Codes induced by alternative codes

Ngo Thi Hien∗ · Do Long Van†

Received: date / Accepted: date

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. As seen below, every alternative code, in its turn, defines an ordinary code. Such codes are called codes induced by alternative codes or alt-induced codes, for short. In this paper we consider these alt-induced codes and subclasses of them. In particular, characteristic properties of such codes are established, and an algorithm to check whether a finite code is alt-induced or not is proposed.

Keywords Code, alt-induced code, strong alt-induced code, alternative code, strong 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 [1,9,13].

∗Hanoi University of Science and Technology
No. 1 Dai Co Viet Road, Hanoi, Vietnam
E-mail: hien.ngothi@hust.edu.vn

†Institute of Mathematics, Vietnam Academy of Science and Technology
No. 18 Hoang Quoc Viet Road, Hanoi, Vietnam
E-mail: dlvan@math.ac.vn
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 subclasses \[7,14\]. Throughout this paper, for simplicity, even alternative codes (see \[7\]) are called simply alternative codes instead. As seen later, an alternative code is nothing but a pair \((X, Y)\) of languages such that \(XY\) is a code and the product \(XY\) is unambiguous (Lemma \[7\]). We say that the code \(XY\) is induced by the alternative code \((X, Y)\). A code is said to be an alt-induced code if there exists an alternative code which induces it.

Obviously, every alt-induced code is a code. There exist, however, codes not being alt-induced ones. Characterizing alt-induced codes among codes is, therefore, a meaningful and interesting question. Given a code \(Z\), to prove that \(Z\) is an alt-induced code, again by Lemma \[7\] we must first show that \(Z\) can be factorized into a product of two languages \(X\) and \(Y\), \(Z = XY\), and then show that the product \(XY\) is unambiguous. The first phase of this seemingly relates to a more general question of factorising a language, a code in particular, into a product of simpler languages (see \[1,2,4,6,8,12,15\] and the references therein).

The paper is organized as follows. Section 2 presents basic notions, notations and facts which will be useful in the sequel. In Section 3 the notion of alt-induced code is introduced. Several basic properties of these codes are shown (Proposition \[1\] Proposition \[2\] Proposition \[3\]). Characterizations for prefix (suffix, bifix) alt-induced codes are established (Theorem \[4\]). A special subclass of alt-induced codes, namely that of strong alt-induced codes is considered in Section 4. Characterizations and properties of these codes are established (Proposition \[5\] Theorem \[6\] Theorem \[7\] Theorem \[8\]). Finally, Section 5 is reserved to finite alt-induced codes. The main problem is to answer the question whether a given finite code is alt-induced (Theorem \[9\]). The section ends with an algorithm (Algorithm FIC), with a exponential time complexity (Theorem \[10\]), for testing whether a given finite code is alt-induced or not.

2 Preliminaries

Let \(A\) throughout be a finite alphabet, i.e. a non-empty finite set of symbols, which are called letters. Let \(A^*\) be the set of all finite words over \(A\). The empty word is denoted by \(\varepsilon\) and \(A^*\) stands for \(A^* \setminus \{\varepsilon\}\). 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 \(\varepsilon \varepsilon = \varepsilon\), a code never contains the empty word \(\varepsilon\).
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 $xy \neq \varepsilon$ (resp., $y \neq \varepsilon, x \neq \varepsilon$). The set of proper prefixes of a word $w$ is denoted by $\text{Pref}(w)$. We denote by $\text{Pref}(X)$ the set of all proper prefixes of the words in $X \subseteq A^*$. The notations $\text{Suff}(w)$ and $\text{Suff}(X)$ are defined in a similar way.

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 exists exactly one pair $(x, y) \in X \times Y$ such that $z = xy$. We also use the notations $X^0 = \{\varepsilon\}$, $X^{n+1} = X^nX$ ($n \geq 0$).

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

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

These notations are extended to sets in a natural way:

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

Let $(X, Y)$ be a pair of non-empty subsets of $A^+$, and let $u_1, u_2, \ldots, u_n \in X \cup Y, n \geq 2$. We say that $u_1u_2\ldots u_n$ 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 $u_1u_2\ldots u_n$ and $v_1v_2\ldots v_m$ on $(X, Y)$ are said to be **similar** if they both begin and end with words in the same set $X$ or $Y$.

**Definition 1** Let $X$ and $Y$ be two non-empty subsets of $A^+$. The pair $(X, Y)$ is called an **alternative code** if no word in $A^+$ admits two different similar alternative factorizations on $(X, Y)$.

For more details of alternative codes and their subclasses we refer to [5, 7, 14].

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

**Lemma 1** ([1]) If $X \subseteq A^+$ is a code, then $X^n$ is a code for all integers $n \geq 1$.

**Lemma 2** (Sardinas-Patterson’s criterion [11], see also [11, 13]) Let $X$ be a subset of $A^+$, and let

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

Then, $X$ is a code if and only if none of the sets $U_n$ defined above contains the empty word $\varepsilon$. 
Recall that, 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. A prefix code (suffix code, bifix code) \( X \) is **maximal** over \( A \) if it is not properly contained in another prefix code (resp., suffix code, bifix code) over \( A \). Prefix codes, suffix codes and bifix codes play a fundamental role in the theory of codes (see [1,6,13]).

As a consequence of Proposition 4.1 in [1] we have

**Lemma 3** Let \( X \) and \( Y \) be non-empty subsets of \( A^+ \). Then

(i) If \( X \) and \( Y \) are prefix codes (maximal prefix codes), then \( XY \) is a prefix code (resp., maximal prefix code);
(ii) If \( XY \) is a prefix code (suffix code), then \( Y \) is a prefix code (resp., \( X \) is a suffix code);
(iii) If \( X \) is a prefix code and \( XY \) is a maximal prefix code, then \( X \) and \( Y \) are both maximal prefix codes.

