations on (X_2, Y_2). Hence, (X_2, Y_2) is a left norm alternative code. However, (X_2, Y_2) is not a right norm alternative code because the word $baab$ has two different right similar alternative factorizations on (X_2, Y_2), $baab = (ba)(ab) = (b)(a)(ab)$.

In an entirely symmetric manner, we also obtain $(\widetilde{X}_1, \widetilde{Y}_1)$ is a norm alternative code but it is not a right norm alternative code, and $(\widetilde{X}_2, \widetilde{Y}_2)$ is a right norm alternative code not being a left norm alternative code.

Remark 1. For two different alternative factorizations on (X, Y), there are only separately possible cases as the following:

$[XX, XX]$ (1) $[YY, YY]$ (2) $[XY, XY]$ (3) $[YX, YX]$ (4) $[XX, XY]$ (5)

$[YY, YX]$ (6) $[XX, YX]$ (7) $[XY, YY]$ (8) $[XY, YX]$ (9) $[XX, YY]$ (10)

where $[XY, YX]$ denotes that the first alternative factorization begin with a word in X, end with a word in Y, and the second alternative factorization begin with a word in Y, end with a word in X. Then, by Definitions 1-3, the following holds true:

The class of codes	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
C_A	−	−	−	−	+	+	+	+	+	+
C_{lA}	−	−	−	−	−	−	+	+	+	+
C_{rA}	−	−	−	−	+	+	−	−	+	+
C_{tA}	−	−	−	−	−	−	−	−	+	+
C_{nA}	−	−	−	−	+	+	+	+	−	−
C_{lnA}	−	−	−	−	−	−	+	+	−	−
C_{rnA}	−	−	−	−	+	+	−	−	−	−
C_{sA}	−	−	−	−	−	−	−	−	−	−

where $+$ and $-$ denote that these classes accept and reject alternative factorizations which have forms like $(1), (2), \ldots, (10)$ respectively.

Next, we exhibit characterizations for norm (left norm, right norm) alternative codes, and some relationships between alternative codes and their subclasses. For this, we need two more auxiliary propositions.

Proposition 1. *The following holds true*

(i) (X, Y) *is norm form if and only if* $(XY)^+ \cap (YX)^+ = \emptyset$*;*

(ii) (X, Y) *is norm form if Pref*$(X) \cap$ *Pref*$(Y) = \emptyset$ *or Suff*$(X) \cap$ *Suff*$(Y) = \emptyset$*.*

Proof. (i) Assume (X, Y) is norm form. If $(XY)^+ \cap (YX)^+ \neq \emptyset$, then there exists $u \in (XY)^+ \cap (YX)^+$ such that $u = x_1y_1x_2y_2 \ldots x_ny_n = y'_1x'_1y'_2x'_2 \ldots y'_mx'_m$ with $x_i, x'_j \in X, y_i, y'_j \in Y, 1 \leq i \leq n, 1 \leq j \leq m$. Thus, the word u admits two different dissimilar alternative factorizations on (X, Y), a contradiction. Hence, $(XY)^+ \cap (YX)^+ = \emptyset$.

Conversely, suppose the contrary that (X, Y) is not norm form. Then, there exists $w \in A^+$ such that $w = u_1u_2 \ldots u_n = v_1v_2 \ldots v_m$, where both $u_1 \in X, v_1 \in Y$ or $u_1 \in Y, v_1 \in X$ and $u_n \in X, v_m \in Y$ or $u_n \in Y, v_m \in X$. We consider separately two possible cases.

Case 1: $u_1, v_m \in X, v_1, u_n \in Y$ or $u_1, v_m \in Y, v_1, u_n \in X$. Then, evidently $w \in (XY)^+ \cap (YX)^+$.

Case 2: $u_1, u_n \in X, v_1, v_m \in Y$ or $u_1, u_n \in Y, v_1, u_m \in X$. Then, we have $ww = u_1u_2 \ldots u_nv_1v_2 \ldots v_m = v_1v_2 \ldots v_mu_1u_2 \ldots u_n$. Therefore, $ww \in (XY)^+ \cap (YX)^+$.

