数理解析研究所講究録
第 1712 巻 2010 年 1-9

Some Combinatorial Properties of Extractable Codes *

國持良行 (Yoshiyuki Kunimochi)
静岡理工科大学・総合情報学部

Faculty of Comprehensive Informatics,
Shizuoka Institute of Science and Technology

Abstract

This paper deals with extractablity of codes. A submonoid N of the free monoid A^* over a finite alphabet is called extractable if $z, xzy \in N$ implies $xy \in N$. Since an extractable submonoid is biunitary, its base forms a bifix code. First, we consider the necessary and sufficient conditions whether a given infix code C is extractable or not. And we introduce the bidecomposition graph of a code to easily check the extractability of languages. Secondly, we investigage the extractability for the families of other related bifix codes. That is, intercodes, comma-free codes, Dyck codes, strong codes and solid codes are extractable codes. We newly define the bifix codes, called $e(m)$-codes and $\overline{e}(m)$-codes, and refer to the extractability of them.

1 Preliminaries

Let A be a finite nonempty set of *letters*, called an alphabet and let A^* be the free monoid generated by A under the operation of catenation with the identity called the empty word, denoted by 1. We call an element of A^* a word over A. The free semigroup $A^* \setminus \{1\}$ generated by A is denoted by A^+. The catenation of two words x and y is denoted by xy. The length $|w|$ of a word $w = a_1 a_2 \dots a_n$ with $a_i \in A$ is the number n of occurrences of letters in w. Clearly, $|1| = 0$.

A word $u \in A^*$ is a prefix(or suffix) of a word $w \in A^*$ if there is a word $x \in A^*$ such that $w = ux$(or $w = xu$). A word $u \in A^*$ is a factor of a word $w \in A^*$ if there exist words $x, y \in A^*$ such that $w = xuy$. Then a prefix (a suffix or a factor) u of w is called proper if $w \neq u$.

A subset of A^* is called a language over A. A language $L \subset A^*$ is called reflective if $uv \in L$ implies $vu \in L$ for any $u, v \in A^*$. A nonempty language which is the set of free generators of a submonoid of A^* is called a code over A. A nonempty language C is called a prefix (or suffix) code if $u, uv \in C$ ($u, vu \in C$) implies $v = 1$. C is called a bifix code if C is both a prefix code and a suffix code. A nonempty language C is called an infix code if $u, xuy \in C$ implies $x = y = 1$. The language $A^n = \{w \in A^* \mid |w| = n\}$ with $n \geq 1$ is called a *full uniform code* over A. A nonempty subset of A^n is called a uniform code over A.

Let M be a monoid and N be its submonoid. N is right unitary (in M) if $u, uv \in N$ implies $v \in N$. Left unitary is defined in a symmetric way. The submonoid N of M is biunitary if it is both left and right unitary. N is called *extractable* if $z, xzy \in N$ implies $xy \in N$ for any $x, y, z \in M$. If N is extractable, then N is biunitary. Indeed, $uv = 1uv, u \in N$ implies $v = 1v \in N$ and $uv = uv1, v \in N$ implies $u = 1u \in N$.

It is known that a submonoid N of A^* is right unitary (resp. left unitary, biunitary) if and only if the minimal set $N_0 = (N \setminus 1) \setminus (N \setminus 1)^2$ of generators of N, namely the base of N, is a prefix code (resp. a suffix code, a bifix code) ([1, p.46],[3, p.108]). If a submonoid N of A^* is extractable, then the base of N is a bifix code.

*This is an abstract and the paper will appear elsewhere.

2 Extractablity of Infix Codes

Our aim in this section is to determine whether for a given infix code C it is an extractable code or not in terms of its syntactic monoid. We introduce the bidecomposition graph of a language to easily check the extractability of the language.

2.1 Checking Extractability by a Syntactic Monoid

We begin with a useful and fundamental lemma concerned with the extractability of infix codes.