A subset \( X \) of \( A^* \) is **thin** if there exists at least one word \( w \in A^* \) which is not an infix of any word in \( X \), i.e. \( X \cap A^*wA^* = \emptyset \). Evidently, for any \( X,Y \subseteq A^* \), if \( XY \) is thin then \( X \) and \( Y \) are thin also. Conversely we have

**Lemma 4** ([1], page 65) For any \( X,Y \subseteq A^* \), if \( X \) and \( Y \) are both thin then their product \( XY \) is thin too.

Concerning the maximality of thin codes we have

**Lemma 5** ([1], Proposition 2.1, page 145) Let \( X \) be a thin subset of \( A^+ \). Then, \( X \) is a maximal bifix code if and only if \( X \) is both a maximal prefix code and a maximal suffix code.

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

**Lemma 6** ([7], see also [14]) Let \( X \) and \( Y \) be non-empty subsets of \( A^+ \). Then, the product \( XY \) is unambiguous if and only if \( X^{-1}X \cap YY^{-1}\setminus\{\varepsilon\} = \emptyset \).

The following result claims a basic characterization for alternative codes.

**Lemma 7** ([7], see also [14]) Let \( X \) and \( Y \) be non-empty subsets of \( A^+ \). Then, \((X,Y)\) is an alternative code if and only if \( XY \) is a code and the product \( XY \) is unambiguous.

### 3 Codes induced by alternative codes

We introduce and consider in this section a new class of codes concerning alternative codes which are called alt-induced codes. Characterizations for prefix (suffix, bifix) alt-induced codes are established.
Definition 2 A subset \( Z \) of \( A^+ \) is called a code induced by an alternative code (alt-induced code, for short) if there is an alternative code \((X, Y)\) over \( A \) such that \( Z = XY \).

As usual, a language \( Z \) is a prefix (suffix, bifix) alt-induced code if it is an alt-induced code as well as a prefix (resp., suffix, bifix) code.

Let us take an example.

Example 1 Consider the sets \( X = \{ab, abba\} \) and \( Y = \{b\} \) over \( A = \{a, b\} \). By virtue of Lemma 6 and Lemma 7, it is not difficult to check that \((X, Y)\) is an alternative code over \( A \). Hence \( Z = XY = \{abb, abbab\} \) is an alt-induced code and therefore it is a suffix alt-induced code.

Our purpose, in the rest of the paper, is to answer the question when a given code \( Z \) is an alt-induced code. Let us begin with some simple cases.

Remark 1 Evidently, a code is not an alt-induced code if it contains at least one word with the length one.

Proposition 1 Let \( Z \) be a code over \( A \) such that \( Z = XY \) with \( \emptyset \neq X, Y \subseteq A^+ \). If \( X \) is a prefix code or \( Y \) is a suffix code, then \( Z \) is an alt-induced code.

Proof If \( X \) is a prefix code, then \( X^{-1}X = \{\varepsilon\} \), and therefore \( X^{-1}X \cap YY^{-1} \setminus \{\varepsilon\} = \emptyset \). By Lemma 6 the product \( XY \) is unambiguous. Thus, by Lemma 7 \((X, Y)\) is an alternative code, and hence \( Z = XY \) is an alt-induced code. Similarly for the case when \( Y \) is a suffix code.

\( \Box \)

Corollary 1 Let \( Z \subseteq A^+ \) be a code such that all the words of \( Z \) have length greater than or equal to 2. Then, if all the words of \( Z \) begin (end) with the same letter in \( A \), then \( Z \) is an alt-induced code.

Proof Suppose that all the words of \( Z \) begin with the same letter \( a \) in \( A \). Then, we have \( Z = \{a\}Y \) with \( \emptyset \neq Y \subseteq A^+ \). Since \( \{a\} \) is a prefix code, by Proposition 1 \( Z \) is an alt-induced code. The argument is similar for the other case of the corollary.

\( \Box \)

Remark 2 If \( A \) is a one-letter alphabet, \(|A| = 1\), then, as well-known, every subset \( Z \) of \( A^+ \) is a code if and only if it is a singleton, \( Z = \{w\} \). Therefore, by Remark 1 and Corollary 1 \( Z \) is an alt-induced code if and only if \( w \) has the length greater than or equal to 2, \(|w| \geq 2\).

Corollary 2 Let \( Z \) be a code over \( A \) such that \( Z = XY \) with \( \emptyset \neq X, Y \subseteq A^+ \). If \(|X| = 2 \) or \(|Y| = 2\), then \( Z \) is an alt-induced code.

Proof We treat the case \(|X| = 2\). If \( X \) is a prefix code, then by Proposition 1 \( Z \) is an alt-induced code. Otherwise, since \(|X| = 2\), we have \( X = \{x, xu\} \) for some \( x, u \in A^+ \). Therefore, \( Z = XY = \{x, xu\}Y = \{x\}Y, uY \), that is all the words of \( Z \) begin with the same letter in \( A \). Thus, by Corollary 1 \( Z \) is also an alt-induced code. For the case \(|Y| = 2\), the argument is similar.

\( \Box \)
Proposition 2 If $Z \subseteq A^+$ is an alt-induced code, then $Z^n$ is also an alt-induced code for all integers $n \geq 1$.

Proof Since $Z$ is a code, by Lemma 1, $Z^n$ is a code for all integers $n \geq 1$. Now we prove by induction that $Z^n$ is also an alt-induced code. Indeed, for $n = 1$ it is true by assumption. Suppose the assertion is already true for $n - 1$ with $n \geq 2$. Put $X = Z^{n-1}, Y = Z$. Then, on one hand, as mentioned above, $XY = Z^n$ is a code. On the other hand, the unambiguity of the product $XY$ follows directly from the fact that $Z$ is a code. Thus, by Lemma 7, $(X, Y)$ is an alternative code and therefore $Z^n = XY$ is an alt-induce code. \(\Box\)

The following example shows that the product of two alt-induced codes is not, in general, an alt-induced code.

Example 2 Let us consider the sets $Z = \{aa, baa\} = \{a, ba\}\{a\}$ and $Z' = \{aa, aab\} = \{a\}\{a, ab\}$ which are, as easily seen, alt-induced codes over $A = \{a, b\}$. Put

$$R = ZZ' = \{a^4, a^4b, ba^4, ba^4b\}.$$  

The word $w = a^4ba^4b$, for example, has two distinct factorizations: $w = (a^4)(ba^4b) = (a^4b)(a^4b)$ on $R$. Thus $R$ is not a code, and therefore not an alt-induced code either.

