On Commutativity of Extractable Codes

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 submonoids and insertable submonoids of free monoids generated by finite alphabets allow to perform deletion operations and insertion operations, respectively. A submonoid $N \subset A^*$ is called extractable (resp. insertable) if $x, uxv \in N$ implies $uv \in N$(resp. $x, uv \in N$ implies $uxv \in N$). The code C is called extractable (resp. insertable) if the submonoid C^* is extractable (resp. insertable)[7]. Both extractable and insertable codes are identical to well-known strong codes, which is deeply related to the identities of syntactic monoids of languages. This paper deals with the commutativity of extractable codes.

After the preliminaries in the first section, we summarize the fundamental properties of codes above in the second section. In the last section, we deal with commutative extractable codes. At first the language operators **S** and **Q**, which make languages commutative, are introduced. We show that a commutative extractable code is finite.

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*(resp. *suffix*) of a word $w \in A^*$ if there is a word $x \in A^*$ such that $w = ux$(resp. $w = xu$). 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 $u, uv \in C$ (resp.$u, vu \in C$) implies $v = 1$. C is called a *bifix* code if C is both a prefix code and a suffix code. 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 $\mathrm{Syn}(L)$. The morphism σ_L of M onto $\mathrm{Syn}(L)$ is called the *syntactic morphism* of L. In particular when $M = A^*$, a language $L \subset A^*$ is regular if and only if $\mathrm{Syn}(L)$ 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*

$$\text{(i)}\ x, y_1 y_2 \in C \implies y_1 x y_2 \in C^+$$
$$\text{(ii)}\ x, y_1 x y_2 \in C^+ \implies y_1 y_2 \in C^*$$

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 $\mathrm{Syn}(L)$ 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 $C = (\overline{1}_L \setminus 1) \setminus (\overline{1}_L \setminus 1)^2$ 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 $\overline{1}_L = C^*$.*

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

$$ins(L) = \{x \in A^* | \forall u \in L, u = u_1 u_2 \Rightarrow u_1 x u_2 \in L\}.$$

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 $n \geq 1$.*

3 Stack and Queue Operations

In this section, we introduce the language operators **S** and **Q** and investigate their properties. In this section, we denote the empty word by ϵ instead of 1, the words over an alphabet A of shorter length than n, $\{w \in A^* \mid |w| \leq n\}$ by $A^{\leq n}$.

DEFINITION 3.1 Let $w = a_0a_1 \cdots a_{n-1}(a_i \in A)$ be a word of length n over A. We denote an element $(i, u) \in Q = \{0, 1, \cdots, n\} \times A^{\leq n}$ by $[i, u]$ and define the finite automaton $M_w^{\mathbf{S}} = (Q, A, \delta_w^{\mathbf{S}}, [0, \epsilon], \{[n, \epsilon]\})$, where

$$\begin{array}{ll} (\mathbf{THRU}) & \delta_w^{\mathbf{S}}([i, v], a_i) = [i + 1, v], \\ (\mathbf{PUSH}) & \delta_w^{\mathbf{S}}([i, u], \epsilon) = [i + 1, a_i u], \\ (\mathbf{POP}) & \delta_w^{\mathbf{S}}([i, au], a) = [i, u], \end{array}$$

for $0 \leq \forall i < n, \forall v \in A^{\leq n}$ and $\forall u \in A^{<n}$. The stack operation $\mathbf{S} : A^* \to 2^{A^*}$ is defined by the language accepted by the finite automaton $M_w^{\mathbf{S}}$

$$\mathbf{S}(w) \stackrel{\text{def}}{=} \{u \in A^* | \delta_w^{\mathbf{S}}([0, \epsilon], u) = [n, \epsilon]\}$$