Thus, in all the cases we have proved that $(XY)^+ \cap (YX)^+ \neq \emptyset$, a contradiction. So, (X, Y) is norm form.

(ii) Suppose Pref$(X) \cap$Pref$(Y) = \emptyset$ or Suff$(X) \cap$Suff$(Y) = \emptyset$. If $(XY)^+ \cap (YX)^+ \neq \emptyset$, then there exists a word $u \in (XY)^+ \cap (YX)^+$ such that $u = x_1y_1x_2y_2 \ldots x_ny_n = y'_1x'_1y'_2x'_2 \ldots y'_mx'_m$ with $x_i, x'_j \in X, y_i, y'_j \in Y, 1 \leq i \leq n, 1 \leq j \leq m$. Therefore, Pref$(x_1) \cap$ Pref$(y'_1) \neq \emptyset$ and Suff$(x'_m) \cap$Suff$(y_n) \neq \emptyset$. This implies that Pref$(X) \cap$Pref$(Y) \neq \emptyset$ and Suff$(X) \cap$Suff$(Y) \neq \emptyset$, a contradiction. Thus, $(XY)^+ \cap (YX)^+ = \emptyset$. Hence, by the item (i) of the proposition, (X, Y) is norm form. □

Proposition 2. *If* XY *is a code and* X *is a prefix code or* Y *is a suffix code, then* (X, Y) *is an alternative code.*

Proof. If X is a prefix code, then $X^{-1}X = \{\epsilon\}$, and therefore $X^{-1}X \cap YY^{-1} \setminus \{\epsilon\} = \emptyset$. By Lemma 2, the product XY is unambiguous. Hence, by Lemma 3, (X, Y) is an alternative code. Similarly for the case when Y is a suffix code. □

As an immediate consequence of Definition 3 and Proposition 1(i), we can now formulate the following result which resumes characterizations for norm (left norm, right norm) alternative codes.

Theorem 2. *The following holds true*

(i) (X, Y) *is a norm alternative code if and only if it is an alternative code and* $(XY)^+ \cap (YX)^+ = \emptyset$*;*

(ii) (X, Y) *is a left norm alternative code if and only if it is a left alternative code and* $(XY)^+ \cap (YX)^+ = \emptyset$*;*

(iii) (X, Y) *is a right norm alternative code if and only if it is a right alternative code and* $(XY)^+ \cap (YX)^+ = \emptyset$*.*

The following result shows some relationships between alternative codes and their subclasses.

Theorem 3. *Let (X, Y) be an alternative code over A. Then, we have*

(i) *If XY is a bifix (prefix, suffix) code and Pref$(X) \cap$Pref$(Y) = \emptyset$ or Suff$(X) \cap$Suff$(Y) = \emptyset$, then (X, Y) is a strict (resp. left norm, right norm) alternative code;*

(ii) *If XY is a bifix (prefix, suffix) code, then (X, Y) is a two-sided (resp. left, right) alternative code;*

(iii) *If Pref$(X) \cap$ Pref$(Y) = \emptyset$ or Suff$(X) \cap$ Suff$(Y) = \emptyset$, then (X, Y) is a norm alternative code.*

Proof. We treat only the item (i). The items (ii) and (iii) follow immediately from the item (i) of the theorem, Lemma 4 and Theorems 1-2.