For prefix (suffix, bifix) alt-induced codes, we have however

Proposition 3 The product of two prefix (suffix, bifix) alt-induced codes is a prefix (resp., suffix, bifix) alt-induced code.

Proof We treat only the case of prefix alt-induced codes. For the other cases the arguments are similar. Let $Z$ and $Z'$ be two prefix alt-induced codes. Then, by Lemma 3(i), $ZZ'$ is a prefix code. Since $Z$ is a prefix code, by Proposition 1, $ZZ'$ is a prefix alt-induced code. \(\Box\)

Next, we exhibit characterizations for prefix (suffix, bifix) alt-induced codes. For this, we need two more auxiliary propositions.

Proposition 4 Let $X$ and $Y$ be non-empty subsets of $A^+$.

(i) If $X$ and $Y$ are prefix (maximal prefix) codes, then $XY$ and $YX$ are prefix (resp., maximal prefix) codes, and the products $XY$ and $YX$ are unambiguous;

(ii) If $X$ and $Y$ are suffix (maximal suffix) codes, then $XY$ and $YX$ are suffix (resp., maximal suffix) codes, and the products $XY$ and $YX$ are unambiguous;

(iii) If $X$ and $Y$ are bifix (maximal bifix thin) codes, then $XY$ and $YX$ are bifix (resp., maximal bifix thin) codes, and the products $XY$ and $YX$ are unambiguous.
Proof (i) If \( X \) and \( Y \) are prefix (maximal prefix) codes, then, by Lemma 3(i), \( XY \) and \( YX \) are prefix (resp., maximal prefix) codes. Again because \( X \) is a prefix code, it follows that \( X^{-1}X = \{\varepsilon\} \), and therefore \( X^{-1}X \cap YY^{-1}\setminus\{\varepsilon\} = \emptyset \).

By Lemma 6 the product \( XY \) is unambiguous. Similarly, since \( Y \) is a prefix code, it follows that \( YX \) is unambiguous.

(ii) The proof is similar as above in the item (i).

(iii) If \( X \) and \( Y \) are bifix codes, then, by (i) and (ii) of the proposition, \( XY \) and \( YX \) are bifix codes and the products \( XY \) and \( YX \) are unambiguous. Now assume moreover that \( X \) and \( Y \) are maximal bifix codes and thin. Then, by Lemma 5 \( X \) and \( Y \) are both maximal prefix codes as well as maximal suffix codes. Again by (i) and (ii) of the proposition, \( XY \) and \( YX \) are both maximal prefix codes as well as maximal suffix codes. By Lemma 4 \( XY \) and \( YX \) are thin. Again by Lemma 5 they are both maximal bifix codes.

\(\Box\)

As the converse of Proposition 4 we have

Proposition 5 Let \( X \) and \( Y \) be non-empty subsets of \( A^+ \).

(i) If \( XY \) and \( YX \) are prefix (maximal prefix) codes, then \( X \) and \( Y \) are prefix (resp., maximal prefix) codes;

(ii) If \( XY \) and \( YX \) are suffix (maximal suffix) codes, then \( X \) and \( Y \) are suffix (resp., maximal suffix) codes;

(iii) If \( XY \) and \( YX \) are bifix (maximal bifix thin) codes, then \( X \) and \( Y \) are bifix (resp., maximal bifix thin) codes.

Proof (i) If \( XY \) and \( YX \) are prefix codes, then, by Lemma 3(ii), \( Y \) and \( X \) are prefix codes. If \( XY \) and \( YX \) are maximal prefix codes, then, again by Lemma 3(ii) and then by Lemma 3(iii), \( X \) and \( Y \) are maximal prefix codes.

(ii) The proof is similar as for the item (i).

(iii) Firstly, by (i) and (ii) of the proposition, \( X \) and \( Y \) are bifix codes if so are \( YX \) and \( XY \). Now, suppose that \( XY \) and \( YX \) are maximal bifix codes and thin. Then, by Lemma 5 \( XY \) and \( YX \) are both maximal prefix and maximal suffix codes. Again by (i) and (ii) of the proposition, \( X \) and \( Y \) are both maximal prefix and maximal suffix codes. The thinness of \( XY \) implies evidently the thinness of \( X \) and \( Y \). Hence, again by Lemma 5 \( X \) and \( Y \) are maximal bifix codes.

\(\Box\)

As an immediate consequence of Proposition 4 and Proposition 5 we can now formulate the following result which resumes characterizations for prefix (suffix, bifix) alt-induced codes.

Theorem 1 Let \( X \) and \( Y \) be non-empty subsets of \( A^+ \).

(i) \( XY \) and \( YX \) are prefix (maximal prefix) alt-induced codes if and only if \( X \) and \( Y \) are prefix (resp., maximal prefix) codes;

(ii) \( XY \) and \( YX \) are suffix (maximal suffix) alt-induced codes if and only if \( X \) and \( Y \) are suffix (resp., maximal suffix) codes;

(iii) \( XY \) and \( YX \) are bifix (maximal bifix thin) alt-induced codes if and only if \( X \) and \( Y \) are bifix (resp., maximal bifix thin) codes.
Note that Theorem 1 provides us with a tool to construct (maximal) alt-induced codes. For example, just by taking product of two prefix (maximal prefix) codes, we obtain a prefix (maximal prefix) alt-induced code. As such, in some sense, the class of alt-induced codes is quite large.

**Example 3** We consider the sets

\[ X = \{ a^n b a \mid n \geq 1 \}, Y = \{ b^m a \mid m \geq 1 \} \]

over \( A = \{ a, b \} \). Clearly, \( X \) and \( Y \) are prefix codes. Hence, by Theorem 1(i),

\[ XY = \{ a^n b a b^m a \mid n, m \geq 1 \} \]

and

\[ Y X = \{ b^m a^n b a \mid n \geq 2, m \geq 1 \} \]

are prefix alt-induced codes.

