Extractable Codes and Conjugacy Classes

静岡理工科大学・総合情報学部 國持 良行
(Yoshiyuki Kunimochi)
Faculty of Comprehensive Informatics,
Shizuoka Institute of Science and Technology

abstract This paper deals with insertability and mainly extractablity of codes, which are extensions of well-known strong codes. A code C is called insertable (or extractable) if the free submonoid C^* generated by C satisfies if $z, xy \in C^*$ implies $xzy \in C^*$(or $z, xzy \in C^*$ implies $xy \in C^*$). We show that a finite insertable code is a full uniform code. On the other hand there are many finite extractable codes which are not full uniform codes. We cannot still characterize the structures of them.

Here we summarize some results on the extractability of infix codes, especially uniform codes. After that we give some new results on the relation between periodicity of a word and extractablity of its conjugacy class.

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 $w = a_1a_2 \dots a_n$ 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 nonempty language C which is the set of free generators of some submonoid M of A^* is called a *code* over A. Then C is called the *base* of M and coincides with the minimal set $(M \setminus \{1\}) \setminus (M \setminus \{1\})^2$ of generators of M. A nonempty language C is called a *prefix* (or *suffix*) code if $u, uv \in C$ (or $u, vu \in C$) 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 $A^n = \{w \in A^* | |w| = n\}$ 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.

A word $x \in A^+$ is *primitive* if $x = r^n$ for some $r \in A^+$ implies $n = 1$, where r^n is the n-th power of r, that is, $r^n = \overbrace{rr \cdots r}^{n}$.

PROPOSITION 1.1 *([1] p.7) Each nonempty word w is a power $w = r^n$ of a unique primitive word r.*

Then r and n is called the *root* and the *exponent* of w, respectively. We sometimes write $r = \sqrt{w}$. Note that $\sqrt{x^n} = \sqrt{x}$ holds for each $n \geq 2$ by Propositoin 1.1.

Two words u, v are called *conjugate*, denoted by $u \equiv v$ if there exist words x, y such that $u = xy, v = yx$. Then $\equiv$ is an equivalence relation and we call the $\equiv$-class of w the *conjugacy class* of w and denote by $cl(w)$. A language L is called *reflexive* if L is a union of conjugacy classes, i.e., $uv \in L \iff vu \in L$.

LEMMA 1.1 *([1] p.7) Two nonempty conjugate words have the same exponent and their roots are conjugate.*

LEMMA 1.2 *([4] p.7) Let $u, v \in A^+$. If $uv = vu$ holds, then $u = r^i, v = r^j$ for some primitive word r and some positive integers i, j.*

LEMMA 1.3 *([4] p.6) Let $u, v, w \in A^+$. If $uw = wv$ holds, then $u = xy, w = (xy)^kx, v = yx$ for some $x, y \in A^*$ and some nonnegative integer k.*

Let N be a submonoid of a monoid M. 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. Especially when $M = A^*$, 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).

Let L be a subset of a monoid M, the congruence $P_L = \{(u,v) \mid \text{for all } x, y \in M,\ xuy \in L \iff xvy \in L\}$ on M is called the *principal congruence* (or *syntactic congruence*) of L. We write $u \equiv v\ (P_L)$ instead of $(u,v) \in P_L$. The monoid M/P_L is called the *syntactic monoid* of L, denoted by Syn(L). The morphism σ_L of M onto Syn(L) is called the *syntactic morphism* of L. In particular when $M = A^*$, a language over A is regular if and only if Syn(L) is finite([1] p.46).

2 Extractable Codes and Insertable Codes

In this section we introduce insertable codes and extractable codes, which are extensions of well-known strong codes. We use the symbols $\subseteq$ and $\subsetneq$ to indicate subset and proper subset respectively.

2.1 Extractable Codes and Insertable Codes

Here extractable codes and insertable codes are defined below, as well as strong codes.