Lemma 2.1 Let $C \subset A^*$ be an infix code. C^* is extractable if and only if $z \in C$ and $xzy \in C^2$ imply $xy \in C$ for any $x, y, z \in A^+$.

(Proof) (only if part) Since C^* is extractable and $xy \neq 1$, we have $xy \in C^+$. $z \in C$ and $xzy \in C^2$ yield that $xu \in C$ and $vy \in C$ with $z = uv$ for some u, $v \in A^+$ because C is an infix code. $xy \in C^+ \setminus C$ means that some factor of either x or y is an element of C. This is a contradiction. Therefore $xy \in C$ must hold.

(if part) Since C is an infix code and thus C^* is biunitary, we may show that $z, xzy \in C^*$ implies $xy \in C^+$ for any x, $y \in A^+$. Moreover as C is an infix code, it suffices to show by induction on k the following implication:

$$z = z_1 z_2 \dots z_k,\ xzy = w_1 w_2 \dots w_{k+1} \text{ implies } xy \in C \tag{1}$$

where $z_i \in C\,(1 \leq i \leq k)$, $w_i \in C\,(1 \leq i \leq k+1)$, $x \neq 1$ is a proper prefix of w_1 and $y \neq 1$ is a proper suffix of w_{k+1}.

(Base step) In case of $k = 1$ it is trivial by the hypothesis of this proposition.

(Induction step) We suppose that the implication (1) holds in case of $k = n$. Now let $k = n+1$. Since $w_{n+1}w_{n+2} = x'z_{n+1}y \in C^2$, $x' \neq 1$ and $z_{n+1} \in C$, $x'y \in C$ by the hypothesis of this proposition. By setting $w'_{k+1} = x'y \in C$, the condition $xz_1z_2 \dots z_n y = w_1 w_2 \dots w_n w'_{n+1}$ holds. By induction hypothesis, we have $xy \in C$. Thus we have shown the implication (1). □

Since a uniform code is an infix code, the next corollary holds immediately.

Corollary 2.1 Let U be a uniform code. U^* is extractable if and only if $z \in U$ and $xzy \in U^2$ imply $xy \in U$ for any $x, y, z \in A^+$. □

We introduce terms of the syntactic monoid of a language. Let M be a monoid, $L \subset M$ and $w \in M$. Let $P_L \subset M \times M$ be the relation on M consisiting of all the pairs $(u, v) \in M \times M$ such that, for all $x, y \in M$, $xuy \in L$ if and only if $xvy \in L$. The relation P_L is a congruence, called the principal congruence of L. Instead of $(u, v) \in P_L$, we write $u \equiv v\ (P_L)$. The set L is said to be disjunctive in M if P_L is the equality relation. The set $W_L = \{u \in M \mid MuM \cap L = \emptyset\}$ is called the residue of L. If $W_L \neq \emptyset$ then W_L is an ideal of M. If L is a singleton set, $L = \{c\}$, we often write c instead of $\{c\}$; thus c being disjunctive means $\{c\}$ is disjunctive, $P_c = P_{\{c\}}$ is the equality relation.

Now assume M is a monoid with identity e and zero 0 and $|M| \geq 2$, hence $e \neq 0$. The intersection of all nonzero ideals of M, if it differs from $\{0\}$, is called the core of M, denoted by core(M). An element $c \in M$ is called an annihilator if $cx = xc = 0$ for all $x \in M \setminus \{e\}$. Annihil(M) denotes the set of all annihilators of M.

When $M = A^*$ for an alphabet A then P_L is also referred to as the syntactic congruence of L and the factor monoid Syn(L) $= A^*/P_L$ as the syntactic monoid of L. The morphism σ_L of X^* onto Syn(L) is called the syntactic morphism of L.

Let C be an infix code. Concerning the three special element introduced below, Theorem 2.1 holds [6].