### 4 Strong alt-induced codes

In this section we consider a special subclass of alt-induced codes which is introduced in the following definition.

**Definition 3** Let \( X \) and \( Y \) be non-empty subsets of \( A^+ \).

(i) An alternative code \( (X, Y) \) is called a strong alternative code if it satisfies the following conditions

\[ X^{-1}(XY) \subseteq Y \quad (1), \quad (XY)Y^{-1} \subseteq X \quad (2). \]

(ii) An alt-induced code \( Z \) over \( A \) is called a strong alt-induced code if there exists a strong alternative code \( (X, Y) \) generating it, \( Z = XY \).

**Remark 3** It is easy to check that the conditions (1) and (2) in Definition 3 are equivalent to the conditions

\[ X^{-1}(XY) = Y \quad (1'), \quad (XY)Y^{-1} = X \quad (2') \]

and these, in its turn, are equivalent to

\[ \forall x \in X : x^{-1}(XY) = Y \quad (1''), \quad \forall y \in Y : (XY)y^{-1} = X \quad (2''). \]

**Example 4** Consider the sets

\[ X = \{ a^n b a \mid n \geq 1 \}, Y = \{ b, bab \} \]

over \( A = \{ a, b \} \). Then, we have \( XY = \{ a^n b b, a^n bab \mid n \geq 1 \} \).

By Lemma 2, \( XY \) is a code because \( U_1 = \{ ab \}, U_2 = \{ b, bab \}, U_3 = \emptyset \). Since \( X \) is a prefix code, by Proposition 1, the product \( XY \) is unambiguous. Hence, by Lemma 2 \( (X, Y) \) is an alternative code, and therefore \( XY \) is an alt-induced code. On the other hand, it is easy to see that, for all \( n \geq 1, a^n bba \in (XY)Y^{-1} \setminus X \), which means that the inclusion (2) in Definition 3 is not satisfied. Thus, \( (X, Y) \) is not a strong alternative code, and therefore \( XY \) is not a strong alt-induced code.

The following result is basic in characterizing the strong alternative codes and, therefore, the strong alt-induced codes.

**Proposition 6** Let \( X \) and \( Y \) be non-empty subsets of \( A^+ \).

(i) If \( X \) is a prefix code, \( Y \) is a suffix code and \( XY \) is a code, then \( (X, Y) \) is a strong alternative code;
(ii) If \((X, Y)\) is a strong alternative code, then \(X\) is a prefix code and \(Y\) is a suffix code.

Proof (i) Suppose \(X\) is a prefix code, \(Y\) is a suffix code and \(XY\) is a code. First, because \(X\) is a prefix code, the product \(XY\) is unambiguous. Thus, by Lemma \(7\) \((X, Y)\) is an alternative code. Now, we will prove that \((X, Y)\) is, moreover, a strong alternative code, that is we have to show that the inclusions (1) and (2) in Definition \(3\) must be satisfied. Indeed, assume the contrary that the inclusion (1), for example, is not true. Then there must exist some word \(u\) such that \(u \in X^{-1}(XY)\) but \(u \notin Y\). From \(u \in X^{-1}(XY)\), it follows that there exist \(x, x' \in X\) and \(y \in Y\) such that \(xu = x'y\). This, again because \(X\) is prefix, implies \(u = y\) and therefore \(u \in Y\), a contradiction. Thus, the inclusion (1) must be true. Similarly, the assumption that the inclusion (2) is not satisfied will also lead to a contradiction. As a consequence it must be true and therefore \((X, Y)\) is really a strong alternative code.

(ii) Suppose \((X, Y)\) is a strong alternative code. Now assume that \(X\) is not a prefix code. Then, there must exist \(x, x' \in X\) such that \(x = x'u\) with \(u \neq \varepsilon\). Choosing any \(y \in Y\) we have \(xy = x'uy\) with \(u \neq \varepsilon\). This implies \(uy \in X^{-1}(XY)\). If \(uy \in Y\) then, from the unambiguity of the product \(XY\), it follows \(uy = y\), a contradiction. Thus, \(uy \notin Y\) which means that the inclusion (1) in Definition \(3\) is not satisfied. This contradicts the assumption that \((X, Y)\) is a strong alternative code. Thus, \(X\) must be a prefix code. The fact that \(Y\) is a suffix code can be proved in a similar way, where the inclusion (2) is used instead of (1).

As an immediate consequence of Definition \(3\) and Proposition \(6\) we obtain the following characterization for strong alt-induced codes as well as strong alternative codes.

**Theorem 2** Let \(X\) and \(Y\) be non-empty subsets of \(A^+\). The following conditions are equivalent:

(i) \(XY\) is a strong alt-induced code;
(ii) \((X, Y)\) is a strong alternative code;
(iii) \(X\) is a prefix code, \(Y\) is a suffix code and \(XY\) is a code.

**Example 5** Consider the sets \(X = \{a^n b \mid n \geq 1\}, Y = \{b, bba\}\) over \(A = \{a, b\}\). Evidently, \(X\) is a prefix code not being a suffix code while \(Y\) is a suffix code not being a prefix code. Then, by virtue of Lemma \(2\) the product \(XY = \{a^n bb, a^n bba \mid n \geq 1\}\) is easily verified to be a code: \(U_1 = \{ba\}, U_2 = \emptyset\). According to Theorem \(2\) \((X, Y)\) is a strong alternative code and \(XY\) is a strong alt-induced code.

The following example shows that in Theorem \(2\) the requirement “\(XY\) is a code” in the item (iii) is essential.

**Example 6** Consider the sets \(X = \{a^n b \mid n \geq 1\}, Y = \{ba^m \mid m \geq 1\}\) over \(A = \{a, b\}\). Clearly, \(X\) is a prefix code not being a suffix code while \(Y\) is a suffix code...
not being a prefix code. But their product \( Z = XY = \{a^n b b a^m \mid n, m \geq 1\} \) is not a code because the word \( w = abbaaabbba \), for example, has two distinct factorizations on \( Z \):

\[
  w = (abba)(aabba) = (abbaa)(abba).
\]

As a consequence of Theorem 2, Lemma 3 and Lemma 5 we have the following characterization for prefix (suffix, bifix) strong alt-induced codes.