Assume that (X, Y) is an alternative code such that XY is a bifix code and Pref$(X) \cap$ Pref$(Y) = \emptyset$ or Suff$(X) \cap$ Suff$(Y) = \emptyset$. Firstly, by Proposition 1, $(XY)^+ \cap (YX)^+ = \emptyset$. Secondly, we have $(XY)^+X^{-1} \cap (XY)^+ = \emptyset$ because XY is a prefix code. Indeed, suppose the contrary. Then, there exist $u \in (XY)^+X^{-1} \cap (XY)^+$ and $x \in X$ such that $u = x_1y_1x_2y_2 \dots x_ny_n$ and $ux = x'_1y'_1x'_2y'_2 \dots x'_my'_m$ with $x_i, x'_j \in X, y_i, y'_j \in Y, 1 \leq i \leq n, 1 \leq j \leq m$. Therefore, $x_1y_1x_2y_2 \dots x_ny_nx = x'_1y'_1x'_2y'_2 \dots x'_my'_m$. Thus there exists i such that x_iy_i is a proper prefix of $x'_iy'_i$ or vice versa, or $x'_my'_m$ is a proper prefix of xy with $y \in Y$, which contradicts the fact that XY is a prefix code. Similarly, we also obtain $Y^{-1}(XY)^+ \cap (XY)^+ = \emptyset$ when XY is a suffix code. Hence, (X, Y) is a strict (left norm, right norm) alternative code by Lemma 5 (resp. by Lemma 4 and Theorem 2(ii), by Lemma 4 and Theorem 2(iii)). □

Remark 2. According to Proposition 2, the items (i) and (ii) of Theorem 3 still hold if we use the assumption X is a prefix code or Y is a suffix code instead of (X, Y) is an alternative code. However, under the stated assumption, the item (iii) of Theorem 3 only holds if assume furthermore that XY is a code, namely (X, Y) is a norm alternative code if XY is a code and Pref$(X) \cap$ Pref$(Y) = \emptyset$ or Suff$(X) \cap$ Suff$(Y) = \emptyset$.

Note that Theorem 3 and Remark 2 provide us with a tool to check whether a give alternative code (more generally a pair language) is in subclasses of alternative codes or not.

Example 3. We again consider the sets $X = \{a^2b, ba^2\}, Y = \{a^2\}$ in Example 1, $X_1 = \{b^na \mid n \geq 1\}, Y_1 = \{a, aba\}, X_2 = \{a, ba\}, Y_2 = \{b, ab\}$ in Example 2, and $X_3 = \{aa, ab\}, Y_3 = \{bb\}$.

Firstly, we have $XY = \{a^2ba^2, ba^4\}$, Pref$(X) \cap$ Pref$(Y) =$ Suff$(X) \cap$ Suff$(Y) = \{a\}$, and X, XY are bifix codes. By Theorem 3 and Remark 2, (X, Y) is a two-sided alternative code not being a strict alternative code.

Next, it is easy to see that Pref$(X_1) \cap$ Pref$(Y_1) = \{b^n \mid n \geq 1\} \cap \{a, ab\} = \emptyset$, $X_2Y_2 = \{ab, a^2b, bab, ba^2b\}$, Pref$(X_2) \cap$ Pref$(Y_2) = \{b\} \cap \{a\} = \emptyset$, and X_2, X_2Y_2 are prefix codes. Thus, by Theorem 3 and Remark 2, (X_1, Y_1) is a norm alternative code, and (X_2, Y_2) is a left norm alternative code.

Finally, it is clear that $X_3Y_3 = \{a^2b^2, ab^3\}$, Pref$(X_3) \cap$ Pref$(Y_3) = \{a\} \cap \{b\} = \emptyset$, and X_3, X_3Y_3 are bifix codes. By Remark 2, (X_3, Y_3) is a strict alternative code.

As an immediate consequence of Definitions 1-3, Lemmas 3-5, Theorems 1-2 and Remark 1, we can now formulate the following result which resumes relative positions of the classes of codes under consideration.