The set $\{1\}$ is a P_C-class; therefore, the identity element e of $\mathrm{Syn}(C)$ is $\{1\}$. Since $W_C \neq \emptyset$ is a P_C-class, $\mathrm{Syn}(C)$ has a zero element $0 = W_C/P_C$. For any $u \in C$, $xuy \in C$ implies $x = y = 1$. Therefore C is also a P_C-class denoted by c, that is, $c = C/P_C \in \mathrm{Syn}(C)$.

Theorem 2.1[6] The following conditions on a monoid M with identity e are equivalent:

(i) M is isomorphic to the syntactic monoid of an infix code L.

(ii) (α) $M \setminus \{e\}$ is subsemigruop of M;
(β) M has a zero;
(γ) M has a disjunctive element c such that $c \notin \{e, 0\}$ and $c = xcy$ implies $x = y = e$.

(iii) (α) ; □
(δ) M has a disjunctive zero;
(ϵ) $\mathrm{core}(M) = \{c, 0\}$ with $c \in \mathrm{Annihil}(M)$.

(iv) $(\alpha), (\delta)$;
(ζ) there exists $0 \neq c \in \mathrm{core}(M) \cap \mathrm{Annihil}(M)$.

Proposition 2.1 Let C be an infix code and $M = \mathrm{Syn}(L)$ be its syntactic monoid Let c be a P_C-class of C, that is $0 \neq c \in \mathrm{core}(M) \cap \mathrm{Annihil}(M)$. Then,

(1) C is an extractable code if and only if

$$c = f_0 f_1 = f_1 f_2 = f_2 f_3 \;\Rightarrow\; c = f_0 f_3 \quad \text{for any } f_0,\, f_1,\, f_2,\, f_3 \in M.$$

(2) C is a reflective and extractable code if and only if

$$c = f_0 f_1 = f_1 f_2 \;\Rightarrow\; f_0 = f_2 \quad \text{for any } f_0,\, f_1,\, f_2 \in M.$$

(Proof) (1) (only if part) There exist $x, u, v, y \in A^+$ such that $xu, uv, vy \in C$ and $\sigma_C(x) = f_0, \sigma_C(u) = f_1, \sigma_C(v) = f_2, \sigma_C(y) = f_3$. By Lemma 2.1, we have $xy \in C$. Therefore, $f_0 f_3 = \sigma_C(xy) = c$. (if part) Assume that $xu, uv, vy \in C$ for some $x, u, v, y \in A^+$. Since $\sigma_C(x)\sigma_C(u) = \sigma_C(u)\sigma_C(v) = \sigma_C(v)\sigma_C(y) = c$, we have $\sigma_C(xy) = \sigma_C(x)\sigma_C(y) = c$, that is $xy \in C$.

(2) (only if part) We suppose that $c = f_0 f_1 = f_1 f_2$ but $f_0 \neq f_2$. Then there exist $x, y \in A^+$ such that $\sigma_C(u) = f_0, \sigma_C(v) = f_2$ and $xuy \notin C$ and $xvy \in C$, or $xuy \in C$ and $xvy \notin C$. By symmetry, we may only consider the case $xuy \notin C$ and $xvy \in C$. Since C is reflective, $uyx \notin C$ and $vyx \in C$. Setting $f_3 = \sigma_C(yx)$, we have $f_0 f_3 \neq c$ and $f_2 f_3 = c$. Therefore we have $c = f_0 f_1 = f_1 f_2 = f_2 f_3$ and $c \neq f_0 f_3$. This contraticts to (1).

(if part) First we show that $c = ab$ implies $c = ba$ for any $a, b \in M$. There exist $u, v \in A^*$ such that $\sigma_C(u) = a, \sigma_C(v) = b$. Then $c = ab$ means $uv \in C$ and $c = ba$ means $vu \in C$. Since C is reflective, we have $c = ba$.