**Theorem 3** Let \( X \) and \( Y \) be non-empty subsets of \( A^+ \).

(i) \( XY \) is a prefix (maximal prefix) strong alt-induced code if and only if \( X \) is a prefix (resp., maximal prefix) code and \( Y \) is a bifix code (resp., \( Y \) is both a maximal prefix code and a bifix code);

(ii) \( XY \) is a suffix (maximal suffix) strong alt-induced code if and only if \( X \) is a bifix code (resp., \( X \) is both a maximal suffix code and a bifix code) and \( Y \) is a suffix (resp., maximal suffix) code;

(iii) \( XY \) is a bifix (maximal bifix thin) strong alt-induced code if and only if \( X \) and \( Y \) are bifix (resp., maximal bifix thin) codes.

**Proof**

(i) Suppose \( XY \) is a prefix strong alt-induced code. Then, by Theorem 2, \( X \) is a prefix code, \( Y \) is a suffix code and \( XY \) is a prefix code. Therefore, by Lemma 3(ii), \( Y \) must be a prefix code and therefore a bifix code. Now, suppose that \( XY \) is moreover a maximal prefix strong alt-induced code. Then, \( XY \) is, in particular, a maximal prefix code and, by the above, \( X \) is a prefix code and \( Y \) is a bifix code. Therefore, by Lemma 3(iii), both \( Y \) and \( X \) are maximal prefix codes.

Conversely, suppose now \( X \) is a prefix code and \( Y \) is a bifix code. Then, by Lemma 3(i), \( XY \) is a prefix code. Therefore, by Theorem 2, \( XY \) is a prefix strong alt-induced code. Next, suppose moreover that \( X \) is a maximal prefix code and \( Y \) is both a maximal prefix code and a bifix code. Then, on one hand, by the above, \( XY \) is a strong alt-induced code and, on the other hand, again by Lemma 3(i), \( XY \) is a maximal prefix code. Hence, \( XY \) is a maximal prefix strong alt-induced code.

(ii) The proof is similar as in the item (i).

(iii) It follows immediately from Lemma 5 and the items (i) and (ii) of the theorem. \( \square \)

**Example 7** Consider the sets \( X = \{a^n b \mid n \geq 1\}, Y = \{ba^m b \mid m \geq 1\} \) over \( A = \{a, b\} \). Evidently, \( X \) is a prefix code and \( Y \) is a bifix code. Hence, by Theorem 3(i), \( XY = \{a^n b b a^m b \mid n, m \geq 1\} \) is a prefix strong alt-induced code.

In the framework of regular languages, the strong alt-induced codes have the following interesting property.

**Theorem 4** If \( Z \subseteq A^+ \) is a regular strong alt-induced code, then there exists only a finite number of strong alternative codes inducing \( Z \).
Proof

Firstly, recall that, for any language \( Z \) over \( A \), the syntactic congruence of \( Z \), denote by \( \equiv_Z \), is defined as follows

For \( u, v \in A^* : u \equiv_Z v \) if and only if \( \forall x, y \in A^* : xuy \in Z \iff xvy \in Z \).

Next, it is easy to verify that, for any word \( x \in A^* \), \( x^{-1}Z \) is a union of equivalence classes of \( \equiv_Z \). It follows therefore that, for any \( X, Y \subseteq A^* \), \( X^{-1}Z \) and \( ZY^{-1} \) are also unions of equivalence classes of \( \equiv_Z \).

Now, suppose \( Z \) is a regular strong alt-induced code. On one hand, because \( Z \) is regular, \( \equiv_Z \) has finite index, i.e. there exists only a finite number of equivalence classes according to the congruence \( \equiv_Z \). On the other hand, since \( Z \) is a strong alt-induced code, there must exist a strong alternative code \((X, Y)\) such that \( Z = XY \). By Definition 3, are verified the following equalities:

\[
X^{-1}(XY) = Y \quad (1'), \quad (XY)Y^{-1} = X \quad (2').
\]

Thus, by the above, \( X \) and \( Y \) are unions of equivalence classes of the congruence \( \equiv_Z \). Because \( \equiv_Z \) has finite index, it follows that the number of such pairs \((X, Y)\) must be finite. \( \square \)

5 Finite alt-induced codes

This section is reserved to consider alt-induced codes in the framework of finite codes. Some properties, allowing to answer the question, whether a given finite code is an alt-induced code or not, are exhibited. An algorithm to test whether a given finite code is an alt-induced code is proposed.

We denote the cardinality of a set \( X \) by \( |X| \). The following fact is evident.

Fact 1 Let \( X \) and \( Y \) be non-empty finite subsets of \( A^+ \). Then, the product \( XY \) is unambiguous if and only if \( |XY| = |X||Y| \).

From Definition 2 and Fact 1 it follows directly

Corollary 3 If \( Z \) is a finite alt-induced code generated by an alternative code \((X, Y)\), \( Z = XY \), then \( |Z| = |X||Y| \).

Evidently also

Fact 2 If \( X \) and \( Y \) are non-empty subsets of \( A^+ \) and \( Z = XY \), then \( X \subseteq \bigcap_{y \in Y} Z y^{-1} \) and \( Y \subseteq \bigcap_{x \in X} x^{-1}Z \).

Remark 4 The converse inclusions of those in Fact 2 are not true in general even with assumption that the sets \( X, Y \) and \( Z \) are finite codes. Consider, for example, the suffix codes \( X = \{ab, ab^3, b^2a\} \), \( X' = \{a, ab^2\} \) and the prefix codes \( Y = \{a^2, ab, ba, bab\} \), \( Y' = \{a, ba\} \) over \( A = \{a, b\} \). Then, we have

\[
Z = XY = \{aba^2, abab, ab^2a^2, ab^2ab, ab^3a^2, ab^3ab, \}
\]
\[
\quad b^2a^3, b^2a^2b, b^2aba^2, b^2abab\}.
\]

\[
Z' = X'Y' = \{a^2, aba, ab^2a, ab^3a\}.
\]
It is easy to verify that $Z$ is a prefix code, $Z'$ is a bifix code, and $\bigcap_{y \in Y} Zy^{-1} = \{ab, ab^2, ab^3, b^2a\} \not\subseteq X$, $\bigcap_{y \in Y} Z'y^{-1} = \{a, ab, abb\} \not\subseteq X'$.

