PAPER

Polynomial Time Learnability of Simple Deterministic Languages from MAT and a Representative Sample

Yasuhiro TAJIMA[†], ***Student Member***, **Etsuji TOMITA**[†], ***and*** **Mitsuo WAKATSUKI**[†], ***Members***

SUMMARY We propose a learning algorithm for simple deterministic languages from queries and a priori knowledge. To the learner, a special finite subset of the target language, called a representative sample, is provided at the beginning and two types of queries, equivalence queries and membership queries, are available. This learning algorithm constructs nonterminals of a hypothesis grammar based on Ishizaka(1990)'s idea. In Ishizaka(1990)'s algorithm, the learner makes rules as many as possible from positive counterexamples, and diagnoses wrong rules from negative counterexamples. In contrast, our algorithm guesses a simple deterministic grammar and diagnoses them using positive and negative counterexamples based on Angluin(1987)'s algorithm.

key words: *learning via queries, simple deterministic languages, representative sample, MAT learning*

1. Introduction

Learning via queries makes some classes of languages be polynomial time learnable even if polynomial time learning of them in the limit is too hard. Angluin [2] showed the polynomial time learnability of regular languages from membership queries and equivalence queries. The teacher which can answer these two queries is called a "Minimally Adequate Teacher (MAT)." The polynomial time learnability of some classes of languages which contain regular languages has been shown under the same or similar settings. Ishizaka [4] showed that simple deterministic languages are polynomial time learnable with membership queries and extended equivalence queries. The extended equivalence query takes a context-free grammar as an input while the target language must be only simple deterministic. Therefore a learning algorithm which uses standard equivalence queries is expected.

There is another approach of query learning in polynomial time. Angluin [1] also showed the polynomial time learnability of regular languages from membership queries and a representative sample. Here, for a deterministic finite automaton M, a representative sample Q is a finite subset of the target language $L(M)$ such that all transitions of M are traced to accept all words in Q. This set is given to the learner as initial information and it is available during the learning process.

We present a polynomial time learning algorithm for a simple deterministic language from MAT and a representative sample. Here, a representative sample of a simple deterministic language is defined as a natural extension of the above one. The time complexity of this learning algorithm is bounded by a polynomial in the followings. 1. The length of the longest word in the representative sample. 2. The length of the longest word in counterexamples. 3. The cardinality of the representative sample. 4. The size of a grammar (cardinalities of terminals and nonterminals) which generates the target language.

Our learning algorithm is based on Ishizaka [4] and Angluin [2]. The construction method of nonterminals is the same as in [4]. The algorithm for making rules and merging nonterminals is similar to [2]. Then the learner diagnoses the wrong hypothesis from positive and negative counterexamples.

2. Definitions

2.1 Basic Definitions

Basic definitions are based on [3]. A *context-free grammar* is a 4-tuple $G = (N, \Sigma, P, S)$ where N is a finite set of *nonterminals*, Σ is a finite set of *terminals*, P is a finite set of *rules* and $S \in N$ is the *start symbol*. If P is ε-*free* and any rule in P is of the form $A \to a\beta$ then $G = (N, \Sigma, P, S)$ is said to be in *Greibach normal form*, where $A \in N, a \in \Sigma$ and $\beta \in N^*$. A grammar G is *simple deterministic* [5] iff G is in Greibach normal form and for every $A \in N$ and $a \in \Sigma$, if $A \to a\beta$ is in P then $A \to a\gamma$ is not in P for any $\gamma \in N^*$ such that $\gamma \neq \beta$.

Let a rule $A \to a\beta$ be in P where $\beta \in N^*$, $A \in N$ and $a \in \Sigma$. Let γ and $\gamma' \in (N \cup \Sigma)^*$. Then $\gamma A\gamma' \underset{G}{\Rightarrow} \gamma a\beta\gamma'$ denotes the *derivation* from $\gamma A\gamma'$ to $\gamma a\beta\gamma'$ in G. We define $\overset{*}{\underset{G}{\Rightarrow}}$ to be the reflexive and transitive closure of $\underset{G}{\Rightarrow}$. When it is not necessary to specify the grammar G, $\alpha \Rightarrow \alpha'$ and $\alpha \overset{*}{\Rightarrow} \beta$ denote $\alpha \underset{G}{\Rightarrow} \alpha'$ and $\alpha \overset{*}{\underset{G}{\Rightarrow}} \beta$, respectively. A *word* generated by G is $w \in \Sigma^*$ such that $S \overset{*}{\underset{G}{\Rightarrow}} w$, where S is the start symbol. We define the *language* generated from γ by

Manuscript received May 6, 1999.
Manuscript revised September 17, 1999.

