International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 3 (2018) pp. 1742-1747

Learning Commutative Finite Automata from Membership Queries and Equivalence Queries

Anuchit Jitpattanakul*

Department of Mathematics, Faculty of Applied Science,
King Mongkut's University of Technology North Bangkok, Thailand.

**Author for correspondence;*

Abstract

Active learning is well-motivated in many modern problems of grammatical inference, where labeled data may be unavailable in context of learning. In this paper, we theoretically study the problem of learning of commutative deterministic finite automata (CDFA) in the framework of active learning. The theoretical results show that the class of CDFA is idenfiable in the limit with membership queries and equivalence queries.

Keywords: commutative deterministic finite automata; learnability; active learning

INTRODUCTION

In grammatical inference (GI), learning refers to a process of identifying a formal language in terms of its grammars or automata by a learner who is given information of the formal language. One of most attractive topics in grammatical inference is of theoretically studying on an important property of classes of defined automata. This property is called learnability of language family. The study in grammatical inference is mostly based on three learning models i.e., Gold's passive learning model [1], Angluin's active learning model [2], Valiant's approximately learning model [3].

In the first model, the Gold's learning model is viewed as a framework of passive learning. In the process of learning an unknown language, a number of examples will be provided at each time to a learner who is to hypothesize a grammatical representation of the language on the basis of the examples given so far. The process continues repeatly. The success of learning process is considered by using a criterion called identification in the limit [1]. It was developed by adding some constraints of complexity [4]. One of those criteria that are widely used in a learning model called identification in the limit from polynomial time and data introduced by Higuera [5].

However, there are several situations where the learning can actively interact with its environment. The mathematical setting to do this is called active learning, where queries are made to an oracle. In this learning framework, the learner has access to a truthfully oracle which is allowed to answer specific type of queries. Active learning is a paradigm firstly introduced with theoretical motivations but that for a number of reasons can today be considered also as a pragmatic approach [6]. Some of the theoretical reasons is to make use of additional information that can be measured. For practical view, the active learning is an important field of grammatical inference because it is becoming more widely used inp case of problems where labeling the examples in the training data set is unavailable.

The problem of active learning is mainly studied on the class of regular languages. In Angluin's work, a polynomial time query learning algorithm for the class of minimal complete deterministic finite automata (DFA) is given, in which the learner can ask membership queries (MQ) and equivalence queries (EQ). There are though other types of possible queries: subset, superset, disjointness and exhaustive queries [7], structured membership queries [8], etc. The learnability of various grammatical representation of formal languages has been also studied in active learning framework through identification of specifically defined automata [9] such as regular expression [10], and multiplicity tree automata [11]. Learning algorithms from these works have been also experimentally investigated to real-world applications such as DNA sequences analysis [12], music style recognition [13], and speech recognition [14]. The obtained results show that some GI algorithms are an effective and efficient alternative to solve the problems.

Research in formal language theory has been fruitful in the discovery of subclasses of the class of regular languages, e.g., k-testable languages, k-reversible languages, k-acceptable languages[15-17] and strictly k-acceptable languages [18-19]. This is because the class is almost only one class of formal languages that is both efficiently learnable and general enough to represent many nontrivial real-life phenomena.

In this paper, we focus our attention on the problem of learning of commutative deterministic finite automata (CDFA) in the framework of active learning. Two types of queries, *i.e.* membership queries and equivalence queries, will be theoretically investigated on this learning framework. The results show that the class of CDFA is identifiable in the limit with membership queries and equivalence queries.

The remains are organized as follows. Section 2 presents basic definitions and notations. In section 3, we introduced commutative finite automata and proved some properties of the class. In section 4 we investigate learnability of the class of commutative deterministic finite automata in framework of active learning. Finally, section 5 provides the conclusion of this work.

PRELIMINARIES

The basic definitions and notations used throughout this paper are provided in this section.