Let $A = \{a_1, a_2, \ldots, a_k\}, k \geq 2$, and $Z \subseteq A^+$. For every $i$, we denote by $Z_{a_i}$ the set of all the words of $Z$ beginning with the letter $a_i$, namely

$$Z_{a_i} = \{w \in Z \mid w = a_iu, u \in A^+\}$$

where $i = 1, 2, \ldots, k$. Evidently, the non-empty $Z_{a_i}$ constitute a partition of $Z$. Namely

$$Z = \bigcup_{i=1}^{k} Z_{a_i}, \quad Z_{a_i} \cap Z_{a_j} = \emptyset \ (i \neq j) \quad (3).$$

The following result, whose proof is based on Fact 2, will be useful in the sequel.

**Proposition 7** Let $A = \{a_1, a_2, \ldots, a_k\}, k \geq 2$, and $Z \subseteq A^+, Z = \bigcup_{i=1}^{k} Z_{a_i}, Z_{a_i} \neq \emptyset, i = 1, 2, \ldots, k$. If $Z = XY$ with $\emptyset \neq X, Y \subseteq A^+$, then $\forall w \in Z_{a_i}, \exists u \in \text{Pref}(w)$ such that $Y \subseteq u^{-1}Z_{a_i}, i = 1, 2, \ldots, k$. Moreover, $Y = u^{-1}Z_{a_i}$ if $X$ is a prefix code.

**Proof** Suppose $Z \subseteq A^+, Z = \bigcup_{i=1}^{k} Z_{a_i}, Z_{a_i} \neq \emptyset, i = 1, 2, \ldots, k$, and $Z = XY$ with $\emptyset \neq X, Y \subseteq A^+$. Then, from $XY = \bigcup_{i=1}^{k} X_{a_i}$, it follows that $Z_{a_i} = X_{a_i}Y$ for all $i = 1, 2, \ldots, k$, where $X = \bigcup_{i=1}^{k} X_{a_i}$. On the other hand, since $Z = XY$, by Fact 2 $Y \subseteq \bigcap_{x \in X} x^{-1}Z$. Therefore $Y \subseteq x^{-1}Z, \forall x \in X$, and hence $Y \subseteq x^{-1}Z, \forall x \in X_{a_i}, i = 1, 2, \ldots, k$. Because $x^{-1}Z = x^{-1}Z_{a_i}, \forall x \in X_{a_i}$, it follows that $Y \subseteq x^{-1}Z_{a_i}, \forall x \in X_{a_i}, i = 1, 2, \ldots, k$. Thus $\forall w \in Z_{a_i}, \exists u \in \text{Pref}(w)$ such that $w \in X_{a_i}$, we have $Y \subseteq u^{-1}Z_{a_i}, i = 1, 2, \ldots, k$.

If moreover $X$ is a prefix code, then $x^{-1}Z_{a_i} = x^{-1}(X_{a_i}Y) = (x^{-1}X_{a_i})Y = \{e\}Y = Y, \forall x \in X_{a_i}, i = 1, 2, \ldots, k$. Hence, $Y = u^{-1}Z_{a_i}$ where $u \in \text{Pref}(w), w \in Z_{a_i}, i = 1, 2, \ldots, k$. \(\Box\)

From now on, we suppose always that the alphabet $A$ consists of at least two letters, $|A| \geq 2$. Then, a finite code $Z$ over $A$ is called to be of **standard form** if all the words of $Z$ have the length greater than or equal to 2, and it is not the case that all the words of $Z$ begin (end) with the same letter in $A$.

The following two results will be useful in checking whether a finite code of standard form is an alt-induced code or not. As usual, $\text{gcd}(n_1, n_2, \ldots, n_k)$ denotes the greatest common divisor of $n_1, n_2, \ldots, n_k$.

**Theorem 5** Let $A = \{a_1, a_2, \ldots, a_k\}, k \geq 2$, and $Z \subseteq A^+$ is a finite code of standard form, $Z = \bigcup_{i=1}^{k} Z_{a_i}$. If $\text{gcd}(|Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|) = 1$, then $Z$ is not an alt-induced code.

**Proof** Suppose the contrary that $Z$ is an alt-induced code. Then, there is an alternative code $(X, Y)$ such that $Z = XY$. The equality $XY = \bigcup_{i=1}^{k} Z_{a_i}$ implies $Z_{a_i} = X_{a_i}Y$ for all $i = 1, 2, \ldots, k$, where $X = \bigcup_{i=1}^{k} X_{a_i}$.

On the other
hand, by (3), we have $|Z| = \sum_{i=1}^{k} |Z_{a_i}|$. Since $(X, Y)$ is an alternative code and $Z$ is a code of standard form, by Corollary 3, $|Z| = |X||Y|$ with $|X|, |Y| \geq 2$. All this implies that $|Z_{a_i}| = |X_{a_i}||Y|$ for all $i = 1, 2, \ldots, k$, which contradicts the assumption $\gcd(|Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|) = 1$. Thus, $Z$ is not an alt-induced code.

\begin{proof}
Assume $A = \{a_1, a_2, \ldots, a_k\}, k \geq 2$, and $Z \subseteq A^+$ is a finite code of standard form. Then, by (3), we have $|Z| = \sum_{i=1}^{k} |Z_{a_i}|$. Since $|Z|$ is prime, it follows that $\gcd(|Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|) = 1$. By Theorem 5, $Z$ is not an alt-induced code.
\end{proof}

The following example shows that the class of codes mentioned in Theorem 5 is strictly larger than that in Corollary 4.