[†]The authors are with the Graduate School of Electro-Communications, The University of Electro-Communications, Chofu-shi, 182–8585 Japan.

G as $L_G(\gamma) = \{w \in \Sigma^* \mid \gamma \overset{*}{\underset{G}{\Rightarrow}} w\}$, where $\gamma \in (N \cup \Sigma)^*$. In addition, $L(G)$ denotes the language generated by G which is defined as $L(G) = L_G(S)$ for the start symbol S. The language generated by a simple deterministic grammar is called the *simple deterministic language*.

In this paper, $|\beta|$ denotes the length of β if β is a string and $|W|$ denotes the cardinality of W if W is a set. For any simple deterministic grammar $G_1 = (N_1, \Sigma, P_1, S_1)$, there exists a simple deterministic grammar $G_2 = (N_2, \Sigma, P_2, S_2)$ such that $L(G_1) = L(G_2)$ and every rule $A \to a\beta$ in P_2 satisfies that $|\beta| \leq 2$. Such a grammar G_2 is said to be in *2-standard form*. For a simple deterministic grammar $G = (N, \Sigma, P, S)$ and $A \in N$, if there exists a derivation such that $S \overset{*}{\Rightarrow} xAz \overset{*}{\Rightarrow} xyz$ for some $x, z \in \Sigma^*$ and $y \in \Sigma^+$ then A is called *reachable and live*. Throughout this paper, we assume that a simple deterministic grammar is in 2-standard form and all nonterminals are reachable and live.

Assume that both $G_1 = (N_1, \Sigma, P_1, S_1)$ and $G_2 = (N_2, \Sigma, P_2, S_2)$ are simple deterministic grammars. G_1 is *homomorphic* to G_2 if there exists a bijection f such that

- $f(S_1) = S_2$ and
- $A \to a\beta$ is in P_1 iff $f(A) \to a \cdot f(B_1) \cdot f(B_2) \cdots f(B_i)$ is in P_2 for $A \in N_1, a \in \Sigma$ and $\beta = B_1 B_2 \cdots B_i \in N_1^*$ where $B_1, B_2, \cdots, B_i \in N_1$.

For $w \in \Sigma^+$, $pre(w) = \{w' \in \Sigma^* \mid w'w'' = w, w'' \in \Sigma^*\}$ is called the set of *prefixes* of w. $proper\_pre(w) = \{w' \in \Sigma^* \mid w'w'' = w, w'' \in \Sigma^+\}$ is called the set of *proper prefixes* of w. Similarly, $suf(w) = \{w'' \in \Sigma^* \mid w'w'' = w, w' \in \Sigma^*\}$ and $proper\_suf(w) = \{w'' \in \Sigma^* \mid w'w'' = w, w' \in \Sigma^+\}$ are called the set of *suffixes* of w and the set of *proper suffixes* of w, respectively.

For a set R, R^2 denotes the set of concatenations of two elements in R, i.e. $R^2 = \{r_1 \cdot r_2 \mid r_1, r_2 \in R\}$.

2.2 A Representative Sample

Definition 1: Let $G = (N, \Sigma, P, S)$ be a 2-standard form simple deterministic grammar and Q be a finite subset of $L(G)$. Then Q is a representative sample of G iff the following holds.

- For any rule $A \to a\beta$ in P, there exists a word $w \in Q$ such that $S \overset{*}{\Rightarrow} xA\gamma \Rightarrow xa\beta\gamma \overset{*}{\Rightarrow} w$ for some $x \in \Sigma^*$ and $\gamma \in N^*$. □

From this definition, for any simple deterministic grammar $G = (N, \Sigma, P, S)$, there exists a representative sample Q such that $|Q| \leq |P|$.