Assume that $c = f_0 f_1 = f_1 f_2 = f_2 f_3$. The argument above yields $c = f_2 f_1 = f_2 f_3$. By hypothesis, we have $f_1 = f_3$ and thus $c = f_0 f_1 = f_0 f_3$. □

2.2 Bidecomposition Graph of a Language

We introduce a graph in order to determine whether a given infix code is an extractable code or not. The bidecomposition graph (2D graph, for abbreviation) $G_L = (V, E)$ of a language L is defined as follows:

(1) $V = \mathrm{Syn}(L)$; the syntactic monoid of L.

(2) $E = \{(a,b) \in V \times V \,|\, ab \in \sigma_L(L)\}$, where σ_L is the syntactic morphism of L.

Especially if L is an infix code, then $ab \in \sigma_L(L)$ is equivalent to $ab = c = \sigma_L(L)$.
$(v_0, v_1, \ldots, v_n)$ is called a path of length n in a graph $G = (V, E)$ if $(v_{i-1}, v_i) \in E$ for all $i\,(1 \le i \le n)$.
Proposition 2.1 can be stated in terms of graph.

Corollary 2.1 Let C be an infix code and $G_C = (V, E)$ be the 2D graph of C. Let c be a P_C-class of C. Then,
(1) C is an extractable code if and only if $(v_0, v_3) \in E$ for every path (v_0, v_1, v_2, v_3) in G_C of length 3.
(2) C is a reflective extractable code if and only if $(v_0, v_1), (v_1, v_2) \in E$ implies $v_0 = v_2$. □

Example 2.1 Let $C = ab^+a$ be a regular infix code over A and $M = \mathrm{Syn}(C) = \{0, e, [a], [ab], c, [b], [ba]\}$, where $0 = W_C = \{u \in A^+ | C \cap A^*uA^* = \emptyset\}$, $e = [1]$ and $c = [aba] = \sigma_L(aba)$. Its multipliplication table is shown in Table 1. We construct the 2D graph $G_C = (V, E)$ shown in Figure 1 and observe the extractability. All the paths of length 3 in G_C are only (e, c, e, c) and (c, e, c, e), while both (e, c) and (c, e) are in E. Therefore ab^+a is an extractable code. □

Table 1: The multipliplication table of M, except for $0, e$.

	$[a]$	$[ab]$	c	$[b]$	$[ba]$
$[a]$	0	0	0	$[ab]$	c
$[ab]$	c	0	0	$[ab]$	c
c	0	0	0	0	0
$[b]$	$[ba]$	0	0	$[b]$	$[ba]$
$[ba]$	0	0	0	0	0

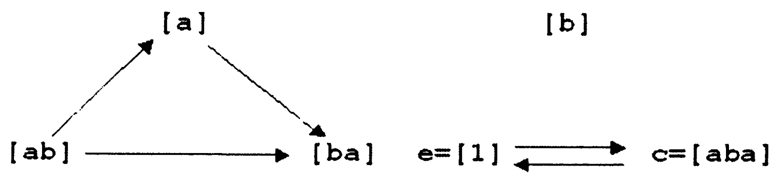

Figure 1: The 2D graph G_C of the code C.

Before we give some examples for reflective infix code which is regular, we pay attention to some remarks and properties [4]. Note that even a reflective regular prefix (or suffix) code is finite. Indeed, suppose that C is infinite. There exists some word $w \in C$ such that $w = uxv$, $x \neq 1$ and $ux^nv \in C$ for any positive integer n by the pumping lemma of regular languages. Since C is reflective, we have $vux \in C$ and $vux^n \in C$. This is contradicts to that C is an prefix code. Thus C is finite. Similarly for a suffix code.

A word $x \in A^+$ is primitive if $x = f^n$ for some $f \in A^+$ implies $n = 1$, where $f^n = \overbrace{ff\cdots f}^{n}$. Two words x, y are called *conjugate*, denoted by $x \equiv y$ if there exist words u, v such that $x = uv, y = vu$. We frequently say that y is a conjugate of x. $\equiv$ is an equivalence relation, we denote by $cl(w)$ the conjugate class of w with respect to the relation $\equiv$.