Formal languages and automata

Let Σ be an *alphabet* that is a finite and nonempty set of letters. The size of Σ is a number of letters, denoted by $|\Sigma|$. A finite sequence of letters from Σ is called a *string*. Given a string w, the length of strings is the total number of letters appearing in w and it is denoted by $|w|$. The string with length zero is called the *null string* denoted by λ. The infinite set of all possible strings over Σ, denoted by Σ^*, is the set of all finite-length strings generated by concatenating zero or more letters of Σ. Here the *Parikh-map*, denoted by π, maps any string over a k-letter alphabet to the k-dimensional integral vector corresponding to the letter counts of each of the k letters in the string. A *language* over Σ denoted by L is any subset of Σ^*. The family of languages over Σ, denoted by $\mathcal{L}$, is called *a class of languages*. A *finite automaton* is a grammatical representation that is typically defined as a 5-tuple $M = (\Sigma, Q, q_0, F, \delta)$,where Σ is a finite alphabet, Q is a finite non-empty set of states, $q_0 \in Q$ is an initial state, $F \subseteq Q$ is a set of final states, and $\delta : Q \times \Sigma \rightarrow Q$ is a state transition function. The state transition function δ can be extended to a mapping $\delta^*: Q \times \Sigma^* \rightarrow Q$ in the following inductive way: (*i*) $\delta^*(q, \lambda) = q$, for each state $q \in Q$, where λ is the null string, and (*ii*) $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$, for each state $q \in Q$, each letter $a \in \Sigma$, and each string $w \in \Sigma^*$. The finite automaton M is *deterministic* if $|\delta(q, a)| \leq 1$ for each $q \in Q$ and for each $a \in \Sigma$. Then M is called *deterministic finite automata* (shortly denoted by DFA).

Theoretically, an automaton plays an important role as a language recognizer. A string w is *recognized* by an automaton $M = (\Sigma, Q, q_0, F, \delta)$ if $\delta^*(q, w) \in F$. The language recognized by M, denoted by $\mathbb{L}(M)$, is the set of all strings which are recognized by the automaton M and this set is called a *regular language*. A language L is *recognizable* if there exists an automaton M such that $L = \mathbb{L}(M)$.

Active learning and convergence criterions

In grammatical inference, a *learning algorithm* $\mathcal{A}_L$ is a mapping function defined as $\mathcal{A}_L : \mathcal{S} \rightarrow \mathcal{G}$, where $\mathcal{S}$ is a set of all presentation and this set is used for learning any language L in a language class $\mathcal{L}$ by identifying a grammatical representation G in a class $\mathcal{G}$ of corresponding grammatical representations. For learning in GI, an unknown language L, we say that the algorithm $\mathcal{A}_L$ *converges* to $G \in \mathcal{G}$ from $S \in \mathcal{S}$ if and only if $\mathbb{L}(\mathcal{A}_L(S)) = L$.

The active learning is based on the existence of an oracle which can be seen in principle as a device that knows the language and has to answer correctly and can only answer queries from a given set of queries. In this paper, we study on learnability of the class of deterministic commutative finite automata by using two different types of queries. Firstly, only membership queries are available in learning context. Secondly, equivalence queries are additionally given. The two different type of queries are defined as below.

Definition 2.1

A *membership query* (MQ) is made by proposing a string to the oracle, who answers *Yes* if the string belong to the language and *NO* if not. We will denote this formally by

MQ : $\Sigma^* \rightarrow \{Yes, No\}$.

Definition 2.2

An *equivalence query* (EQ) is made by proposing a grammatical representation G to the oracle. The oracle answers *Yes* if the grammatical representation G is equivalent to the target and *NO* if not. We will denote this formally by

EQ : $\Gamma \rightarrow \{Yes, No\}$.

In the active learning process, we define a class of grammatical representation G and the sort of queries that we are allowed to make and the oracle will have to answer. We call this class of queries QUER. Typically if the learner is only allowed to make membership queries, we will have QUER = {MQ}. However, if the learner is allowed to make membership queries and equivalence queries, we will have QUER = {MQ, EQ}.

The learning criterion that will be used in this paper can be found in [1] and if is well known as *identification in the limit with queries*. A formal definition of this criterion is restated as follows.

Definition 2.3

A class Γ is *identifiable in the limit with queries* from QUER if there exists an algorithm $\mathcal{A}$ such that given any grammatical representation G in Γ, $\mathcal{A}$ identifies Γ in the limit, i.e. returns a grammatical representation G' equivalent to G and halts.