Theorem 4. *Over any alphabet consisting of at least two letters, the following holds true*

(i) $C_{sA} \subset C_{lnA}, C_{sA} \subset C_{rnA}, C_{sA} \subset C_{tA}$,
$C_{sA} = C_{nA} \cap C_{tA} = C_{lnA} \cap C_{rA} = C_{rnA} \cap C_{lA} = C_{lA} \cap C_{nA} \cap C_{rA}$
$= C_{lnA} \cap C_{tA} = C_{tA} \cap C_{rnA} = C_{rnA} \cap C_{lnA} = C_{lnA} \cap C_{rnA} \cap C_{tA}$;

(ii) $C_{lnA} \subset C_{lA}, C_{lnA} \subset C_{nA}$, $C_{rnA} \subset C_{rA}, C_{rnA} \subset C_{nA}$, $C_{tA} \subset C_{lA}$,
$C_{tA} \subset C_{rA}, C_{lnA} = C_{lA} \cap C_{nA}, C_{rnA} = C_{rA} \cap C_{nA}, C_{tA} = C_{lA} \cap C_{rA}$;

(iii) $C_{lA} \subset C_A, C_{nA} \subset C_A$, $C_{rA} \subset C_A$.

Remark 3. By Theorem 4, $C_{sA} = C_{nA} \cap C_{tA}$. However, C_{nA} and C_{tA} are not comparable by inclusion. Indeed, the two-sided alternative code (X, Y) in Example 1 which has the word $a^2ba^2 \in (XY)^+ \cap (YX)^+$ that means $(XY)^+ \cap (YX)^+ \neq \emptyset$. By Theorem 2, (X, Y) is not a norm alternative code. On the other hand, the norm alternative code (X_1, Y_1) in Example 2 is easily verified to be not a two-sided alternative code.

By virtue of Theorem 4 and Remark 3, the relative positions of the classes of codes under consideration can be illustrated in the Figure 1, where the arrow → stands for a strict inclusion. It is worthy to note that if we restrict ourselves to considering only one-letter alphabets then we have $C_{sA} = C_{ln} = C_{rnA} = C_{nA} = \emptyset$, whereas the remaining classes represented in the Figure 1 coincide.

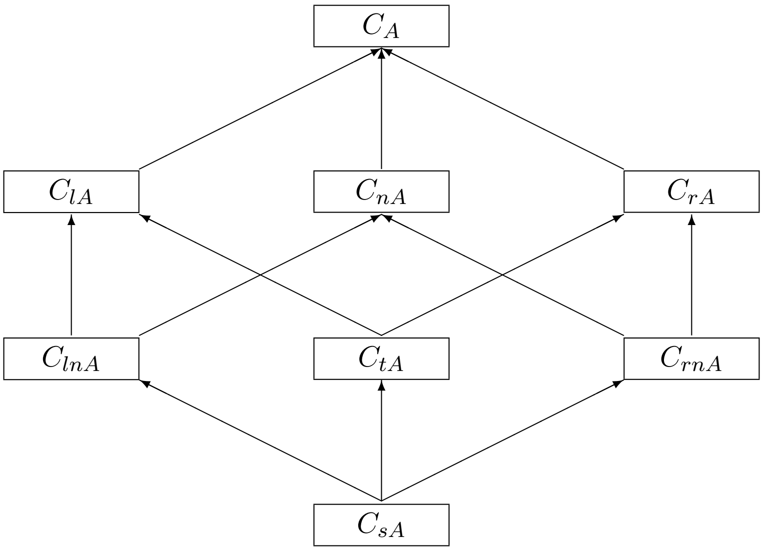

Figure 1: Relative positions of alternative codes and their subclasses

4. A TEST FOR ALTERNATIVE CODES

It is not always easy to verify that a given pair language is an alternative code. The problem is more difficult for subclasses of alternative codes. In the case where X and Y are finite languages, or more generally they are regular languages, the following result plays a fundamental role which helps us to establish algorithms to test for new subclasses of alternative codes.

Theorem 5. *If X and Y are regular languages, then there exists an algorithm to decide the pair (X, Y) is*

(i) *a two-sided alternative code in $O(8^{m+n})$ worst-case time;*

(ii) *a norm (norm left, norm right) alternative code in $O(16^{m+n})$ worst-case time;*

where m and n are respectively the number of states in the deterministic finite automata recognizing X and Y.