DEFINITION 2.1 *[3] A nonempty code $C \subseteq A^+$ is called a strong code if*

$$\text{(i) } x, y_1y_2 \in C \implies y_1xy_2 \in C^+$$
$$\text{(ii) } x, y_1xy_2 \in C^+ \implies y_1y_2 \in C^*$$

DEFINITION 2.2 *Let C be a nonempty code. Then, C is called an insertable (or extractable) code if C satisfies the condition (i) $(or\ (ii))$.*

A strong code C is described as the base of the identity $\bar{1}_L = \{w \in A^* \mid w \equiv 1(P_L)\}$ of the syntactic monoids Syn(L) of a language L. Moreover if C is finite, it is known that its structure is quite simple, i.e., it is a full uniform code.

PROPOSITION 2.1 *[3] Let $L \subseteq A^*$. Then $C = (\bar{1}_L \setminus \{1\}) \setminus (\bar{1}_L \setminus \{1\})^2$ is a strong code if it is not empty. Conversely, if $C \subseteq A^+$ is a strong code, then there exists a language $L \subseteq A^*$ such that $\bar{1}_L = C^*$.*

LEMMA 2.1 *([7],p.166) Let M be a monoid, $\theta : A^* \rightarrow M$ be a surjective morphism and $L = \theta^{-1}(P)$ for some subset P of M. There exists a unique surjective homomorphism $\phi : M \rightarrow Syn(L)$ such that $\sigma_L = \phi \circ \theta$, where σ_L is the syntactic morphism $\sigma_L : A^* \rightarrow Syn(L)$.*

COROLLARY 2.1 *Let G be an Abelian group and N is its subgroup. Let $\theta : A^* \rightarrow G$ be a surjective morphism. Then, the base of $L = \theta^{-1}(N)$ is a strong code.*

Proof) Let G/N be the quotient group of G by N. Since G/N is also an Abelian group and N is its identity, we may consider only the case that $N = \{e\}$, where e is the identity of G. By Lemma 2.1, there exists a unique surjective homomorphism $\phi : G \rightarrow Syn(L)$ such that $\sigma_L = \phi \circ \theta$. Since $\phi(e) = \sigma_L(1)$ holds, $\theta^{-1}(e) \subseteq {\sigma_L}^{-1}(1) = \bar{1}_L$. Conversely, Let $u \in \sigma^{-1}(1)$ and $x, y \in A^*$ with $xuy \in L$. Since then $x1y = xy \in L$ and $L = \theta^{-1}(e)$, $\theta(xuy) = \theta(xy) = e$. Therefore $\theta(u) = \theta(x)^{-1}\theta(y)^{-1} = (\theta(y)\theta(x))^{-1} = \theta(xy)^{-1} = e$. Therefore ${\sigma_L}^{-1}(1) = \bar{1}_L \subseteq \theta^{-1}(e)$. Thus $L == \bar{1}_L$ and its base is a strong code by Proposition 2.1. ∎

PROPOSITION 2.2 *[3] Let C be a finite strong code over A and $B =$ alph(C), where alph$(C) = \{a \in A \mid xay \in C\}$. Then $C = B^n$ for some positive integer n.*

EXAMPLE 2.1 *The followings are examples of strong codes by Corollary 2.1.*

(1) Let $G = \langle g \rangle$ be a cyclic group of order n, e be the identity of G and $\theta : X^* \rightarrow G$ be a morphism such that $\theta(a) = g$ for any $a \in X$. Then, $C =$ base$(\theta^{-1}(N)) = X^n$ is a strong code.

(2) Let $G = (\mathbf{Z},+), N = n\mathbf{Z}$*,* $\theta : \{a,b\}^* \to G, a \mapsto +1, b \mapsto -1$*. Then* $C = \mathrm{base}(\theta^{-1}(N)) = \{a^n, ab, aabb, \ldots bbaa, ba, b^n\}$ *(infinite, regular, palindromic).*

(3) $G = \langle g \rangle, o(g) = 4, N = \{e\}, e = g^4$*,* $\theta : \{a,b\}^* \to G, a \in X \mapsto g, b \mapsto g^2$*. Then* $C = \mathrm{base}(\theta^{-1}(e)) = \{a^4, aab, aba, baa, bb\} \cup (\{a^3, ab, ba\}b)^+\{a^3, ab, ba\}$.