COMMUTATIVE DETERMINISTIC FINITE AUTOMATA AND THEIR PROPERTIES

Given two words u and v we say that commutatively equivalent if $u = a_1 a_2 \ldots a_n$ with $a_i \in \Sigma$ for $1 \leq i \leq n$, and there exist a permutation σ on $\{1, 2, \ldots, n\}$ such that $a_{\sigma(1)} a_{\sigma(2)} \ldots a_{\sigma(n)} = v$. We denote it by $u \sim_{com} v$. For instance, $abca \sim_{com} cbaa$. Given an alphabet Σ, a language L is commutative if and only if it is the union of some $\sim_{com}$ class.

Definition 3.1

A *commutative deterministic finite automaton* (CDFA) is a 5-tuple $M = (\Sigma, Q, q_0, F, \delta)$ where

- $\Sigma = \{a_1, a_2, \ldots, a_n\}$ is a finite alphabet,
- $Q = Q_{a1} \times Q_{a2} \times \ldots \times Q_{an}$ is a finite set of states,
- $q_0 \in Q$ is the initial state,
- $F \subseteq Q$ is a set of accepting states, and
- $\delta((q_1, \ldots q_i, \ldots, q_n), a_i) = (q_1, \ldots \delta(q_i, a_i), \ldots, q_n)$ where δ_i is a function from Q_{ai} onto Q_{ai} for $1 \leq i \leq n$.

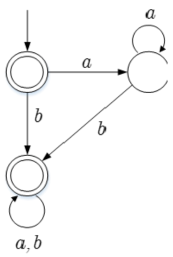
We let **CDFA**(k) denote *the class of k-letter CDFAs* and let **CDFA** = $\cup_{k \in \mathbb{N}}$ **CDFA**(k). We let **CDFA**$(k)_s$ denote *the subclass of* **CDFA**(k) *of size at most s.*

Definition 3.2

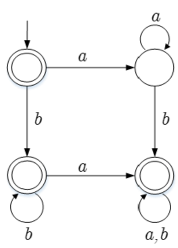
A *minimal commutative finite automaton* (MCFA) is a 5-tuple $M = (\Sigma, Q, q_0, F, \delta)$ where

- $\Sigma = \{a_1, a_2, \ldots, a_n\}$ is a finite alphabet,
- $Q = Q_{a1} \times Q_{a2} \times \ldots \times Q_{an}$ such that $Q_{ai} = \cup_{m \leq 0}[a_i^m]_{\sim L}$ for $1 \leq i \leq n$,
- $q_0 = ([\lambda]_{\sim L}, [\lambda]_{\sim L}, \ldots, [\lambda]_{\sim L})$,
- $F = \{\pi_{a1}(x), \pi_{a2}(x), \ldots, \pi_{an}(x), \mid x \in L\}$, and
- $\delta_{ai}([a_i^m]_{\sim L}, a_i) = [a_i^{m+1}]_{\sim L}$ for $1 \leq i \leq n$.

Example 3.1 Let $L = \{w \in \Sigma^* : |w|_a = 0 \text{ or } |w|_b > 0\}$ be a formal language defined over 2-letter alphabet $\Sigma = \{a, b\}$. The minimal DFA of L is depicted in Fig. 1(a) and the minimal CDFA of the language L is depicted in Fig. 1(b).



(a) A minimal DFA of L



(b) A minimal CDFA of L

Figure 1. The minimal automata recognizing

$L = \{w \in \Sigma^* : |w|_a = 0 \text{ or } |w|_b > 0\}$

Definition 3.3

A language recognized by CDFA $M = (\Sigma, Q, q_0, F, \delta)$ is called a *commutative regular language* (CRL) defined as $L = \{w : \delta(q_0, w) \in F\}$. A set of all commutative regular languages is called that a *class of commutative regular languages* denoted by **CRL**.

To study learnability of **CRL**, some theoretical properties are needed. Then, we have proved some propositions and lemmas that will be referred in next section.

Definition 3.4

A CDFA is called *a single final state CDFA* if it contains exactly one final state. We let **sfs-CDFA**(k) denote *the class of k-letter CDFA with a single final state*, and **sfs-CDFA**$(k)_s$ denote denote *the subclass of* **sfs-CDFA**(k) *of size at most s.*

We note the following simple fact about the single final state 1-letter commutative deterministic finite automata.