Definition 2: For a simple deterministic language L, a finite set $Q \subseteq L$ is a representative sample iff there exists a 2-standard form simple deterministic grammar $G = (N, \Sigma, P, S)$ such that $L(G) = L$ and Q is a representative sample of G. □

This is a natural extension of the definition of a representative sample for a regular language by Angluin [1].

Example 3: Let $G = (N, \Sigma, P, S)$ be a simple deterministic grammar such that $N = \{S, A, B\}, \Sigma = \{a, b, c\}$ and $P = \{S \to aSA,\ S \to bSB,\ S \to c,\ A \to a,\ B \to b\}$. Then $\{abcba\}$ is a representative sample of the grammar G, because all rules in P are applied in the derivation for $abcba \in \{abcba\}$ such that $S \Rightarrow aSA \Rightarrow abSBA \Rightarrow abcBA \Rightarrow abcbA \Rightarrow abcba$.

Also $\{aca, bcb\}$ is a representative sample of G. Consider two derivations such that $S \Rightarrow aSA \Rightarrow acA \Rightarrow aca$ and $S \Rightarrow bSB \Rightarrow bcB \Rightarrow bcb$, then the former uses rules $S \to aSA$, $S \to c$ and $A \to a$, and the latter uses rules $S \to bSB$, $S \to c$ and $B \to b$. Thus all rules in P are applied to generate $\{aca, bcb\}$.

On the other hand, each of $\{aca, aacaa\}$, $\{abcba, aa\}$ and $\{aaabbb, abb\}$ is not a representative sample of G. Because, for $\{aca, aacaa\}$, it is generated by $S \to aSA$, $S \to c$ and $A \to a$. For $\{abcba, aa\}$ and $\{aaabbb, abb\}$, neither of them is a subset of the target language. Thus each of them is not a representative sample. □

3. Settings of Queries

We define two types of queries as follows.

(1) $MEMBER(w)$

A membership query for a simple deterministic language L_t is defined as follows.

Input : $w \in \Sigma^*$.
Output : "*yes*" if $w \in L_t$ and "*no*" if $w \notin L_t$.

(2) $EQUIV(G')$

An equivalence query for a simple deterministic language L_t is defined as follows.

Input : a simple deterministic grammar G'.
Output : "*yes*" $\cdots$ if $L(G') = L_t$.
"*no*" and a *counterexample* $u \in \Sigma^*$ such that $u \in (L_t - L(G')) \cup (L(G') - L_t)$ $\cdots$ if $L(G') \neq L_t$.

The counterexample u is called positive if $u \in L_t$ and negative otherwise.

Here, an equivalence problem for simple deterministic grammars is solvable [5]. Thus these oracles are both computable.

Throughout this paper, we denote the target language by L_t and a representative sample given to the learner by Q. In addition, we suppose that $G_t = (N_t, \Sigma, P_t, S_t)$ is a simple deterministic grammar such that $L_t = L(G_t)$ and Q is a representative sample of G_t.

4. The Main Result

Now, we show the main theorem.

Theorem 4: Simple deterministic languages are polynomial time learnable from MAT and a representative sample. Here, the polynomial consists of $|Q|$, $|N_t|$, $|\Sigma|$, $\max\{|w| \mid w \in Q\}$ and the length of the longest counterexample replied by $EQUIV()$. □

For some simple deterministic language L_t, there exists a grammar G_t and a representative sample Q such that $L_t = L(G_t)$ and the length of the longest word in Q can not be bounded by a polynomial of the size of G_t. For example, suppose that $G_t = (N_t, \Sigma, P_t, S_t) = (\{A_0, A_1, \cdots, A_n\}, \{a\}, P_t, A_0)$ is a simple deterministic grammar for some given integer $n > 0$ where

$$P_t = \{A_0 \to aA_1A_1, A_1 \to aA_2A_2, \cdots, A_{n-1} \to aA_nA_n, A_n \to a\}.$$

Then only $\{a^{2^{n+1}-1}\}$ is the representative sample and the length of the word in it is not bounded by a polynomial of $|N_t|$. On the other hand, G_t is equivalent to $G'_t = (N'_t, \Sigma, P'_t, S'_t) = (\{B_0, B_1, \cdots, B_{2^{n+1}-2}\}, \{a\}, P'_t, B_0)$ such that

$$P'_t = \{B_0 \to aB_1, B_1 \to aB_2, \cdots, B_{2^{n+1}-3} \to aB_{2^{n+1}-2}, B_{2^{n+1}-2} \to a\}.$$

Now, the length of the longest word in $Q = \{a^{2^{n+1}-1}\}$ is bounded by a polynomial of $|N'_t|$.