EXAMPLE 2.2 *The followings are examples of extractable codes and insertable codes.*
(1) *A singleton* $\{w\}$ *with* $w \in \{a\}^+$ *is a strong code.*
(2) *Let* A *be a finite alphabet with* $|A| \geq 2$*. A singleton* $\{w\}$ *with* $w \in A^+ \setminus \cup_{a \in A}\{a\}^+$ *is not a strong code by Proposition 2.2 because it is not a full uniform code. But it is an extractable code. Indeed,* $w^2 = uwv$ *implies* $uv = w$*. Therefore there exist finite extractable codes which are not full uniform codes.*
(3) *The conjugacy class* $cl(ab)$ *of* ab *is an extractable code (by Proposition 2.6) but not a strong code.*
(4) $\{a^n b^n \mid n$ *is a positive integer*$\}$ *is an (context-free) extractable code but not a strong code.*
(5) a^*b *and* ba^* *are (regular) insertable codes but not strong codes. Indeed,* a^*b *is a (prefix) code and satisfies (i) in Definition 2.2. It is not an extractable code because* $ab, b \in a^*b$ *but* $a \notin a^*b$*. Similarly in case of* ba^**.*

Note that when C satisfies the condition (ii), we can easily check whether the submonoid C^* is extractable. If C^* is extractable, then C^* is biunitary (and thus free). Indeed, $uv = 1uv, u \in C^*$ implies $v = 1v \in C^*$ and $uv = uv1, v \in C^*$ implies $u = 1u \in C^*$. Then the minimal set $C = (C^* \setminus \{1\}) \setminus (C^* \setminus \{1\})^2$ of generators of C^* becomes a bifix code. Therefore both strong codes and extractable codes are necessarily bifix codes. Conversely if C is an extractable code, then $M = C^*$ forms an extractable submonoid of A^*.

Remark that an insertable submonoid M of A^*, the minimal set of generators of M is not necessarily a code. For example, if $C = \{a^2, a^3\}$, then the submonoid C^* is insertable but its minimal set C of generators is not a code.

2.2 Insertable Codes

We show that if an insertable code C over A is finite, then C is necessarily a full uniform code over some nonempty alphabet $B \subseteq A$, as well as in case of a strong code.

First of all, for a language $L \subseteq A^*$, $ins(L)$ is defined by

$$ins(L) = \{x \in A^* | \forall u \in L, u = u_1u_2 \Rightarrow u_1xu_2 \in L\}.$$

A language L such that $L \subseteq ins(L)$ is called *ins-closed.*

PROPOSITION 2.3 *[5] Let* $L \subseteq A^*$ *be a finitely generated ins-closed language and* K *be its minimal set of generators. Then:*
(i) K *contains a finite maximal prefix (suffix) code* $alph(L)$*;*
(ii) If K *is a code over* $alph(L)$ *then* $K = alph(L)^n$ *for some* $n \geq 1$*;*

COROLLARY 2.2 *If* C *is a finite insertable code then* $C = alph(C)^n$ *for some* $n \geq 1$*.*

2.3 Extractablity of Regular 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 syntactic graph of a language to check the extractability of the language.

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* $C \subseteq A^+$ *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* $x, y, z \in A^+$*.*

Let M be a general 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. $W_L = \{u \in M \mid MuM \cap L = \emptyset\}$ is called the *residue* of a subset L. If $W_L \neq \emptyset$ then W_L is an ideal of M, that is, $MW_LM \subseteq W_L$. If L is a singleton set, $L = \{c\}$, we often write c instead of $\{c\}$; thus c being disjunctive means $\{c\}$ is disjunctive, that is, $P_c = P_{\{c\}}$ is the equality relation.