Lemma 3.1 $\forall s \in \mathrm{N}$, card(**sfs-CDFA**$(1)_s$) = s^2.

Proof: Every 1-letter CDFA is in a very restrict form. It starts out with a start state, and then follows a sequence of final and non-final states until it finally comes to one of the previous states. Since there are s states, we know that the sequence of states comes to a previous state at the s^{th} step. Thus the number of sfs-CDFA(1) is s^2. ■

Definition 3.5

Tuples of 1-letter single finite state CDFA, denoted by TCDFA, is the class of finite tuples of sfs-CDFA such that

(i) **TCDFA**(k) = $\{\langle M_1, M_2, \ldots, M_k\rangle\}$: $\forall i \leq k$, $M_i \in$ **sfs-CDFA**(k)},

(ii) **TCDFA** = $\cup_{k \in \mathrm{N}}$ **TCDFA**(k).

(iii) The *language represented by a TCDFA*

$T = (M_1, M_2, \ldots, M_k)$

is defined as follows:

$L(T) = \pi^{-1}(\pi(L(M_1)) \times \pi(L(M_2)) \times \ldots \times \pi(L(M_1)))$.

Definition 3.6

Sets of tuples of CDFA, denoted by STCDFA is the class of finite sets of TCDFA(k) for some k such that

(i) **STCDFA**(k) = $\{\{T_1, T_2, \ldots, T_j\}\} : \forall i \leq j, T_i \in$ **TCDFA**(k)},

(ii) **STCDFA** = $\cup_{k \in □}$ **STCDFA**(k).

(iii) The *language represented by a STCDFA*

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 3 (2018) pp. 1742-1747
© Research India Publications. http://www.ripublication.com

$S = \{T_1, T_2, \ldots, T_j\}$

is defined as follows:

$L(S) = L(T_1) \cup L(T_2) \cup \ldots \cup L(T_j)$.

In this section, we prove that any language accepted by a CDFA is a union of at most (s^k) commutative regular languages that are k-dimensional Cartesian products of 1-letter commutative regular languages, each of which can be represented by a 1-letter CDFA with a single finite state with at most s states.

We present our study with the case $k = 2$ for simplicity. The study extends straightforwardly to the general case. We define the projection M_a and M_b of M onto $\{a\}^*$ and onto $\{b\}^*$, respectively, as follows.

Definition 3.7

Let $M = (\Sigma, Q, q_0, F, \delta)$ be any 2-letter CDFA. The *projection of M onto {a}** is defined as $M_a = (\Sigma, Q_a, q_0, Q_a \cap F, \delta[Q_a \times \{a\}^*])$ where Q_a is the subset of Q which is reachable by some a^x, and $\delta[Q_a \times \{a\}^*]$ is the restriction of δ to $Q_a \times \{a\}^*$. The *projection of M onto {b}** denoted by M_b is similarly defined.

For each state q in Q_a, we pick a representative for a member a^x of $\{a\}^*$ such that $M(a^x) = q$, and similarly for Q_b.

Definition 3.8

Let $Q_a = \{q_{a,1}, q_{a,2}, \ldots, q_{a,l}\}$. We fix $R_a = \{a^{x1}, a^{x2}, \ldots, a^{xl}\} \subseteq \{a\}^*$, where for each $i \leq l$, a^{xi} is a representative for $q_{a,i}$. Similarly, we let $Q_b = \{q_{b,1}, q_{b,2}, \ldots, q_{b,m}\}$. We fix $R_b = \{b^{y1}, b^{y2}, \ldots, b^{ym}\} \subseteq \{b\}^*$, where for each $j \leq m$, b^{yj} is a representative for $q_{b,j}$.

Definition 3.9

For an arbitrary string $a^x \in \{a\}^*$, we let $[a^x]_{Ma}$ denote the right invariance equivalence class of M_a which contain a^x. Also we let $[x]_{Ma}$ denote $\{y : a^y \in [a^x]_{Ma}\}$. We similarly define $[b^y]_{Mb}$ and $[y]_{Mb}$ for each $y \in$ N.