In addition, it is independent of the learner that which grammar is applied to construct Q. Thus it is reasonable to regard the length of the longest word in Q as an independent parameter.

5. The Learning Algorithm

In the followings of this paper, we assume that $G_h = (N_h, \Sigma, P_h, S_h)$ denotes a hypothesis grammar which is guessed by the learner.

At first we show the whole learning algorithm in Fig. 1. We describe its details in the followings.

(step 1) makes nonterminals from Q

Nonterminals of a hypothesis grammar are made from 3-tuples of words in Q. For every word $w \in Q$ (which is a positive example), the learner makes 3-tuples of subwords in w such that $(w_p, w_m, w_s) \in \Sigma^+ \times \Sigma^+ \times \Sigma^*$ and $w_pw_mw_s = w$. The set of these 3-tuples is denoted by R. This idea is introduced by Ishizaka [4].

Lemma 5 (Ishizaka [4] Lemma 10): Let $G_t = (N_t, \Sigma, P_t, S_t)$ be a simple deterministic grammar. For any $A(\neq S_t) \in N_t$, there exist $w \in L(G_t)$ and a 3-tuple (w_p, w_m, w_s) such that

- $w_pw_mw_s = w$,

```
Procedure learning(INPUT, OUTPUT);
INPUT :  Q (a representative sample of L_t);
OUTPUT : G_h (the correct hypothesis grammar);
begin
  (step 1)
  R := {(x,y,z) | z ∈ Σ*, x,y ∈ Σ+, x·y·z ∈ Q}
    ∪{(ε,w,ε) | w ∈ Q};
  W := {y ∈ Σ+ | x,z ∈ Σ*, and x·y·z ∈ Q};
  LOG := {};  /* words asked by the learner */
  repeat
  (step 2)
    find T(r,w) for all r ∈ R and w ∈ W;
    LOG := LOG ∪ {u | u = p_r · w · short(r), where
          r = (p_r, m_r, s_r) ∈ R, w ∈ W};
    make N_h and define S_h;
  (step 3)
    find handle(r) for all r ∈ R;
    find body(r) for all r ∈ R;
    find prt(body(r)) for all r ∈ R;
  (step 4)
    find P(A,a) for all A ∈ N_h and a ∈ Σ;
    make the hypothesis grammar G_h = (N_h, Σ, P_h, S_h);
  (step 5)
    /* if L(G_h) is not consistent with the results of
       membership queries */
    if (∃v ∈ LOG, v ∈ (L_t − L(G_h)) ∪ (L(G_h) − L_t)) then
      W := W ∪ {y ∈ Σ+ | x,z ∈ Σ*, and x·y·z = v};
    else
      if (EQUIV(G_h) = yes) then
        output G_h and terminate;
      else
        for a counterexample w ,
        W := W ∪ {y ∈ Σ+ | x,z ∈ Σ*, and x·y·z = w};
      end-if
    end-if
  until (forever)
end.
```

Fig. 1 The learning algorithm.

- $S_t \stackrel{*}{\Rightarrow} w_p \cdot A \cdot w_s \stackrel{*}{\Rightarrow} w$, and
- it holds that

$$MEMBER(w_p \cdot u \cdot w'_s) = yes \iff u \in L_{G_t}(A)$$

for all $u \in \Sigma^+$, where w'_s is the shortest suffix of w_s such that $w_pw_mw'_s \in L(G_t)$. □

Obviously, the above lemma holds where $w_p = w_s = \varepsilon$ and $A = S_t$. Thus the learner adds such 3-tuples to R, i.e.

$$R := \{(x,y,z) \mid z \in \Sigma^*, x,y \in \Sigma^+, xyz \in Q\} \cup \{(\varepsilon, w, \varepsilon) \mid w \in Q\}.$$

From the definition of the representative sample, for any nonterminal $A \in N_t$, there exists a 3-tuple in R which satisfies the above lemma.

