Bases of extractable closures of languages

Yoshiyuki Kunimochi
Faculty of Informatics,
Shizuoka Institute of Science and Technology

abstract Deletion and insertion are interesting and common operations which often appear in string rewriting systems. Extractable and Insertable submonoids in free monoids generated by finete alphabets allow to perform Deletion and Insertion operations, respectively. A submonoid $N \subset A^*$ is called extractable (or insertable) if $x, uxv \in N$ implies $uv \in N$(or $x, uv \in N$ implies $uxv \in N$). The code C is called extractable (or insertable) if the submonoid C^* is extractable (or insertable)[7]. Both extractable and insertable codes are identical to well-known strong codes, which is deeply related to syntactic monoids of languages. This paper deals with insertability and mainly extractablity of codes.

After the preliminaries in the first section, we summarize the fundamental properties of these codes. In the second section, 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. The infinite part of these codes is still unknown for us. So in the remaining sections we mainly investigate extractability codes in limited classes of codes.

In the last section, we deal with the smallest extractable submonoid $D(L^*)$ containing a ginven language L, where D is the del-clusure[5]. Since the base of $D(L^*)$ is a bifix code, denoted by $L^{\rightarrow}$. So we give the definition of the language operation which convert a language L to $L^{\rightarrow}$ and investigate this operation.

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 $w = a_1 a_2 \ldots a_n$ 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* (or *outfix*) code if $u, xuy \in C$ (or $xy, xuy \in C$) implies $x = y = 1$ (or $u = 1$). 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. The symbols $\subset$ and $\subsetneq$ are used for a subset and a proper subset respectively.

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 $r = \sqrt{w}$.

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 $u = xy, w = (xy)^k x, v = yx$ 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 $L \subset A^*$ 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.

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

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

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

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 are 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 some 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 $L \subset A^*$. 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 $C \subset A^+$ is a strong code, then there exists a language $L \subset A^*$ such that $\bar{1}_L = C^*$.*

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

EXAMPLE 2.1 (1) *A singleton $\{w\}$ with $w \in \{a\}^+$ is a strong code. $\{w\}$ with $w \in A^+ \setminus \bigcup_{a \in A}\{a\}^+$ is not a strong code but it is an extractable code. Therefore there exist finite extractable codes which are not full uniform codes.*
(2) *The conjugacy class $cl(ab)$ of ab is an extractable code but not a strong code.*
(3) *$\{a^n b^n \mid n$ is an integer$\}$ is an (context-free) extractable code but not a strong code.*
(4) *a^*b and ba^* are (regular) insertable codes but not strong codes.*

Note that when C satisfies the condition (ii), we can easily check that 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 are not necessarily a code.

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 $B \subset A$, as well as in case of a strong code. First of all, for a language $L \subset A^*$, $ins(L)$ 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 $L \subset ins(L)$ is called *ins-closed*.

PROPOSITION 2.3 [5] *Let $L \subset A^*$ 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) *K is a code over $alph(L)$ then $K = alph(L)^n$ for some $n \geq 1$;*

COROLLARY 2.1 *If C is a finite insertable code then $C = alph(C)^n$ for some $n \geq 1$.*

Extractablity of Regular Infix Codes

Our aim here 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.

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

LEMMA 2.2 *Let C be an extractable code. If C is an outfix code, then C is an infix code.*

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, $MLM \subset W_L$. 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 $C \subset A^+$ 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 [11] *The following conditions on a monoid M with identity e are equivalent:*
(i) *M is isomorphic to the syntactic monoid of an infix code C.*
(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) core$(M) = \{c, 0\}$ with $c \in$ Annihil(M).
(iv) *(α), (δ);*
(ζ) there exists $0 \neq c \in$ core$(M) \cap$ Annihil(M).

PROPOSITION 2.4 *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 $0 \neq c \in$ core$(M) \cap$ Annihil(M). Then,*

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

$$c = f_0f_1 = f_1f_2 = f_2f_3 \;\Rightarrow\; c = f_0f_3 \quad \text{for any } f_0,\, f_1,\, f_2,\, f_3 \in M.$$

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

$$c = f_0f_1 = f_1f_2 \;\Rightarrow\; f_0 = f_2 \quad \text{for any } f_0,\, f_1,\, f_2 \in M.$$

Extractability of Uniform Codes

First we consider some kinds of extractable codes which is a uniform code over a finite nonempty alphabet A.