Lemma 3.2 Let $M = (\Sigma, Q, q_0, F, \delta)$ be any minimal 2-letter CDFA, as given before. Let q be an arbitrary state in Q. Then the set of strings leading to q, $L(q)$, is the following language: $L(q) = \pi^{-1}(L_q)$ such that the inverse Parikh-image of $L_q \subseteq$ N where:

$L_q = \cup\{[x_i]_{Ma} \times [y_i]_{Mb} : a^{xi} \in R_a \text{ and } b^{yj} \in R_b \text{ and } M(a^{xi}b^{yj}) = q\}$,

Hence, the language accepted by M is nothing but □$(M) = \pi^{-1}(L_q)$ such that the inverse Parikh-image of $L_F \subseteq$ □^{2} where:

$L_F = \cup\{L_q : q \in F\}$.

Proof: Let q be an arbitrary state of M. Let $a^k \in \{a\}^*$ and $b^l \in \{b\}^*$ be such that $M(a^k b^l) = q$. Now, we assume that $a^x \in [a^k]_{Ma}$ and $b^y \in [b^l]_{Mb}$ Then, we can show that $M(a^x b^y) = q$ also, using the commutativity of M as follows.

$M(a^x b^y) = \delta(q_0, a^x b^y)$

$= \delta(M_a(a^x), b^y)$ by definition of M_a

$= \delta(M_a(a^k), b^y)$ by $a^x \in [a^k]_{Ma}$

$= M(a^k b^y)$ by definition of M_a

$= M(b^y a^k)$ by commutativity of M

$= \delta(M_b(b^y), a^k)$ by definition of M_b

$= \delta(M_b(b^l), a^k)$ by $b^y \in [b^l]_{Mb}$

$= M(b^l a^k)$ by definition of M_b

$= M(a^k b^l)$ by commutativity of M

$= q$

From this it follows immediately that if for any two representatives a^{xi} and b^{yi}, it is that case that $M(a^{xi} b^{yj}) = q$, then $\pi^{-1}([x_i]_{Ma} \times [y_j]_{Mb}) \subseteq L(q)$. It also follows that for any two representatives a^{xi} and b^{yi}, if it is not the case that $M(a^{xi} b^{yj}) = q$, then for no $a^x \in [a^{xi}]_{Ma}$ and $b^y \in [b^{yj}]_{Mb}$ is it the case that $M(a^{xi} b^{yj}) = q$, and hence $\pi^{-1}([x_i]_{Ma} \times [y_j]_{Mb}) \cap L(q) = \varnothing$. Hence $L(q)$ exactly equals the union of $\pi^{-1}([x_i]_{Ma} \times [y_j]_{Mb})$ for all pairs x_i, y_j such that $M(a^{xi} b^{yj}) = q$. ■

Example 3.2 Let $L = \{w \in \Sigma^* : |w|_a = 6 \text{ or } |w|_b = 2\}$ be a formal language over 2-letter alphabet $\Sigma = \{a, b\}$. The 2-letter CDFA M recognizing L is depicted in Fig.2. From lemma 3.2, the projection M_a of M onto $\{a\}^*$ and M_b of M onto $\{b\}^*$, shown in Fig.3(a) and Fig.3(b).

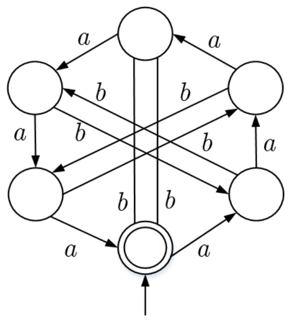
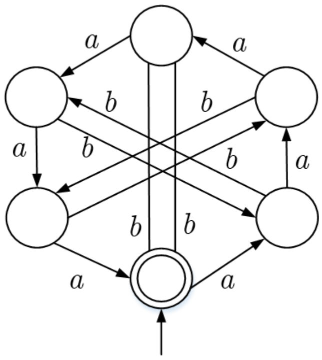
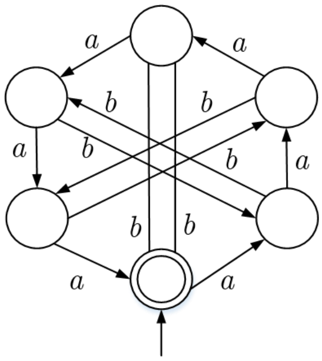