(step 2) merges nonterminals

Before making rules for a hypothesis grammar, the learner merges some members in R. For $r = (p, m, s) \in R$, we define $short(r)$ as the shortest suffix of s such that $p \cdot m \cdot short(r) \in L_t$. Let both 3-tuples $r_1 =$

(p_1, m_1, s_1) and $r_2 = (p_2, m_2, s_2)$ be in R. The learner assumes r_1 and r_2 are equivalent if

$$MEMBER(p_1 \cdot u \cdot short(r_1)) = MEMBER(p_2 \cdot u \cdot short(r_2))$$

for all $u \in W$. Here, $W \subset \Sigma^+$ is a part of an "observation table." In other words, N_h is the equivalence class on R.

This merger means that two elements in R should represent the equivalent nonterminal if their behavior on W is the same. When $W = \{\}$, all nonterminals are merged into the same $S_h \in N_h$. In addition, if W monotonically increases then the number of nonterminals monotonically increases. Obviously, the number of nonterminals in a hypothesis grammar is bounded by $|R|(< l_q^2|Q|$ where l_q is the length of the longest word in Q).

Now, we define an observation table (R, W, T, H). R is the set of 3-tuples defined in (step 1). $W \subset \Sigma^+$ is a finite set which is closed under not only prefixes but also suffixes. At the beginning of the algorithm, W is initialized to $\{u \in \Sigma^+ \mid x, y \in \Sigma^*, and\ x \cdot u \cdot y \in Q\}$.

T is a mapping $R \times W \to \{0, 1\}$. It represents results of observations. Let $w \in W$ and $r = (p_r, m_r, s_r) \in R$, then

$$T(r, w) = \begin{cases} 1 & (MEMBER(p_r \cdot w \cdot short(r)) \\ & = yes, and\ for\ all\ w' \\ & \in proper\_pre(w),\ it\ holds\ that \\ & MEMBER(p_r \cdot w' \cdot short(r)) \\ & = no) \\ 0 & (otherwise). \end{cases}$$

For the learner, at most n times of membership queries about $p_r \cdot m_r \cdot a_n, p_r \cdot m_r \cdot a_{n-1} \cdot a_n, \cdots, p_r \cdot m_r \cdot a_1 \cdot a_2 \cdots a_n$ are enough to find $short(r)$ for $r \in R$ where $s_r = a_1 a_2 \cdots a_n$, and $a_1, a_2, \cdots, a_n \in \Sigma$. Thus it takes $O(|R||W|n)$ time complexity to find $T(r, w)$ for all $r \in R$ and $w \in W$, where $n \le l_q$.

H consists of three mappings. Its details are defined in the next step. We show an example of the observation table constructed from $Q = \{aabb\}$ and $W = \{a, b, aa, bb, ab, aab, abb, aabb\}$ in Table 1.

Definition 6: Let $r \in R$. Then $row(r)$ is the mapping $f : W \to \{0, 1\}$ such that

$$f(w) = T(r, w)$$

where $w \in W$. □

Then the learner makes N_h as follows.

$$N_h = \{row(r) \mid r \in R\}.$$

The start symbol S_h is set to $row(s_0)$ where $s_0 = (\varepsilon, w, \varepsilon)$ for some $w \in Q$. From the definition of $row()$, it holds that $row((\varepsilon, w, \varepsilon)) = row((\varepsilon, w', \varepsilon))$ for any pair of words $w \in Q$ and $w' \in Q$.

(step 3) finds the candidates of rules for a hypothesis grammar

At first, we define some relations among $r \in R$.

Definition 7: Let $r = (p_r, m_r, s_r) \in R$ and $s = (p_s, m_s, s_s) \in R$. Then r and s are in *head-body relation* if the followings hold.

- $p_s = p_r \cdot a_1$, $s_r = s_s$ and $m_r = a_1 \cdot m_s$ for $a_1 \in \Sigma$.
- For all $w \in W$ such that $a_1 w \in W$,

$$T(s, w) = 1 \iff T(r, a_1 w) = 1.$$ □

Definition 8: Let $r = (p_r, m_r, s_r) \in R$, $r_1 = (p_{r1}, m_{r1}, s_{r1})$ and $r_2 = (p_{r2}, m_{r2}, s_{r2}) \in R$. Then r is *partitionable* into r_1 and r_2 if the followings hold.