Let M be a free monoid A^* and $C \subseteq A^+$ be an infix code. The syntactic monoid Syn(C) of C has the identity element $e = \{1\}$ since the set $\{1\}$ is a P_C-class. Syn(C) has a zero element $0 = W_C/P_C$ since $W_C \neq \emptyset$ is a P_C-class. 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, $c = C/P_C$. Then the following theorem holds:

THEOREM 2.1 *[10] The following conditions* (i) *and* (ii) *on a monoid* M *with identity* e *are equivalent:*

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

(ii) (a) *$M \setminus \{e\}$ is a subsemigruop of M;*

(b) *M has a disjunctive zero;*

(c) *there exists $0 \neq c \in core(M) \cap Annihil(M)$.*

PROPOSITION 2.4 *Let C be an infix code and $M =$ Syn(L) be its syntactic monoid Let c be a P_C-class of C, that is $0 \neq c \in core(M) \cap Annihil(M)$. Then,*

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

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

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

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

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

(1) $V =$ Syn(L); the syntactic monoid of L.
(2) $E = \{(a, b) \in V \times V \mid ab \in \sigma_L(L)\}$, 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 $(1 \leq i \leq n)$. Proposition 2.4 can be stated in terms of graph.

PROPOSITION 2.5 *Let C be an infix code and $G_C = (V, E)$ be the 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 $v_0 = v_2$.*

Extractability of Uniform Codes

We summarize some results on extractability of uniform codes over a finite nonempty alphabet A.

PROPOSITION 2.6 *Let G be a group and let H be a normal subgroup of G. Let $\varphi : A^* \to G$ be a surjective morphism. If $C = \varphi^{-1}(H) \cap A^n$ $(n > 0)$ is nonempty, then it is an extractable reflective uniform code.*

EXAMPLE 2.3 *Let B be a nonempty subset of an alphabet A and n, k $(k \leq n)$ be positive integers. Set $U = \{w \in A^n \mid |w|_B = k\}$ where $|w|_B$ is the number of occurrences of elements of B in w. Then U is an extractable code.*

PROPOSITION 2.7 *Let n be an integer with $n \geq 2$. Let $f_1,\ f_2, \ldots f_k$ be distinct words with $|f_i| = |f_j|$ for any $i, j \in \{1, 2, \ldots, k\}$. Then U^* is extractable, where $U = \{f_1{}^n,\ f_2{}^n, \ \ldots\ f_k{}^n\}$.*

PROPOSITION 2.8 *Let $x, y \in A^*$ with $|x| = |y| > 0$ and $C = \{x^2, xy, yx, y^2\}$. C^* is extractable.*

3 Extractability of Conjugacy Classes

Here we state new results on the extractability of conjugacy classes. The extractablity of a conjugacy class seems to be affected by the periodicity of the class. First we investigate the relation between the extractablity and the conjugacy exponent of a word. Secondly we introduce the deletion closure operation to the free monoids generated by extractable uniform codes.

3.1 Periodicity and Conjugacy Classes

A period of $w = a_1 \dots a_n$ with $a_i \in A$ is an integer p such that $a_{p+i} = a_i$ for $i = 1, \dots, n-p$. The smallest one among periods of w is called *the period* of w, denoted by $p(w)$. We call the value defined by $\min\{p(u) \mid u \in cl(w)\}$ *the conjugate period* of w, denoted by $p^\circ(w)$. The rate $|w|/p^\circ(w)(\geq 1)$ of the length $|w|$ of w for the conjugate period $p^\circ(w)$ is called *the conjugate exponent* of w or $cl(w)$, denoted by $e^\circ(w)$

PROPOSITION 3.1 *Let $w \in A^+$ be not a primitive word and $cl(w)$ be its conjugacy class. Then $cl(w)^*$ is extractable.*

PROPOSITION 3.2 *Let $w \in A^+$ be a primitive word of the form $(uv)^n u$ with $n \geq 2$ and $u, v \in A^+$, and $cl(w)$ be its conjugacy class. Then $cl(w)^*$ is not extractable.*

PROPOSITION 3.3 *Let $w \in A^+$ and $e^\circ(w) = 1$. Then $cl(w)^*$ is extractable.*

Proof) Let $w_1, w_2, w \in cl(w)$. We show that $w_1 w_2 = uwv$ for some $u, v \in A^*$ implies $u = 1$ or $u = w_1$. Suppose that $u \neq 1$. By Lemma 1.3, $u = xy, w_1 = xy(xy)^k x, w = (xy)^k xyx$ for some $x, y \in A^*$ and some $k \geq 0$. Since $e^\circ(w) = 1$ holds and thus w is primitive, $x = 1$ and $k = 0$. $u \neq 1$ yields that $u = w_1 = w = w_2$. By Proposition 2.1, $cl(w)^*$ is extractable. ▮