For a nonempty word $w \in A^+$, its conjugate class $cl(w) = \{vu \mid uv = w\}$ is a reflective uniform, thus regular infix, code. The following result is given in [4] concerned with the extractabilty of conjugate classes.

Proposition 2.2[4] Let $w = (uv)^n u$ with $u \in A^*$, $v \in A^+$ and $n \geq 2$. Then, $cl(w)$ is an extractable code $\iff$ $u = 1$. □

Example 2.2 (1) Let $w_1 = ababa$ and $cl(w_1)$ be its conjugate class. Then we can construct the 2D graph $G_{cl(w_1)} = (\mathrm{Syn}(cl(w_1)), E)$ of $cl(w_1)$ shown in Figure 2. $([ab],[aba]),([aba],[ba]) \in E$ but $[ab] \neq [ba]$ implies that $cl(w_1)$ is not an extractable code.

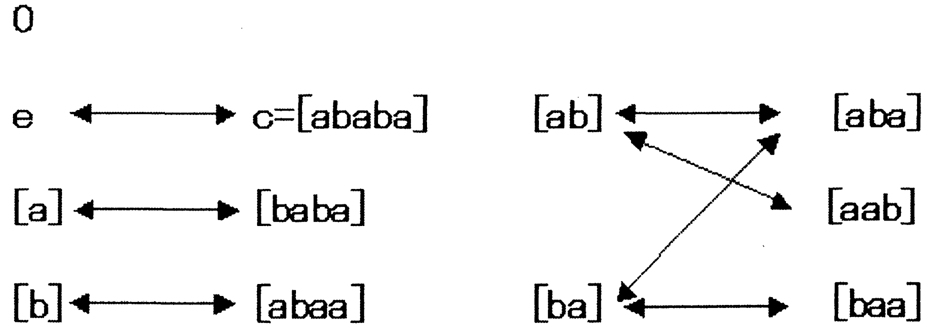

Figure 2: The 2D graph $G_{cl(w_1)}$ of $cl(w_1)$.

(2) Let $w_2 = abab$ and $cl(w_2)$ be its conjugate class. The 2D graph $G_{cl(w_2)} = (\mathrm{Syn}(cl(w_2)), E)$ of $cl(w_2)$ is shown in Figure 3. Since every edge $(x, y) \in E$ has the reverse edge $(y, x) \in E$, $cl(w_2)$ is an extractable code. □

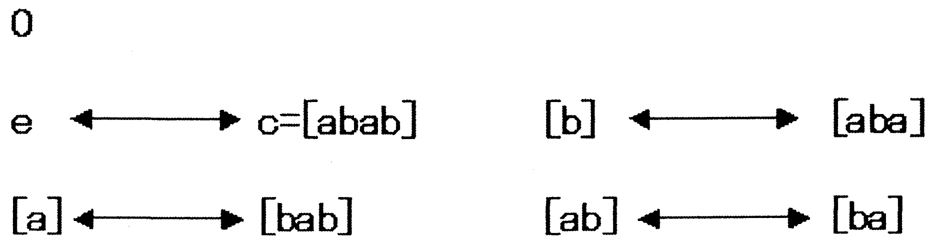

Figure 3: The 2D graph $G_{cl(w_2)}$ of $cl(w_2)$.

3 Extractability of Other Related Codes

We investigage the extractability for the families of other related bifix codes. First we newly introduce some kinds of bifix codes, that is, $e(m)$-codes and $\bar{e}(m)$-codes and consider the extractability of these codes. Secondly, we investigate the extractability of intercodes, comma-free codes, Dyck codes, etc..

3.1 Some special classes of Bifix Codes

Special kinds of submonoids are introduced as follows.