- $p_r = p_{r1}, s_r = s_{r2}, s_{r1} = m_{r2} s_r, p_{r2} = p_r m_{r1}$ and $m_{r1} m_{r2} = m_r$.
- For any $w \in W$ such that $T(r, w) = 1$, there exist $w_1 \in W$ and $w_2 \in W$ such that $w = w_1 w_2$ and the followings hold :

$$T(r_1, w_1) = 1\ and\ T(r_2, w_2) = 1.$$

- $T(r, u_1 u_2) = 1$ for any $u_1 \in W$ and $u_2 \in W$ such that
  - $u_1 \cdot u_2 \in W$,
  - $T(r_1, u_1) = 1$ and
  - $T(r_2, u_2) = 1$. □

Suppose that $r_1 \in R$ and $r_2 \in R$ are in head-body relation and $r_1 = (p_1, a \cdot m_1', s_1)$ for $a \in \Sigma$. Then the rule $row(r_1) \to a \cdot row(r_2)$ is one of candidates for a hypothesis grammar. In addition, if it holds that r_2 is partitionable into some $r_{21} \in R$ and $r_{22} \in R$ then a rule $row(r_1) \to a \cdot row(r_{21}) \cdot row(r_{22})$ is regarded as one of candidates, too.

Now, we define H which consists of the following

Table 1 An observation table.

		W									H		
	$T(\cdot,\cdot)$	a	b	aa	bb	ab	aab	abb	$aabb$		$handle(\cdot)$	$body(\cdot)$	$prt(body(\cdot))$
	r_1 :(ε , $aabb$, ε)	0	0	0	0	1	0	0	1	$\to$	a	r_2	{}
	r_2 :(a, abb, ε)	0	1	0	0	0	0	1	0	$\to$	a	r_3	$\{r_{2'} \cdot r_4\}$
	r_3 :(aa, bb, ε)	0	0	0	1	0	0	0	0	$\to$	b	r_4	{}
R	r_4 :(aab, b, ε)	0	1	0	0	0	0	0	0	$\to$	b	ε	{}
	$r_{1'}$:(a, ab, b)	0	0	0	0	1	0	0	1	$\to$	a	$r_{2'}$	{}
	$r_{2'}$:(aa, b, b)	0	1	0	0	0	0	1	0	$\to$	b	ε	{}
	r_5 :(a, a, bb)	1	0	0	0	0	0	0	0	$\to$	a	ε	{}

three mappings.

- $handle : R \to \Sigma$

  $$handle(r) = a_1 \in \Sigma,$$

  where $r = (p_r, a_1 \cdot m'_r, s_r) \in R$.
- $body : R \to R \cup \{\varepsilon\} \cup \{0\}$

  $$body(r) = \begin{cases} s \in R & \cdots (r \text{ and } s \text{ are in head-body relation}) \\ \varepsilon & \cdots (|m_r| = 1) \\ 0 & \cdots (otherwise) \end{cases}$$

  where $r = (p_r, m_r, s_r) \in R$.
- $prt : R \cup \{\varepsilon\} \cup \{0\} \to 2^{R^2}$

  $$prt(r) = \begin{cases} \{r_1 \cdot r_2 \in R^2 \mid r \text{ is partitionable into } r_1 \text{ and } r_2\} & \cdots (r \in R) \\ \{\} & \cdots (otherwise) \end{cases}$$

It holds that $|prt(body(r))| \leq |R^2|$ for all $r \in R$. The learner regards

$$\{row(r) \to handle(r) \mid r \in R, body(r) = \varepsilon\},$$
$$\{row(r) \to handle(r) \cdot row(body(r)) \mid r \in R, body(r) \neq \varepsilon \text{ and } body(r) \neq 0\}$$

and

$$\{row(r) \to handle(r) \cdot row(r_1) \cdot row(r_2) \mid r \in R, r_1 r_2 \in prt(body(r))\}$$

as the set of candidates of hypothesis rules.