Thus, $cl(w)^*$ is extractable if w is a nonprimitive word w, that is, $e^\circ(w)$ is an integer ≥ 2. $cl(w)^*$ is not extractable if $e^\circ(w)$ is a noninteger ≥ 2. If $1 \leq e^\circ(w) < 2$, $cl(w)^*$ is almost extractable. That is, there exists a word w such that $1 \leq e^\circ(w) < 2$ and $cl(w)^*$ is not extractable. For example, $w = abbabbabab$ is of length 10 and $e^\circ(w) = |w|/p^\circ(w) = 10/7$ but $cl(w)^*$ is not extractable. Table 1 summarizes the extractability and the periodicity of conjugacy classes. In the table e denotes the conjugate exponent of a conjugacy class.

Table 1. Summary of the extractability and the periodicity of words.

Primitive			not Primitive
$e = 1$	$1 < e < 2$	$e \geq 2$	
		noninteger	integer
Extractable	Extractable not Extactable	not Extactable	Extractable
(Prop.3.3)	Unknown	(Prop. 3.2)	(Prop.3.1)

Table 2 shows the extractability of conjugacy classes of words of length ≤ 20 on a binary alphabet. In the table the column of **Len** indicates the length n of (the conjugacy classes of) words. **not Primitive** and **Primitive** mean repetitive and primitive. e is the conjugacy period of a conjuacy class $cl(w)$, that is $e = e^\circ(w)$. #word and #class are the numbers of words and conjugacy classes of length n respectively. Ext (or $\overline{Ext}$) indicates the number of conjugacy classes of length n which are extractable codes (or not extractable codes). rate is the number of conjugacy classes which are of conjugacy period < 2 and not extractable code divided by the number of all conjugacy classes of length n.

For $n = 1, 2, 3, 4, 6$, the conjugacy class $cl(w)$ of each word of length n is extractable code. The conjugacy classes of the shortest length on $\{a, b\}$ which are not extractable codes are $cl(ababa)$ and $cl(babab)$.

Table 2. Extractability and the periodicity of words over a binary alphabet.

			Primitive				not Primitive		
			$1 \leq e < 2$		$2 \leq e$		$2 \leq e$		
			rational		noninteger		integer		
Len	#word	#class	Ext	$\overline{Ext}$	Ext	$\overline{Ext}$	Ext	$\overline{Ext}$	rate
1	2	2	2	0	0	0	0	0	0
2	4	3	1	0	0	0	2	0	0
3	8	4	2	0	0	0	2	0	0
4	16	6	3	0	0	0	3	0	0
5	32	8	4	0	0	2	2	0	0
6	64	14	9	0	0	0	5	0	0
7	128	20	12	0	0	6	2	0	0
8	256	36	26	0	0	4	6	0	0
9	512	60	40	0	0	8	4	0	0
10	1024	108	85	2	0	12	9	0	0.023
11	2048	188	160	0	0	26	2	0	0
12	4096	352	317	2	0	16	17	0	0.00627
13	8192	632	574	2	0	54	2	0	0.00347
14	16384	1182	1099	6	0	56	21	0	0.00543
15	32768	2192	2082	2	0	98	10	0	0.00096
16	65536	4116	3960	8	0	112	36	0	0.00202
17	131072	7712	7470	6	0	234	2	0	0.00080
18	262144	14602	14312	16	0	204	70	0	0.00112
19	524288	27596	27104	8	0	482	2	0	0.00030
20	1048576	52488	51881	20	0	476	111	0	0.00039

3.2 Deletion Closure and Extractability of Conjugacy Classes

Let L_1, L_2 be languages. The *deletion* of L_2 from L_1 is defined as $L_1 \longrightarrow L_2 = \{u_1u_2 \mid u_1wu_2 \in L_1, w \in L_2\}$. A language L is *del-closed* iff $L \longrightarrow L \subset L$. The intersection of all the del-closed languages containing L is called the *del-closure* of L.