Proof. It follows immediately from Lemma 6 and Theorems 1-2 with note that always there exist algorithms for testing the emptiness of regular languages. □

From Lemma 6, Theorems 1-2 and Theorem 5, we can exhibit the following algorithms to test for two-sided alternative codes and norm (left norm, right norm) alternative codes.

```
(A test for two-side alternative codes)
Input: $X$ and $Y$ are regular languages of $A^+$.
Output: $(X, Y)$ is a two-sided alternative code or not.
1. If $(X, Y) \notin C_A$ then goto Step 5.
2. If $(XY)^+X^{-1} \cap (XY)^+ \neq \emptyset$ then goto Step 5.
3. If $Y^{-1}(XY)^+ \cap (XY)^+ \neq \emptyset$ then goto Step 5.
4. $(X, Y)$ is a two-sided alternative code; STOP.
5. $(X, Y)$ is not a two-sided alternative code; STOP.
```

```
(A test for norm alternative codes)
Input: $X$ and $Y$ are regular languages of $A^+$.
Output: $(X, Y)$ is a norm alternative code or not.
1. If $(X, Y) \notin C_A$ then goto Step 4.
2. If $(XY)^+ \cap (YX)^+ \neq \emptyset$ then goto Step 4.
3. $(X, Y)$ is a norm alternative code; STOP.
4. $(X, Y)$ is not a norm alternative code; STOP.
```

```
(A test for left norm alternative codes)
Input: $X$ and $Y$ are regular languages of $A^+$.
Output: $(X, Y)$ is a left norm alternative code or not.
1. If $(X, Y) \notin C_A$ then goto Step 5.
2. If $(XY)^+X^{-1} \cap (XY)^+ \neq \emptyset$ then goto Step 5.
3. If $(XY)^+ \cap (YX)^+ \neq \emptyset$ then goto Step 5.
4. $(X, Y)$ is a left norm alternative code; STOP.
5. $(X, Y)$ is not a left norm alternative code; STOP.
```

Let us take some examples.

```
(A test for right norm alternative codes)
Input: $X$ and $Y$ are regular languages of $A^+$.
Output: $(X, Y)$ is a right norm alternative code or not.
1. If $(X, Y) \notin C_A$ then goto Step 5.
2. If $Y^{-1}(XY)^+ \cap (XY)^+ \neq \emptyset$ then goto Step 5.
3. If $(XY)^+ \cap (YX)^+ \neq \emptyset$ then goto Step 5.
4. $(X, Y)$ is a right norm alternative code; STOP.
5. $(X, Y)$ is not a right norm alternative code; STOP.
```

Example 4. Consider the regular languages $X = ab^+, Y = ba^+$ over $A = \{a, b\}$. It is easy to see that $XY = ab^+ba^+$ and the sets X, Y, XY are suffix codes. By Algorithm 1, we have:

1. By Proposition 2, (X, Y) is an alternative code because X and XY are suffix codes.
2. Since $(ab^+ba^+)(ab^+)^{-1} = \emptyset$, it follows that $(XY)^+X^{-1} \cap (XY)^+ = \emptyset$.
3. Since $(ba^+)^{-1}(ab^+ba^+) = \emptyset$, it follows that $Y^{-1}(XY)^+ \cap (XY)^+ = \emptyset$.
4. (X, Y) is a two-sided alternative code, and the algorithm ends.

Example 5. Consider the sets $X = \{b\}, Y = \{ab, bab\}$ over $A = \{a, b\}$. We have $XY = \{bab, bbab\}, YX = \{abb, babb\}$. Evidently, the sets X, Y and XY are prefix codes. By Algorithm 2, we have:

1. By Proposition 2, (X, Y) is an alternative code because X and XY are prefix codes.

2. Because $ab \in \text{Suff}(u), bb \in \text{Suff}(v)$ for all $u \in (XY)^+, v \in (YX)^+$, it follows that $(XY)^+ \cap (XY)^+ = \emptyset$.