Definition 3.1 Let m be a positive integer. The conditions $e(m)$ and $\overline{e}(m)$ with respect to a submonoid C^* of A^* are defined as follows:

$$\begin{aligned} e(m): &\quad u_0u_1, u_1u_2, \ldots, u_{m-1}u_m \in C^* \Rightarrow u_0u_m \in C^*, \\ \overline{e}(m): &\quad u_0u_1, u_1u_2, \ldots, u_{m-1}u_m \in C^* \Rightarrow u_mu_0 \in C^*. \end{aligned}$$

□

Proposition 3.1 Let m be a positive integer with $m \geq 2$. If C^* satisfies the condition eihter $e(m)$ or $\overline{e}(m)$, then C^* is biunitary.

(Proof) We only show the case of $e(m)$. Let $u, uv \in C$. Since $\overbrace{1 \cdot 1, \ldots, 1 \cdot 1}^{m-2\,\text{times}}, 1u, uv \in C^*$ and C is an $e(m)$-bifix code, we have $v = 1 \cdot v \in C^*$. Similary $u, vu \in C^*$ implies $v \in C^*$. Therefore, C^* is biunitary. □

By this proposition, the base of a submonoid satisfying the condition $e(m)$ (or $\overline{e}(m)$) is a bifix code and we call it an $e(m)$-bifix code (or an $\overline{e}(m)$-bifix code).

Proposition 3.2 Let $m, n \geq 2$ be integers. If $m-1$ is a divisor of $n-1$, then an $e(m)$-bifix code is an $e(n)$-bifix code.

(Proof) Assume that C is an $e(m)$-bifix code and $m-1$ is a divisor of $n-1$. First of all, we show that C is an $e(m+k(m-1))$-bifix code by induction on $k = 0, 1, 2, \ldots$.

It is trivial in case of $k = 0$. Suppose that C is an $e(m+(k-1)(m-1))$-bifix code for $k \geq 1$. Then,

$$u_0u_1, \ldots, u_{m+k(m-1)-1}u_{m+k(m-1)} \in C^*.$$

By induction hypothesis, we have

$$u_0u_{m+(k-1)(m-1)}, \ldots, u_{m+(k-1)(m-1)+m-2}u_{m+(k-1)(m-1)+m-1} \in C^*.$$

Since C is an $e(m)$-bifix code, we have $u_0u_{m+k(m-1)} \in C^*$. Therefore C is an $e(m+k(m-1))$-bifix code for any nonnegative integer k.

There exists a positive integer c such that $n-1 = c(m-1)$. As $n = m+(c-1)(m-1)$, the argument above leads the conclusion. □

Corollary 3.1 An $e(2)$-bifix code is an $e(m)$-bifix code for any integer $m \geq 2$. □

Example 3.1 Let $m \geq 2$, $A = \{a_0, a_1, \ldots, a_m\}$, $C = \{a_0a_1, a_1a_2, \ldots, a_{m-1}a_m, a_0a_m\}$. Then C is an $e(m)$-bifix code but is not an $e(n)$-bifix code for any n with $2 \leq n < m$.

Indeed, suppose that $u_0u_1, u_1u_2, \ldots, u_{m-1}u_m \in C^*$. If $|u_0|$ is even, then $u_i \in C^*(0 \leq i \leq m)$ because C^* is biunitary. Therefore $u_0u_m \in C^*$.

So we consider the case that $|u_0|$ is odd. Then $|u_i|(0 \leq i \leq m)$ is odd because $|u_iu_{i+1}| = 2n_i(0 \leq i < m)$ is even and nonzero. We have $u_iu_{i+1} = a_{j_i}a_{j_i+1} \cdots a_{j_i+2n_i-1}$ $(0 \leq i < m)$ and $0 \leq j_0 < j_1 < \cdots < j_m \leq m$. Therefore $u_i = a_i$ must hold for any $i\,(0 \leq i \leq m)$. Thus $u_0u_m = a_0a_m \in C^*$.