For a language L, $D(L)$ is defined by $D(L) = \bigcup_{k \geq 0} D_k(L)$, where $D_0(L) = L$ and $D_{k+1}(L) = D_k(L) \longrightarrow (D_k(L) \cup \{1\})$

PROPOSITION 3.4 *[5] $D(L)$ is the del-closure of a language L.*

PROPOSITION 3.5 *Let M be a submonoid of A^*. Then, $D(M) = \bigcup_{k \geq 0} D_k(M)$ is also a submonoid of A^*.*

LEMMA 3.1 *Let $k \geq 0$. $x, y \in D_k(M) \Longrightarrow xy \in D_{2k}(M)$*

Proof) In case of $k = 0$, the statement is trivial. Assume that the statement holds for $k \geq 0$. $x, y \in D_{k+1}(M) = D_k(M) \longrightarrow D_k(M)$. Let $x = x_1x_2$ with $x_1zx_2, z \in D_k(M)$ and $y = y_1y_2$ with $y_1wy_2, w \in D_k(M)$. By hypothesis, $x_1zx_2y_1wy_2 \in D_{2k}(M)$. This implies $xy \in D_{2k+2}(M)$. ∎

Proof of Proposition 3.5) Since $1 \in M = D_0(M) \subseteq D_1(M) \subseteq \cdots \subseteq D(M)$ holds, the empty word 1 is in $D(M)$. Let $x, y \in D(M)$. There exists some integer k such that $x, y \in D_k(M)$. By Lemma 3.1, $xy \in D_{2k} \subseteq D(M)$. ∎

Note that each $D_k(M)(k \geq 1)$ is not necessarily a submonoid but it contains 1.

LEMMA 3.2 *Let $\emptyset \neq C \subset A^n$. Then, $D_k(C^*) \subset (D(C^*) \cap A^n)^*$ for each $k \geq 0$.*

Proof) In case of $k = 0$. Since $D_0(C^*) = C^* = (C^* \cap A^n)^* = (C^* \cap A^n)^* = (D_0(C^*) \cap A^n)^*$ holds by definition, the statement is true.

Assume that the statement holds for $k \geq 0$. Let $x \in D_{k+1}(C^*) = D_k(C^*) \longrightarrow D_k(C^*)$. $x = x_1x_2$ with $x_1zx_2, z \in D_k(C^*)$. By the hypothesis, we can write

$$\begin{aligned} x_1zx_2 &= u_1u_2\dots u_\ell, \quad & u_i &\in D(C^*) \cap A^n \\ z &= z_1z_2\dots z_m, & z_i &\in D(C^*) \cap A^n \end{aligned}$$

$x_1 = u_1 \dots u_{s-1}u'$, $x_1 = u''u_{t+1}\dots u_\ell$ and $|u'u''| = n$. Since $x, u_1 \dots u_{s-1}, u_{t+1} \dots u_\ell \in D(C^*)$ and $D(C^*)$ is del-closure, we have $u'u'' \in D(C^*)$ and thus $x \in (D(C^*) \cap A^n)^*$. Hence the statement is true for any integer $k \geq 0$. ▮

PROPOSITION 3.6 *Let $\emptyset \neq C \subset A^n$ and $D(\overset{\circ}{C}^*)$ the minimal set of generators of $D(C^*)$. That is,*

$$D(\overset{\circ}{C}^*) \overset{\text{def}}{=} (D(C^*) \setminus 1) \setminus (D(C^*) \setminus 1)^2.$$

Then, $D(\overset{\circ}{C}^*) \subseteq A^n$ that is, a uniform code over A containing C.

Proof) Let $x \in D(\overset{\circ}{C}^*)$. There exists some integer k such that $x \in D_k(C^*)$. By Lemma 3.2, $x = x_1 \dots x_m$, $x_i \in D(C^*) \cap A^n$ $(1 \leq i \leq m)$. Since x is a generator of the submonoid $D(C^*)$ of A^*, $m = 1$ must hold. Thus $D(\overset{\circ}{C}^*) = D(C^*) \cap A^n \subseteq A^n$. Moreover $C \subseteq D(C^*)$ implies $C \subseteq D(C^*) \cap A^n$. ▮

Remark that even if C is reflexive, $D(\overset{\circ}{C}^*)$ is not necessarily reflexive. For example, let $w = ababa$ and $C = cl(w)$. $D(\overset{\circ}{C}^*) = cl(w) \cup \{aabba, abbaa, baaab\}$ but $bbaaa \notin D(\overset{\circ}{C}^*)$.