Figure 2. The 2-letter CDFA of

$L = \{w \in \Sigma^* : |w|_a = 6 \text{ or } |w|_b = 2\}$



(a) the projection M_a of M onto $\{a\}^*$



(b) the projection M_b of M onto $\{b\}^*$

Figure 3. Two 1-letter projection of CDFA

Lemma 3.3 For each CDFA M of size s over a k-letter alphabet, there exists $S \in$ **STCDFA**(k) such that

1. $\Lambda(S) = \Lambda(M)$,
2. $card(S) \leq s^k$,
3. $\forall T \in S$, if $T = \langle M_1, M_2, \ldots, M_k \rangle$ then $\forall i \leq k$ size$(M_i) \leq s$.

Proof: Let $M \in$ **CDFA**(k) be given, with s being the number of states in M, i.e., s = size(M). We can assume without loss of generality that M is minimal, since otherwise M can be replaced by a language equivalent, minimal CDFA with the less state. Let M_a^{xi} denote the 1-letter single final state automaton obtained by modifying Ma so as to make the state to which the string a^{xi} leads the unique final state in M_a^{xi}. Define M_b^{yj} analogously. Then, the statement of Lemma 3.2 can be rewritten as follows.

$$\Lambda(M) = \cup(\{ \Lambda(\langle M_a^{xi}, M_b^{yj} \rangle) : M(a^{xi}b^{yj}) \in F\})$$

If we let $S = \{ \langle M_a^{xi}, M_b^{yj} \rangle : M(a^{xi}b^{yj}) \in F\}$, then the S witnesses the claim in the statement of Lemma 3.2: For a given M, M_a, and Mb are constant and thus card(S) = card($\{\langle M_a^{xi}, M_b^{yj} \rangle : x_i, y_j \in \mathrm{N}\}$) = s^2, and each TCDFA in S is clearly in the required form, since size $(M_a^{xi}) \leq s$ and $(M_b^{yj}) \leq s$. ■

LEARNING COMMUTATIVE FINITE AUTOMATA IN FRAMEWORK OF ACTIVE LEARNING

In this section, we theoretically show that the class of commutative deterministic finite automata is identifiable in the limit by using both membership queries and equivalence queries. A reduction technique will be used in this proof.

The reduction technique for grammatical inference have been firstly formalized in [17] by Higuera. This algebraic technique allows us to refine previous theoretical results of learnability.

Theorem 4.1 (from Theorem 2 in [17])

If the B of languages is learnable in terms of R(B) from Pres(B), and there exists a computable function $\chi: R(B) \rightarrow R(A)$ *such that* $\psi \circ \chi = Id$, *and* ξ *is a computable reduction, then the class A of languages is learnable in term of R(A) from Pres(A).*

Proof: see in [17].

To get better understanding, a diagram representing the situation is shown in Fig. 2.

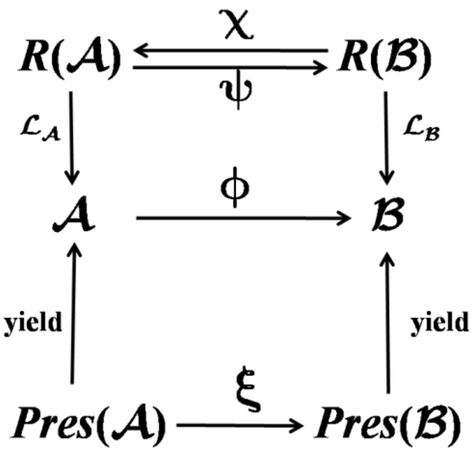


Figure 2. The commutation diagram

In [20]. the class of deterministic finite automata (DFA) of regular languages (REG) have been proved that it is identifiable in the limit with queries by using both membership queries and equivalence queries. For this work, we refine the theoretical results.

Theorem 4.2 *The class of commutative deterministic finite automata is identifiable in the limit with queries by using both membership queries and equivalence queries.*

Proof: Let A_{REG} be a learning algorithm that identifies languages in the class of regular languages (REG). Consider algorithm A_{CDFA} below, that learns a set $QUER_{CDFA}$ of membership queries (MQ) and equivalence queries (EQ) and then output a CDFA corresponding the learning set. The algorithm A_{CDFA} executes as follows.