Careful observation leads that for any n with $2 \le n < m$, $a_0a_1, a_1a_2, \ldots, a_{n-1}a_n \in C^*$ but $a_{n-1}a_n \notin C^*$. Thus C is not an $e(n)$-bifix code. $\square$

The inclusion order among the classes of $e(m)$-bifix codes is shown in Figure 4, which is similar to the division order in the set of all natural numbers.

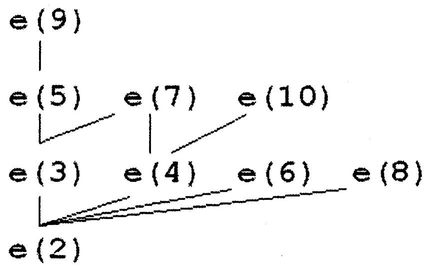

Figure 4: The Hierarchy of the classes of $e(m)$-bifix codes.

Proposition 3.3 An extractable code is an $e(3)$-bifix code. The converse does not hold. However, an $e(3)$-bifix code which is an infix code is an extractable code.

(Proof) It is obvious that an extractable code is an $e(3)$-bifix code. $\{bac, a\}$ is an $e(2)$-bifix code but not an extractable code.

Assume that C is an infix code and an $e(3)$-bifix code. Let $z \in C, xzy = uv$ with $x, y \in A^+, u, v \in C$. Since C is an infix code, we have $u = xz_1, z = z_1z_2, v = z_2y \in C$. $1 \neq xy \in C^*$ because C is an $e(3)$-bifix code. As any factor of x or y cannot be in C, $xy \in C$ must hold. By Lemma 2.1, C is an extractable code. $\square$

Note that $C = \{bac, a\}$ is an $e(2)$-bifix code but is not an extractable code. A full uniform code is an extractable code but is not an $e(2)$-bifix code. Indeed, let $u_0 = a, u_1 = a^2, u_2 = a$, then $u_0u_1 = u_1u_2 = a^3 \in \{a^3\}$ but $u_0u_2 = a^2 \notin \{a^3\}$. Figure 5 shows the inclusion relation between the class of extractable codes and the classes of $e(m)$-bifix codes. $e(2), e(3)$ and Ext. mean classes of $e(2)$-bifix codes, $e(3)$-bifix codes and extractable codes, respectively.

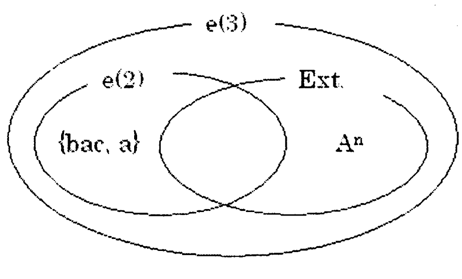

Figure 5: The inclusion relation between the class of extractable codes and the classes of $e(m)$-bifix codes.

3.2 Other Related Codes

In this section, we consider the class of extractable codes and other classes of intercodes, strong codes, solid code and so on. We begin with the class of intercodes.

Let m be a positive integer. A language I is called an *intercode of index* m if $I^{m+1} \cap A^+ I^m A^+ = \emptyset$. The family of intercodes of index m is denoted by $\mathbf{I}_m$. It is known that an intercode is a thin bifix code and $\mathbf{I}_1 \subsetneqq \mathbf{I}_2 \subsetneqq \cdots \subsetneqq \mathbf{Q}$, where $\mathbf{Q}$ denotes family of all sets of primitive words.

Proposition 3.4 An intercode C of index 1 is an extractable code.

(Proof) The case that $z \in C, xzy \in C^2$ for some $x, y \in A^+$ never happens. Therefore $z \in C, xzy \in C^2$ implies $xy \in C$ for any $x, y \in A^+$. Since C is an infix code, we have the conclusion by Lemma 2.1. □

The class of intercodes of index 1 and the class of comma-free codes are identical. The Dyck code is an intercode of index 1. So the following corollary holds.