PROPOSITION 2.5 *Let G be a group and H a normal subgroup of G. Let $\varphi : A^* \to G$ be a surjective morphism. Then if $C = \varphi^{-1}(H) \cap A^n$ $(n > 0)$ is nonempty, then it is an extractable reflective uniform code.*

EXAMPLE 2.2 *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 $\in B$ in w. Then U is an extractable code.*

PROPOSITION 2.6 *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 2.7 *Let $x, y \in A^*$ with $|x| = |y| > 0$ and $C = \{x^2, xy, yx, y^2\}$. C^* is extractable.*

Extractability of Conjugacy Classes

The extractablity of a conjugacy class is affected by the periodicity of the class.

PROPOSITION 2.8 *Let $w \in A^+$ be not a primitive word and $cl(w)$ be its conjugacy class. Then $cl(w)^*$ is extractable.*

PROPOSITION 2.9 *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.*

We slightly touch the periodicity and extractability. A period of $w = a_1 \ldots a_n$ with $a_i \in A$ is an integer p such that $a_{p+i} = a_i$ for $i = 1, \ldots, n - p$. The smallest one among periods of w is called *the period* of w, denoted by $p(w)$. We call the value defined by $\max\{p(u) \mid u \in cl(w)\}$ *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 exponet* of w, denoted by $e^\circ(w)$

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. $w = abbabbabab$ is of length 10 and $e^\circ(w) = |w|/p^\circ(w) = 10/7$ but $cl(w)^*$ is not extractable.

3 Extractable Submonoid of a Language

An extractable submonoid $M \subset A^*$ satisfies the condition that $x, uxv \in M$ implies $uv \in M$. Since M is biunitary, the base C of M must be a bifix code. Moreover $M = C^*$ is a free submonoid of A^*. It is a natural way to consider the smallest extractable submonoid $D(L^*)$ containing a ginven language L. The base of $D(L^*)$ become a bifix code, denoted by $L^\rightarrow$. We give the definition of the language operation which convert a language L to $L^\rightarrow$.

Deletion Closure

DEFINITION 3.1 [5] *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.*

DEFINITION 3.2 [5] *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 3.1 [5] *$D(L)$ is identical to the del-closure of a language L.*

If a submonoid $M \subset A^*$ is extractable, then M is del-closed, $D_k(M) \subset M$ for any $k \geq 0$ and thus $D(M) = M$. Let M be an extractable submonoid containing a language L. Then $L^* \subset M$, $D(L^*) \subset D(M) = M$. The following proposition 3.2 implies that the del-closure of a submonoid is also a submonoid. This concludes that $D(L^*)$ is the smallest extractable submonoid containing L and its base is a bifix code. To prove PROPOSITION 3.2, we use LEMMA 3.1.

PROPOSITION 3.2 *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 M be a submonoid of A^* and $k \geq 1$. Then, $x, y \in D_k(M)$ implies $xy \in D_{2k}(M)$.*

Proof of Lemma 3.1) Note that $1 \in M$ and $1 \in D_k(M)$ for each $k \geq 0$ since M is a submonoid.
In case of $k = 1$, let $x, y \in D_1(M) = M \longrightarrow M$. There exist $z, w \in M$ such that $x = x_1x_2, y = y_1y_2, x_1zx_2 \in M$ and $y_1wy_2 \in M$. Since M is a submonoid of A^*, we have $x_1zx_2y_1wy_2 \in M$ and $x_1x_2y_1wy_2 \in D_1(M) = M \longrightarrow M$. Since $w \in D_1(M)$, $xy = x_1x_2y_1y_2 \in (D_1(M) \longrightarrow D_1(M)) \subset D_2(M)$.