3. (X, Y) is a norm alternative code, and the algorithm ends.

Example 6. We again consider the sets $X = \{b\}, Y = \{ab, bab\}$ in Example 5. By Algorithm 4, we may perform as follows.

1. By the above, (X, Y) is an alternative code.

2. We have $Y^{-1}(XY)^+ \cap (XY)^+ \neq \emptyset$ because $bab \in Y^{-1}(XY)^2 \cap XY$. Hence, (X, Y) is not a right norm alternative code, and the algorithm ends.

5. CONCLUSIONS

The purpose of this paper was to deal with the development of the class of alternative codes. Four new classes of codes, which are subclasses of alternative codes, were introduced and considered. Some characteristics and relationships of these classes were proposed (Theorems 1-4). One of them provided us with a tool to check whether a given alternative code (more generally a pair language) is in subclasses of alternative codes or not. Algorithms, with a power size complexity (Theorem 5), to test for new subclasses of alternative codes were established (Algorithms 1-4).

In future works, we hope we can find some better algorithms and many interesting problems for alternative codes as well as their subclasses.

ACKNOWLEDGMENT

The author would like to thank the colleagues in Seminar Mathematical Foundation of Computer Science at Institute of Mathematics, Vietnam Academy of Science and Technology for attention to the work. Especially, the author expresses her sincere thanks to Prof. Do Long Van and Dr. Kieu Van Hung for their useful discussions.

REFERENCES

[1] J. Berstel and D. Perrin, *Theory of Codes.* Academic Press, New York, 1985.

[2] N. D. Han and P. T. Huy, "On unambiguity of languages related to codes," in *Future Information Technology, Application, and Service*, J. P. James, C. M. L. Victor, L. W. Cho, and S. Taeshik, Ed. Springer Netherlands, pp. 32–38, 2012.

[3] N. D. Han, H. N. Vinh, D. Q. Thang, and P. T. Huy, "Quadratic algorithms for testing of codes and $\Diamond$-codes," *Fundamenta Informaticae*, vol. 130, no. 2, pp. 163-177, 2014.

[4] N. T. Hien and D. L. Van, "Codes induced by alternative codes," *Acta Mathematica Vietnammica*, 16 pages, 2017 (submitted).

[5] K. V. Hung and D. L. Van, "Prime decomposition problem for several kinds of regular codes," *Lecture Notes in Computer Science*, vol. 4281, pp. 213–227, 2006.

[6] P. T. Huy and V. T. Nam, "Alternative codes and pre-context codes," in *Proceedings of the 7th National conference: Selected problems about IT and Telecommunication*, 2004, pp. 188–197 (in Vietnamese).

[7] H. Jürgensen and S. Konstantinidis, "Codes," in *Handbook of Formal Languages*, G. Rozenberg and A. Salomaa, Ed. Springer, Berlin, pp. 511–607, 1997.

[8] A. A. Sardinas and C. W. Patterson, "A necessary and sufficient condition for the unique decomposition of coded messages," *IRE International Convention Record*, vol. 8, pp. 104–108, 1953.

[9] H. J. Shyr, *Free Monoids and Languages.* Hon Min Book Company, Taichung, 1991.

[10] D. L. Van and K. V. Hung, "On codes defined by binary relations, Part I: Embedding Problem, " *Vietnam Journal of Mathematics*, vol. 39, no. 2, pp. 159–176, 2011.

[11] H. N. Vinh, P. T. Huy, and D. L. Van, "Extension of codes and algorithms for alternative codes and codes of bounded words," *Journal of Computer Science and Cybernetics*, vol. 26, no. 4, pp. 301–311, 2010 (in Vietnamese).

[12] H. N. Vinh, V. T. Nam, and P. T. Huy, "Codes based on unambiguous products," *Lecture Notes in Computer Science*, vol. 6423, pp. 252–262, 2010.

Received on March 21, 2017
Revised on May 26, 2017