Algorithm : A_{CDFA}

Input : $QUER_{CDFA} = \{MQ, EQ\}$

Output : CDFA

1: $QUER_{REG} \leftarrow \xi(QUER_{CDFA})$;

2: DFA$\leftarrow A_{REG}(QUER_{REG})$

3: CDFA$\leftarrow \chi$(DFA)

Return CDFA

Since ξ is identity, and χ is the natural transformation as we have proved in lemma 3.3. Hence the commutative deterministic finite automata is identifiable in the limit with queries by using both membership and equivalence queries. ■

CONCLUSION

In this work we study learnability of the commutative deterministic finite automata on active learning model with two different types of queries. The reduction technique for grammatical inference have been used in this study. We have proved that the class of commutative finite automata is identifiable in the limit with queries by using both membership queries and equivalence queries.

ACKNOWLEDGMENT

This research was funded by King Mongkut's University of Technology North Bangkok. Contract no. KMUTNB-GEN-58-19.

REFERENCES

[1] Gold E.M., 1967, "Language identification in the limit," Information and Control, vol. 10, no. 5,pp. 447–474.

[2] Angluin D., 1987, "Queries and concept learning," Machine Learning Journal, 2, pp. 319–342.

[3] Valiant L. G., 1984, "A theory of the learnable," Comm. ACM , vol. 27, no. 11, pp. 1134–1142.

[4] Pitt L., 1989, "Inductive inference, DFA's, and computational complexity," Proc. of the Int. Workshop AII'89, pp. 18–44.

[5] de la Higuera C., 1997, "Characteristic sets for polynomial grammatical inference," Machine Learning Journal, 27, pp. 125–138.

[6] Angluin, D., 1987, "Learning regular sets from queries and counterexamples," Information and Computation 75 (2),pp. 87–106.

[7] Angluin, D., 1988, "Queries and concept learning," Machine Learning 2 (4), pp. 319–342.

[8] Sakakibara, Y., 1990, "Learning context-free grammars from structural data in polynomial time," Theoretical Computer Science 76, pp. 223–242.

[9] de la Higuera C., 2005, "A bibliographical study of grammatical inference," Pattern Recognition, 38, pp. 1332–1348.

[10] Kinber, E.B., "On learning regular expressions and patterns via membership and correction queries," [33] 125–138

[11] Amaury, H., 2006, "Learning multiplicity tree automata," Proc. of ICGI'06, pp.268-280.

[12] Sakakibara Y., 2005, "Grammatical Inference in Bioinformatics," IEEE Trans. Pattern Anal. Mach. Intell., 27, no. 7, pp. 1051–106.

[13] Cruz P. and Vidal E., 1998, "Learning regular grammars to model musical style: Comparing different coding scheme," Proc. of ICGI'98, pp. 211-222.

[14] García P. et al., 1994, "On the use of the morphic generator grammatical inference (mggi) methodology in automatic speech recognition," Int. J. Pattern Recogn. Artif. Intell., vol. 4, pp. 667–685.

[15] Jitpattanakul A. and Surarerks A., 2012, "The study of learnability of the class of *k*-acceptable languages on Gold's learning model," Chiang Mai Journal of Science.

[16] Jitpattanakul A. and Surarerks A., 2010, "Characteristic sets for learning k-acceptable languages," in Proceeding of ECTI-CON 2010 - The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, pp. 483-487.

[17] Jitpattanakul A., 2017, "Decidability properties of the class of formal languages recognized by k-edge finite state automata," Journal of Theoretical and Applied Information Technology, vol. 95, no. 24, pp. 6964-6973.

[18] Jitpattanakul A., 2016, "Closure properties of languages recognized by strictly k-edge finite state automata," Far East Journal of Mathematical Sciences, vol. 100, no. 3, pp. 397-411.

[19] Jitpattanakul A., 2012, "Learnability of the class of strictly k-acceptable languages," Far East Journal of Mathematical Sciences, vol. 71, no. 1, pp. 169-184.

[20] de la Higuera C., 2010 "Grammatical inference: learning automata and grammars" Cambridge University Press.