(step 4) constructs a hypothesis grammar

To construct a hypothesis in a simple deterministic grammar, the learner reduces the set of candidate rules.

Definition 9: For $r \in R$, let

$$\Sigma_T(r) = \{a \in \Sigma \mid T(r, a \cdot w) = 1 \text{ for some } w \in \Sigma^*\}.$$

We define $\Sigma_{H_n}(r)$ for $r \in R$ inductively as follows.

$$\Sigma_{H_0}(r) = \{handle(r') \mid row(r) = row(r')\}$$
$$\Sigma_{H_n}(r) = \{handle(r') \mid row(r) = row(r') \text{ and } r' \text{ satisfies at least one of the following conditions}$$

- r' is of the form $(p_{r'}, a, s_{r'})$ for $a \in \Sigma$ and $p_{r'}, s_{r'} \in \Sigma^*$, or
- $\Sigma_T(r'') = \Sigma_{H_{n-1}}(r'')$ where $body(r') = r''$, or
- there exists $r''_1 r''_2 \in prt(body(r'))$ such that $\Sigma_T(r''_1) = \Sigma_{H_{n-1}}(r''_1)$ and $\Sigma_T(r''_2) = \Sigma_{H_{n-1}}(r''_2)$ $\}$

If $\Sigma_T(r) = \Sigma_{H_n}(r)$ for any $n \geq 0$ then r is called *valid*. □

If $\Sigma_T(r) \neq \Sigma_{H_l}(r)$ for some integer $l \geq 0$ then $\Sigma_T(r) \neq \Sigma_{H_k}(r)$ holds for any $k \geq l$. Thus there exists $n \leq |R|$ such that $\Sigma_{H_n}(r) = \Sigma_{H_m}(r)$ for any $m > n$. It implies that we can check whether r is valid or not with $\Sigma_{H_0}(r), \Sigma_{H_1}(r), \cdots, \Sigma_{H_n}(r)$ and $\Sigma_T(r)$.

For $r \in R$, we define

$$P_2(r) = \{row(r) \to handle(r) \cdot row(r_1) \cdot row(r_2) \mid r_1 r_2 \in prt(body(r)), \text{and both } r_1, r_2 \text{ are valid}\},$$
$$P_1(r) = \{row(r) \to handle(r) \cdot row(r_1) \mid body(r) = r_1, \text{and } r_1 \text{ is valid}\},$$
$$P_0(r) = \{row(r) \to handle(r) \mid body(r) = \varepsilon\}.$$

For $A = row(r) \in N_h$ and $a \in \Sigma$, we define

$$P_2(A, a) = \bigcup_{r' \in R, row(r) = row(r'), handle(r') = a} P_2(r'),$$
$$P_1(A, a) = \bigcup_{r' \in R, row(r) = row(r'), handle(r') = a} P_1(r'),$$
$$P_0(A, a) = \bigcup_{r' \in R, row(r) = row(r'), handle(r') = a} P_0(r').$$

Then $P(A, a)$ is defined as follows.

$$P(A, a) = \begin{cases} P_2(A, a) & \ldots (|P_2(A, a)| > 0) \\ P_1(A, a) & \ldots (|P_2(A, a)| = 0, \text{ and } |P_1(A, a)| > 0) \\ P_0(A, a) & \ldots (|P_0(A, a)| > 0) \\ \{\} & \ldots (otherwise) \end{cases}$$

The learner makes rules of the hypothesis grammar by selecting one rule from $P(A, a)$ arbitrarily for all $A \in N_h$ and $a \in \Sigma$. Formally, the following algorithm is applied to construct P_h.

```
Procedure hypothesis_rules(INPUT, OUTPUT);
INPUT: P(A,a) for all A ∈ N_h and a ∈ Σ;
OUTPUT: rules of the hypothesis grammar P_h;
begin
  P_h := {}
  for all A ∈ N_h begin
    for all a ∈ Σ begin
      select a rule A → a·β from
        P(A,a) arbitrarily; (Here β ∈ N_h^*)
      P_h := P_h ∪ {A → a·β};
    end
  end
  output P_h;
end.
```

If G_h contains a nonterminal which is not reachable and live then delete it from G_h. The time complexity of this simplification is bounded by a polynomial of the size of G_h [3]. Such G_h is a simple deterministic grammar and satisfies the following lemma.