∎

DEFINITION 3.2 Let $w = a_0a_1 \cdots a_{n-1}(a_i \in A)$ be a word of length n over A. We denote a state $(i, u) \in Q = \{0, 1, \cdots, n\} \times A^{\leq n}$ by $[i, u]$ and define the finite automaton $M_w^{\mathbf{Q}} = (Q, A, \delta_w^{\mathbf{Q}}, [0, \epsilon], \{[n, \epsilon]\})$, where

$$\begin{array}{ll} (\mathbf{THRU}) & \delta_w^{\mathbf{Q}}([i, v], a_i) = [i + 1, v], \\ (\mathbf{PUSH}) & \delta_w^{\mathbf{Q}}([i, u], \epsilon) = [i + 1, ua_i], \\ (\mathbf{POP}\ ) & \delta_w^{\mathbf{Q}}([i, au], a) = [i, u], \end{array}$$

for $0 \leq \forall i < n, \forall v \in A^{\leq n}$ and $\forall u \in A^{<n}$. The queue operation $\mathbf{Q} : A^* \to 2^{A^*}$ is defined by the language accepted by the FA $M_w^{\mathbf{Q}}$

$$\mathbf{Q}(w) \stackrel{\text{def}}{=} \{u \in A^* | \delta_w^{\mathbf{Q}}([0, \epsilon], u) = [n, \epsilon]\}$$

∎

EXAMPLE 3.1 Let $w = abcd$, $A = \{a, b, c, d\}$, $n = |w| = 4$ and $Q = \{0, 1, 2, 3, 4\} \times A^{\leq 4}$. The transition of the finite automaton $M_w^{\mathbf{S}}$ for $dcab$ is depicted in Figure.1. Therefore $cdba \in \mathbf{S}(w)$.

$$\delta_w^{\mathbf{S}} \quad : \quad [0, \epsilon] \xrightarrow{\epsilon} [1, a] \xrightarrow{\epsilon} [2, ba] \xrightarrow{c} [3, ba] \xrightarrow{d} [4, ba] \xrightarrow{b} [4, a] \xrightarrow{a} [4, \epsilon]$$

The transition of the finite automaton $M_w^{\mathbf{Q}}$ for $cdab$ is depicted in Figure 2. Therefore $cdab \in \mathbf{Q}(w)$.

$$\delta_w^{\mathbf{Q}} \quad : \quad [0, \epsilon] \xrightarrow{\epsilon} [1, a] \xrightarrow{\epsilon} [2, ab] \xrightarrow{c} [3, ab] \xrightarrow{d} [4, ab] \xrightarrow{a} [4, b] \xrightarrow{b} [4, \epsilon]$$

The operators $\mathbf{S}$ and $\mathbf{Q}$ are extended from words to languages in the natural way. The powers of these operators are also defined. And then we give some examples.

DEFINITION 3.3 For a language L over A, $\mathbf{S}(L)$ and $\mathbf{Q}(L)$ are defined by

$$\mathbf{S}(L) \stackrel{\text{def}}{=} \bigcup_{w \in L} \mathbf{S}(w), \mathbf{Q}(L) \stackrel{\text{def}}{=} \bigcup_{w \in L} \mathbf{Q}(w).$$