\begin{example}
Consider the set $Z = \{abc, acab, bac, beca, bbac, cab, cba, caab\}$ over $A = \{a, b, c\}$. It is easy to see that $Z$ is a prefix code of standard form, $Z = Z_a \cup Z_b \cup Z_c$, where

$$Z_a = \{abc, acab\}, Z_b = \{beca, bbac\}, Z_c = \{cab, cba, caab\}.$$

We have $|Z| = 8$ which is composite, but $\gcd(|Z_a|, |Z_b|, |Z_c|) = \gcd(2, 3, 3) = 1$. Hence, by Theorem 5, $Z$ is not an alt-induced code.
\end{example}

From Corollary 3, Fact 2, Proposition 7 and Theorem 5, we can exhibit the following algorithm for testing whether a given finite code is alt-induced or not. As usual, when $Z$ is a finite set of $A^+$, $\min Z$ denotes the minimal wordlength of $Z$. The set of all subsets of $Z$ is denoted by $2^Z$.

\begin{algorithm} 
(A test for finite alt-induced codes)
\begin{itemize}
  \item [Input:] A finite code $Z$ of standard form and $Z = \bigcup_{i=1}^{k} Z_{a_i}$.
  \item [Output:] $Z$ is an alt-induced code or not.
  \item 1. If $\gcd(|Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|) = 1$ then go to Step 4.
  \item 2. Choose $w \in Z_{a_i}, 1 \leq t \leq k$, such that $|w| = \min Z$:
    \begin{itemize}
    \item Set $P_w = \text{Pref}(w) \setminus \{\varepsilon\}$
    \item and $D = \{d \geq 2 \mid d \text{ is a common divisor of } |Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|\}$.
    \end{itemize}
  \item 3. While $P_w \neq \emptyset$ do {
    \begin{itemize}
    \item Take $u \in P_w$; Set $S = u^{-1}Z_{a_i}$ and $Q = \{U \in 2^S \mid |U| \in D\}$;
    \item While $Q \neq \emptyset$ do {
      \begin{itemize}
      \item Take $Y \in Q$;
      \item Set $P = \bigcap_{y \in Y} Z_{Y^{-1}}$ and $R = \{V \in 2^P \mid |V| = |Z|/|Y|\}$;
      \item While $R \neq \emptyset$ do {
        \begin{itemize}
        \item Take $X \in R$;
        \item If $Z = XY$ then go to Step 5 else $R := R \setminus X$;
        \end{itemize}
      \end{itemize}
    \end{itemize}
    \item $Q := Q \setminus Y$;
    \end{itemize}
  } 
\end{itemize}
\end{algorithm}
Proof Suppose \(u\) code if and only if there exist \(u\) \(|X| \geq |Y|\). On the other hand, by (3), we have \(|X|\) \(= |Z|\) \(= \sum_{i=1}^{k} |Z_{a_i}|\). Since \((X,Y)\) is an alternative code and \(Z\) is an alt-induced code, \(Z\) is a code of standard form, by Corollary 5 \(|Z| = |X|, |Y| \geq 2\). All this implies that |Y| is a common divisor of \(|Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|\). Thus, \(Y \in Q\). Next, since \(Z = XY\) with \(\emptyset \neq X, Y \subseteq A^+\), by Fact 1 \(X \subseteq \cap_{y \in Y} Z_y^{-1}\). Hence, we have \(X \in R\) because \(|Z| = |X|, |Y|\).

Conversely, suppose there exist \(u \in P_w\), \(Y \in Q\) and \(X \in R\) such that \(Z = XY\). Then, \(|Z| = |XY| = |X|, |Y|\) with \(\emptyset \neq X, Y \subseteq A^+\). Therefore, by Fact 1 the product \(XY\) is unambiguous. Thus, by Lemma \(\emptyset\) \((X,Y)\) is an alternative code. Hence, \(Z\) is an alt-induced code. \(\square\)

Let us take some examples.

Example 9 Consider the set \(Z = \{a^3, a^2b, ba^2, b^3\}\) over \(A = \{a, b\}\). Clearly, \(Z\) is a code of standard form. \(Z = Z_a \cup Z_b\) with \(Z_a = \{a^3, a^2b\}\) and \(Z_b = \{ba^2, b^3\}\). By Algorithm FIC, we have:

1. Since \(|Z| = 4\) and \(\gcd(|Z_a|, |Z_b|) = 2\), we go to Step 2.
2. Choose \(w = a^3 \in Z_a\) with \(|w| = 3 = \min Z\). Set \(P_w = \{a, aa\}\) and \(D = \{d \geq 2 \mid d\text{ is a common divisor of }|Z_a|, |Z_b|\} = \{2\}\).
3. While \(P_w = \emptyset\) do {
   3.1. Take \(a \in P_w\) we have \(S = a^{-1}Z_a = \{aa, ab\}\) and \(Q = \{aa, ab\}\);
   While \(Q = \emptyset\) do {
      Take \(Y = \{aa, ab\} \in Q\);
      Set \(P = Z(aa)^{-1} \cap Z(ab)^{-1} = \{a, b\} \cap \{a\} = \{a\}\) and \(R = \{\{a\}\}\);
      While \(R = \emptyset\) do {
         Take \(X = \{a\} \in R\);
         Since \(Z \neq XY\), which implies \(R := R \setminus X = \emptyset\);
      }
      \(Q := Q \setminus Y = \emptyset\);
   }
   \(P_w := P_w \setminus \{a\} = \{aa\} \neq \emptyset\);
3.2. Take \( aa \in P_w \), we have \( S = (aa)^{-1}Z_a = \{ a, b \} \) and \( Q = \{ \{ a, b \} \} \);

While \( Q \neq \emptyset \) do 

\( Y = \{ a, b \} \in Q; \)

Set \( P = Z_a^{-1} \cap Z_b^{-1} = \{ a^2, ba \} \cap \{ a^2, b^5 \} = \{ a^2 \} \) and \( R = \{ \{ a^2 \} \} ; \)

While \( R \neq \emptyset \) do 

\( X = \{ a^2 \} \in R; \)

Since \( Z \neq XY \), which implies \( R := R \setminus X = \emptyset; \)

\( Q := Q \setminus Y = \emptyset; \)

\( P_w := P_w \setminus \{ aa \} = \emptyset; \)

We go to Step 4.