Next, assume that the statement holds for $k \geq 1$. $x, y \in D_{k+1}(M) = D_k(M) \longrightarrow D_k(M)$. 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, $x_1zx_2y_1wy_2 \in D_{2k}(M)$. Since $w \in D_{2k}(M) and w \in D_{2k+1}(M)$, $x_1x_2y_1wy_2 \in D_{2k}(M) \longrightarrow D_{2k}(M) \subset D_{2k+1}(M)$ and $xy = x_1x_2y_1y_2 \in D_{2k+1}(M) \longrightarrow D_{2k+1}(M) \subset D_{2k+2}(M)$. This implies $xy \in D_{2k+2}(M)$. ∎

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

the base of the del-clusure of an extractable submonoid

EXAMPLE 3.1 (1) *Let $C = cl(ab) = \{ab, ba\}$. Since C^* is an extractable submonoid, we have $D(C^*) = C^*$.*
(2) *Let $C = cl(ababa) = \{ababa, babaa, abaab, baaba, aabab\}$. Then the base C_1 of $D(C^*)$ is $C_1 = C \cup \{aabba, abbaa, baaab\}$ and $D(C^*) = D({C_1}^*)$.*

Let L be a language over A. We denote by $L^{\rightarrow}$ the minimal set of generators of the smallest extractable submonoid $D(L^*)$ containing L. That is,

$$L^{\rightarrow} \stackrel{\text{def}}{=} (D(L^*) \setminus 1) \setminus (D(L^*) \setminus 1)^2.$$

If L is an extractable code, L^* is an extractable submonoid, $D(L^*) = L^*$, and thus $L^{\rightarrow} = L$. Extractability is closed under the language operator $^{\rightarrow}$. We show that uniformality is also closed under $^{\rightarrow}$.

LEMMA 3.2 *Let C be a nonempty uniform code of length n, that is, $\emptyset \neq C \subset A^n$. Then, $D_k(C^*) \subset (D(C^*) \cap A^n)^*$ for each $k \geq 0$.*

Proof) In case of k = 0, trivial. Assume that the statement is true for $k \geq 0$. Let $x \in D_{k+1}(C^*) = D_k(C^*) - D_k(C^*)$. $x = x_1x_2$ with $x_1zx_2, z \in D_k(C^*)$. Since $D(C^*)$ is a extractable submonoid, by induction hypothesis, we have $x_1zx_2, z \in D(C^*)$ and $x_1x_2 \in D(C^*)$. Both $|x_1zx_2|$ and $|z|$ are multiples of n. Therefore $x \in (D(C^*) \cap A^n)^*$ ∎

PROPOSITION 3.3 *Let $\emptyset \neq C \subset A^n$. Then,the following statements hold.*
(1) *$C^{\rightarrow}$ is a subset of A^n (a uniform code over A) containing C.*
(2) *If C is reflexive, then $C^{\rightarrow}$ is also reflexive.*

Proof) (1) Trivial by Lemma3.2.
(2) We can easily check that $D_k(C^*)$ is reflexive for any $k \geq 0$. $D(C^*)$ and its base $C^{\rightarrow} = D(C^*) \cap A^n$ are also reflexive. ∎

The following issues remain unsolved.
(1) If C is an infix code, $C^{\rightarrow}$ is also an infix code ?
(2) If C, C_1, C_2 are uniform codes, the following equations are true ?
$(C_1 \cup C_2)^{\rightarrow} = {C_1}^{\rightarrow} \cup {C_2}^{\rightarrow}$.
$(C_1 \cap C_2)^{\rightarrow} = {C_1}^{\rightarrow} \cap {C_2}^{\rightarrow}$.
$(C^{\rightarrow})^c = (C^c)^{\rightarrow}$, where ^{c} means the set complement.

4 Conclusion

We introduce extractable and insertable codes, which generate submonoids that allow deletion and insertion operations, respectively. We summarize the definition and the properties of these codes in limited language families.

In the last section, we deal with the smallest extractable submonoid $D(L^*)$ containing a ginven language L, where D is the del-clusure[5]. So we give the definition of the language operation which convert a language L to the base of $D(L^*)$, denoted by $L^{\rightarrow}$, and investigate this operation. We just start to study these submonoids and many interesting problems remain unsolved.

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*, 3:9–16, 1977.

[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*, 183:3–19, 1997.

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

[7] Y. Kunimochi. Some properties of extractable codes and insertable codes. *International Journal of Foundations of Computer Science*, 27(3):327–342, 2016.

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

[9] M. Lothaire. *Combinatorics on Words*, volume 17 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1983.

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

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

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

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

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

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

[16] L. Zhang. Rational strong codes and structure of rational group languages. In *Semigroup Forum*, volume 35, pages 181–193. Springer, 1986.

Faculty of Informatics,
Shizuoka Institute of Science and Technology
Toyosawa 2200-2, Fukuroi-shi, Shizuoka 437-8555,
JAPAN
Email: kunimochi.yoshiyuki@sist.ac.jp