For a language L over A, the powers of the operators are defined by

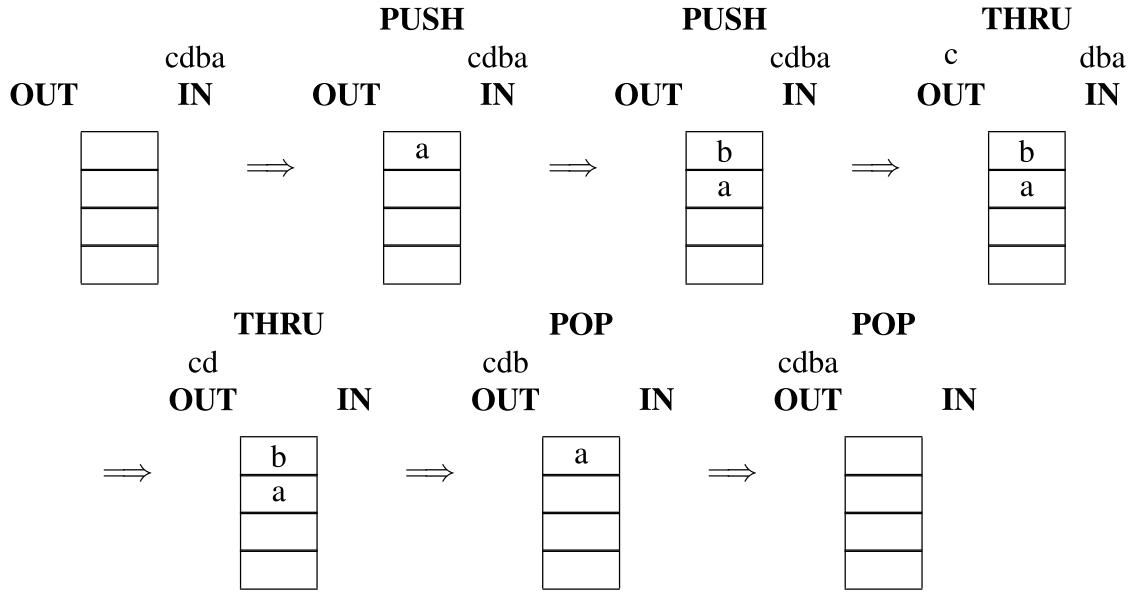

Figure 1. A transition of FA $M_w^{\mathbf{S}}$ for $w = cdba$

$$\begin{array}{ll} \mathbf{S}^0(L) \stackrel{\text{def}}{=} L, & \mathbf{Q}^0(L) \stackrel{def}{=} L, \\ \mathbf{S}^n(L) \stackrel{\text{def}}{=} \mathbf{S}(\mathbf{S}^{n-1}(L)), & \mathbf{Q}^n(L) \stackrel{\text{def}}{=} \mathbf{Q}(\mathbf{Q}^{n-1}(L)) \quad \text{for } n > 0. \\ \mathbf{S}^\infty(L) \stackrel{\text{def}}{=} \bigcup_{n\geq 0} \mathbf{S^n}(L), & \mathbf{Q}^\infty(L) \stackrel{\text{def}}{=} \bigcup_{n\geq 0} \mathbf{Q^n}(L). \end{array}$$

EXAMPLE 3.2 The followings are examples of the operators $\mathbf{S}$ and $\mathbf{Q}$.

(1) For any word w over A and any $n \geq 0$, $\mathbf{S}^n(w)$ (resp.$\mathbf{Q}^n(w)$) is finite and a uniform code (each word is of length $|w|$).

(2) For any word w over a **binary** alphabet $A = \{0, 1\}$, $\mathbf{S}(w) = \mathbf{Q}(w) = \{u||u|_a = |w|_a \text{ for } a \in A\}$, which is a uniform code and a commutative extractable code (but not necessarily a strong code).

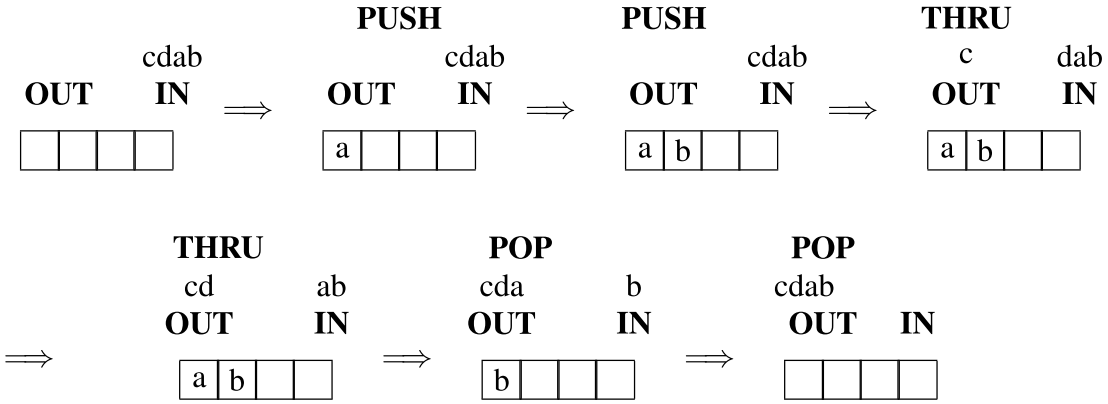

Figure 2. A transition of FA $M_w^{\mathbf{Q}}$ for $w = cdab$

(3) For the conjugacy class $cl(w)$ of a word w over a **binary** alphabet, $\mathbf{S}(cl(w)) = \mathbf{S}(w) = \mathbf{Q}(cl(w)) = \mathbf{Q}(w)$

(4) For the word 012 over a ternary alphabet $\{0,1,2\}$, $\mathbf{S}(012) = \{012, 021, 102, 120, 210\}$ and $201 \notin \mathbf{S}(012)$. $\mathbf{S}^n(012) = \{012, 021, 102, 120, 210, 201\}$ for $n \geq 2$ or $n = \infty$.