4. \( Z \) is not an alt-induced code, and the algorithm ends.

Example 10 Again consider the set \( Z \) in Remark 4. Evidently, \( Z \) is a code of standard form, \( Z = Z_a \cup Z_b \), where

\[
Z_a = \{ aba^2, abab, ab^2a^2, ab^2ab, ab^3a^2, ab^3ab, ab^4a^2, ab^4ab \}
\]

\[
Z_b = \{ b^2a^3, b^2a^2b, b^2aba^2, b^2abab \}.
\]

By Algorithm FIC, we have:

1. Since \( |Z| = 12 \) and \( \gcd(|Z_a|, |Z_b|) = 4 \), we go to Step 2.
2. Choose \( w = aba^2 \in Z_a \) with \( |w| = 4 = \min Z \). Set \( P_w = \{ a, ab, aba \} \) and \( D = \{ d \geq 2 \mid d \text{ is a common divisor of } |Z_a|, |Z_b| \} = \{ 2, 4 \} \).
3. While \( P_w \neq \emptyset \) do 

\( \text{Take } ab \in P_w, \text{ we have } S = \{ a^2, ab, ba^2, bab, b^2a^2, b^2ab, b^3a^2, b^3ab \} \)

and \( Q = \{ U \in 2^S \mid |U| \in \{ 2, 4 \} \} \); 

While \( Q \neq \emptyset \) do 

\( Y = \{ a^2, ab, ba^2, bab \} \in Q; \)

Set \( P = \bigcap_{Y \in Q} Z_{Y^{-1}} = \{ ab, ab^2, ab^3, b^2a \} \)

and \( R = \{ V \in 2^P \mid |V| = 3 \} \); 

While \( R \neq \emptyset \) do 

\( X = \{ ab, ab^3, b^2a \} \in R; \)

Because \( Z = XY \), we go to Step 5;

5. \( Z = XY = \{ ab, ab^3, b^2a \} \{ a^2, ab, ba^2, bab \} \) is an alt-induced code, and the algorithm ends.

Theorem 6 Given a finite code \( Z \) of standard form over \( A = \{ a_1, a_2, \ldots, a_k \} \), \( k \geq 2 \), we can determine whether \( Z \) is an alt-induced code or not in \( O(km^2n^2) \) worst-case time, where \( m = \min Z \) and \( n = \min \{|Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|\} \).

Proof The best-case for the algorithm is when \( \gcd(|Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|) = 1 \), which takes \( O(hk) \) steps to perform the task, where \( h \) is the number of digits in the number \( n \) (see [10], pages 21–22).

In Step 2, we can choose \( w \in Z_{a_1} \) such that \( |w| = m = \min Z \), \( |Z_{a_1}| \) = \( \min \{|Z_{a_1}|, |Z_{a_2}|, \ldots, |Z_{a_k}|\} \), and the set \( P_w \) has at most \( m - 1 \) words. Then,
in Step 3, for each word \( u \) in \( P_w \), it takes \( O(2^{2n}) \) worst-case time to perform for \( Q \) and \( R \), and \( O(|Y||Z|) = O(mkn) \) worst-case time in finding \( P \). Thus, the total running time for determining, whether \( Z \) is an alt-induced code or not, is \( O(m).O(2^{2n}).O(mkn) = O(km^2n^4) \) in the worst-case.

Acknowledgements

The authors 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 authors express our sincere thanks to Dr. Nguyen Huong Lam and Dr. Kieu Van Hung for their useful discussions.

References

1. Berstel, J., Perrin, D.: Theory of codes. Academic Press, New York (1985).
2. Biegler, F., Daley, M., McQuillan, I.: Algorithmic decomposition of shuffle on words. Theoretical of Computer Science 454, 38–50 (2012).
3. Han, N.D., Huy, P.T.: On unambiguity of languages related to codes. In: James, J.P., Victor, C.M.L., Cho, L.W., Taeshik, S. (eds.) Future Information Technology, Application, and Service, pp. 32–38. Springer Netherlands (2012).
4. Han, Y.S., Salomaa, A., Salomaa, K., Wood, D., Yu, S.: On the existence of prime decompositions. Theoretical of Computer Science 376, 60–69 (2007).
5. Hien, N.T.: Characterizations for several classes of alternative codes. Journal of Computer Science and Cybernetics 32, 273–283 (2016).
6. Hung, K.V., Van, D.L.: Prime decomposition problem for several kinds of regular codes. Lecture Notes in Computer Science 4281, 213–227 (2006).
7. Huy, P.T., Nam, V.T.: Alternative codes and pre-context codes. The 7th National conference: Selected problems about IT and Telecommunication, 188–197, (2004) (in Vietnamese).
8. Jastrzab, T., Czech, Z.J.: A parallel algorithm for the decomposition of finite languages. Studia Informatica 34, 5–16 (2014).
9. Jürgensen, H., Konstantinidis, S.: Codes. In: Rozenberg, G., Salomaa, A. (eds.), Handbook of Formal Languages, Springer, Berlin, pp. 511–607 (1997).
10. Mollin, R.A.: Fundamental number theory with applications (2nd ed). Boca Raton: Chapman & Hall/CRC (2008).
11. Sardinas, A.A., Patterson, C.W.: A necessary and sufficient condition for the unique decomposition of coded messages. IRE International Convention Record 8, 104–108 (1953).
12. Schützenberger, M.P.: On a question concerning certain free submonoids. Journal of Combinatorial Theory 1, 437–442 (1966).
13. Shyr, H.J.: Free monoids and languages. Hon Min Book Company, Taichung (1991).
14. Vinh, H.N., Nam, V.T., Huy, P.T.: Codes based on unambiguous products. Lecture Notes in Computer Science 6423, 252–262 (2010).
15. Wieczorek, W.: An algorithm for the decomposition of finite languages. Logic Journal IGPL 18, 355–366 (2010).