Corollary 3.3 A comma-free code and the Dyck code are extractable codes. □

The converse of this proposition does not hold. Indeed, a full uniform code A^n is an extractable and infix code but since $A^+(A^n)^m A^+ \cap (A^n)^{m+1} = A^{n(m+1)} \neq \emptyset$, A^n is not an intercode of index m for any $m \geq 1$. Moreover, the following examples show that there exists an intercode I_{m+1} of index $m+1$ such that it is neither an extractable code nor an intercode of index m for any $m \geq 1$.

Example 3.2 (1) Let $m \geq 1$, $A = \{a, b\}$ and $u_i = a^i b^i a^i$ for $i \geq 1$. The language

$$L = \{u_1 u_2 \dots u_{m+1} u_{m+2}, u_2, \dots u_m, u_{m+1}\}$$

satisfies the condition $L^{m+2} \cap A^+ L^{m+1} A^+ = \emptyset$, that is, L is an intercode of index $m+1$. While $u_1 u_2 \dots u_{m+1} u_{m+2} \in L \cap A^+ L^m A^+$ and thus $L^{m+1} \cap A^+ L^m A^+ \neq \emptyset$. Therefore L is neither an intercode of index m nor an extractable code.

(2) Let $m \geq 1$, $A = B \cup \{\$\}$ and $B = \{a_1, a_2, \dots, a_m\}$.

$$L' = B \cup (\bigcup_{i=0}^{m} \$B^i\$).$$

is an intercode of index $m+1$ but not of index m. And L is an extractable code.

The next example shows us that classes of extractable codes and right semaphore codes are independent with respect to inclusion. By left-right duality, we have a similar result for left semaphore codes.

Example 3.3 (1) C is called a *right semaphore code* if $A^*C \subset CA^*$ and C is a prefix code. This condition is equivalent to that C is a p-infix code and a maximal prefix code, where p-infix means that $w, uwv \in L$ implies $v = 1$ for any $u, v, w \in A^*$ [3, p.66]. a^*b is a right semaphore code but not an extractable code. Indeed $ab, b \in a^*b$ but $a \notin a^*b$. Conversely, $[abab]$ is an extractable code but is not a right semaphore code since it is not a maximal prefix code.

(2) C is called a *solid code* if it is an overlap-free infix code. Obviously a solid code is an extractable code and the converse does not hold. C is called a *strong code* if (i) $x, y_1 y_2 \in C$ implies $y_1 x y_2 \in C^+$ and (ii) $x, y_1 x y_2 \in C^+$ implies $y_1 y_2 \in C^+$. Obviously a strong code is an extractable code and the converse does not hold.

References

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

[2] H.J.Shyr, *Free Monoid and Languages*, Hon Min Book Co, Taichung, Taiwan,1991.

[3] Shyr-shen Yu, *Languages and Codes*, Tsang Hai Book Publishing Co, Taiwan, 2005.

[4] G. Tanaka, Y. Kunimochi and M. Katsura, *Remarks on extractable submonoids*, 京都大学数理解析研究所講究録 1655, pp.106-110, 2009.6.

[5] A.Luca and S.Varricchio, *Finiteness and Regularity in Semigroups and Formal Languages*, Monographs in Theoretical Computer Science, Springer, 1999.

[6] M.Petrich and G.Thierrin, *The Syntactic Monoid of an Infix Code*, Proceedings of the American Mathematical Society, Vol.109, No.4, 1990.

[7] H.Jurgensen and S.Konstantinidis, *Codes*, In G.Rozenberg and A.Salomaa (eds.) Handbook of Formal Languages. Word, Language, Grammar, Vol.1, Springer, Heidelberg, 1997.

[8] H. J. Shyr and S. S. Yu, *Intercodes and Some Related Properties*, Soochow J. Math., Vol.16, No.1, pp.95–107 (1990).

[9] S. S. Yu, *Characterization of Intercodes*, Intern. J. Computer Math., Vol.36, pp.39–45 (1990).