If $D(C^*)$ is a submonoid of A^*, then we can define a language operator **EXT** by

$$C^{\mathbf{EXT}} \overset{\text{def}}{=} D(\overset{\circ}{C}^*)\ .$$

EXAMPLE 3.1
(1) Let $L = \{a^{i_1}, a^{i_2}, \dots, a^{i_n}\}$ be a language over $\{a\}$ but not a code. Then $L^{\mathbf{EXT}} = \{a^d\}$, where d is the greatest common divisor of $i_1, i_2, \dots, i_n$.
(2) If C is an extractable code, then $C^{\mathbf{EXT}} = C$.

The followings are problems related to operator **EXT**.
(1) When is $D(C^*)$ a submonoid of A^* ? and then when is $C^{\mathbf{EXT}}$ a code ?
(2) If C is an infix code, $C^{\mathbf{EXT}}$ is also an infix code ?
(3) If C, C_1, C_2 are uniform codes, are the following equations true ?

$$(C_1 \cup C_2)^{\mathbf{EXT}} = {C_1}^{\mathbf{EXT}} \cup {C_2}^{\mathbf{EXT}}.$$
$$(C_1 \cap C_2)^{\mathbf{EXT}} = {C_1}^{\mathbf{EXT}} \cap {C_2}^{\mathbf{EXT}}.$$
$$(C^c)^{\mathbf{EXT}} = (C^{\mathbf{EXT}})^c.$$

References

[1] J. Berstel and D. Perrin, *Theory of Codes*, Pure and Applied Mathematics (Academic Press, 1985).

[2] A. de Luca and S. Varricchio, *Finiteness and Regularity in Semigroups and Formal Languages*, Monographs on Theoretical Computer Science · An EATCS Series (Springer, July 1999).

[3] H.J.Shyr, Strong codes, *Soochow J. of Math. and Nat. Sciences* **3** (1977) 9–16.

[4] H.J.Shyr, *Free monoids and Languages*, Lecture Notes (Hon Min book Company, Taichung, Taiwan, 1991).

[5] M. Ito, L. Kari and G. Thierrin, Insertion and deletion closure of languages, *Theoretical Computer Science* **183** (1997) 3–19.

[6] J.M.Howie, *Fundamentals of Semigroup Theory*, London Mathematical Society Monographs New Series 12, London Mathematical Society Monographs New Series 12 (Oxford University Press, 1995).

[7] G. Lallement, *Semigroups and combinatorial applications* (John Wiley & Sons, Inc., 1979).

[8] M. Lothaire, *Combinatorics on Words*, Encyclopedia of Mathematics and its Applications, Vol. 17 (Cambridge University Press, 1983).

[9] T. Moriya and I. Kataoka, Syntactic congruences of codes, *IEICE TRANSACTIONS on Information and Systems* **E84-D**(3) (2001) 415–418.

[10] M.Petrich and G.Thierrin, The syntactic monoid of an infix code, *Proceedings of the American Mathematical Society* **109**(4) (1990) 865–873.

[11] G. Rozenberg and A. Salomaa, *Handbook of Formal Languages, Vol.1 WORD, LANGUAGE, GRAMMAR* (Springer, 1997).

[12] G. Tanaka, Y. Kunimochi and M. Katsura, Remarks on extractable submonoids, *Technical Report kokyuroku, RIMS, Kyoto University* **1655** (6 2009) 106–110.

[13] S.-S. Yu, *Languages and Codes* (Tsang Hai Book Publishing Company, Taiwan, 2005).

[14] S. Yu, A characterization of intercodes, *International Journal of Computer Mathematics* **36**(1-2) (1990) 39–45.

[15] L. Zhang, Rational strong codes and structure of rational group languages, *Semigroup Forum*, **35**(1), Springer (1986), pp. 181–193.