(5) Let $L = (01)^*$. $\mathbf{S}(L)$ is not regular but context-free. So the class of regular language is not closed under $\mathbf{S}$. On the other hand, maybe $\mathbf{Q}(L)$ is generally context-sensitive.

PROPOSITION 3.1 If a language L is regular, then $\mathbf{S}(L)$ is context-free.

Sketch of Proof) Let M be a finite automaton $M = (Q, A, \delta, q_0, F)$ without ϵ-move. We can construct a PDA $M' = (Q, A, \bar{A} \cup \{Z_0\}, \delta', q_0, Z_0, F)$, where $\bar{A} = \{\bar{a} \mid a \in A\}$ is a copy of A and the each rule in transition δ' is one of the following forms:

$(q, a, X) \to (p, X)$, if $\delta(q, a) = p$.
$(q, \epsilon, X) \to (p, \bar{a}X)$ if $\delta(q, a) = p$. **(PUSH)**
$(q, a, \bar{a}) \to (q, \epsilon)$ for $\forall q \in Q, \forall a \in A$. **(POP)**
$(q, \epsilon, Z_0) \to (q, \epsilon)$ for $\forall q \in F$. **(POP)**

Then $\mathbf{S}(L) = L(M')$ with empty-stack acceptance. ∎

PROPOSITION 3.2 Let w be a word over an alphabet A with $|A| = m > 1$. If $n \geq 2(m-1) - 1$ or $n = \infty$, $\mathbf{S}^n(w) = \mathbf{Q}^n(w)$ is the set of all permutations of w and then is commutative.

COROLLARY 3.1 Let L be a language over an alphabet A with $|A| = m > 1$. If $n \geq 2(m-1) - 1$ or $n = \infty$,

$$\mathbf{S}^n(L) = \mathbf{Q}^n(L) = \Phi^{-1}\Phi(L)$$

is commutative, where Φ is the Parikh mapping.

PROPOSITION 3.3 The language $L = (012)^+$ is regular. $\mathbf{S}^\infty(L) = \mathbf{Q}^\infty(L) = \Phi^{-1}\Phi(L)$ is not context-free.

Proof) $\Phi^{-1}\Phi(L) \cap 0^*1^*2^* = \{0^i1^i2^i \mid i > 0\}$ is not context-free but context-sensitive. Since $\mathbf{CFL} \cap \mathbf{REG} \subseteq \mathbf{CFL}$, $\mathbf{S}^\infty(L)$ is not context-free.

DEFINITION 3.4 Let m be a positive integer and $\mathbf{N} = \{0, 1, 2, \cdots\}$. Two elements $\mathbf{a} = (a_1, a_2, \cdots, a_m)$ and $\mathbf{b} = (b_1, b_2, \cdots, b_m)$ in $\mathbf{N}^m$ are said to be incomparable if there does not exist $\mathbf{c} \in \mathbf{N}^m$ such that $\mathbf{a} + \mathbf{c} = \mathbf{b}$ or $\mathbf{b} + \mathbf{c} = \mathbf{a}$.

If nonempty subset H of $\mathbf{N}^m$ is said to be incomparable if any distinct two elements of H are incomparable.

Fact An incompareble subset H of $\mathbf{N}^m$ is finite.

PROPOSITION 3.4 A language C over an alphabet A with $|A| = m$ is a commutative extractable code if $C = \Phi^{-1}(H)$ for some incomparable subset $\emptyset \neq H$ of $\mathbf{N}^m$, where Φ^{-1} is the inverse image of Parikh mapping Φ.

THEOREM 3.1 A commutative extractable code is finite,

COROLLARY 3.2 If C is a commutative extractable code over A, There is some finite language $K = \{w_1, w_2, \cdots, w_k\} \subset A^*$ such that $C = \mathbf{S}^\infty(K) = \Phi^{-1}\Phi(K)$

4 Conclusion

We introduce the language operators **S** and **Q**, which make languages commutative. We show that a commutative extractable code is finite. There are the followings questions:

(1) If a language L is regular, then is $\mathbf{Q}(L)$ context-free or not ?

(2) For words u and v in A^* with $\Phi(u) = \Phi(v)$, find the smallest integer n such that $u \in \mathbf{S}^n(v)$. Does this number n became a distance between u and v ? Does the similar question hold for $\mathbf{Q}$?

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