Lemma 10: Let $A = row(r) \in N_h$ where $r \in R$. Let

$w \in W$. Then $T(r,w) = 1$ iff $A \overset{*}{\underset{G_h}{\Rightarrow}} w$.

Proof : We prove this lemma by induction on the length of w.

Assume that $|w| = 1$, i.e. $w = a \in \Sigma$. From the definition of T, it holds that $T(r,a) = 1$ iff r' is of the form $(p_{r'}, a, s_{r'})$ for all $r' \in R$ such that $row(r) = row(r')$ and $handle(r') = a$ where $p_{r'}, s_{r'} \in \Sigma^*$. It implies that $P_2(row(r), a) = P_1(row(r), a) = \{\}$ and $P_0(row(r), a) = \{row(r) \to a\}$ iff $T(r,a) = 1$. Hence, P_h has the rule $row(r) \to a$ iff $T(r,a) = 1$. Thus it holds that $T(r,w) = 1$ iff $row(r) \overset{*}{\underset{G_h}{\Rightarrow}} w$ when $|w| = 1$.

Now, suppose that $w = a_1 a_2 \cdots a_n$ where $a_i \in \Sigma$, $i = 1, 2, \cdots, n$ and this lemma holds for any $u \in W$ such that $|u| \leq n-1$, where $n \geq 2$.

Since r is valid, $T(r,w) = 1$ iff there exists $r' \in R$ such that $handle(r') = a_1$ and $row(r) = row(r')$ which satisfies at least one of the following conditions:

1. $body(r') = s$ and s is valid, or
2. $s_1 s_2 \in prt(body(r'))$ for $s_1, s_2 \in R$ and both s_1 and s_2 are valid.

Here, exactly one of the following two cases (A) and (B) is possible, since $n \geq 2$.

(A) The case in which $row(r) \to a_1 \cdot row(body(row(r')))$ is in P_h where the former case 1. holds for r'. Then from the second condition of Definition 7, it holds that

$$T(r,w) = 1 \iff T(s, a_2 a_3 \cdots a_n) = 1.$$

From the assumption of this induction,

$$T(s, a_2 a_3 \cdots a_n) = 1 \iff row(s) \overset{*}{\underset{G_h}{\Rightarrow}} a_2 a_3 \cdots a_n$$

holds. It implies that

$$T(r,w) = 1 \iff row(r) \overset{*}{\underset{G_h}{\Rightarrow}} w.$$

(B) The case in which $row(r) \to a_1 \cdot row(s_1) \cdot row(s_2)$ is in P_h where $s_1 s_2 \in prt(body(r'))$ and the latter case 2. holds for r'. Then from the second and the third conditions of Definition 8, there exists $1 < m < n$ such that

$$T(r,w) = 1 \iff$$
$$T(s_1, a_2 \cdots a_m) = 1 \text{ and } T(s_2, a_{m+1} \cdots a_n) = 1.$$

Now, from the assumption of this induction, it holds that

$$T(s_1, a_2 \cdots a_m) = 1 \iff row(s_1) \overset{*}{\underset{G_h}{\Rightarrow}} a_2 \cdots a_m$$

and

$$T(s_2, a_{m+1} \cdots a_n) = 1$$
$$\iff row(s_2) \overset{*}{\underset{G_h}{\Rightarrow}} a_{m+1} \cdots a_n.$$

It implies that

$$T(r,w) = 1 \iff row(r) \overset{*}{\underset{G_h}{\Rightarrow}} w.$$

Thus this lemma holds. □

(step 5) diagnoses the hypothesis grammar with counterexamples

Before asking an equivalence query, the learner checks consistency between a hypothesis language and the results of membership queries. From Lemma 10, there are no inconsistent results of membership queries for $w \in W$ and $r \in R$ such that $row(r)$ is in G_h. Thus $row(q)$ is not reachable and live if $q_p \cdot w \cdot short(q) \in (L_t - L(G_h)) \cup (L(G_h) - L_t)$ where $q = (q_p, q_m, q_s) \in R$ and $w \in W$.

When such a conflict word is found, the learner regards it as a counterexample. If it is not found then the learner asks an equivalence query and gets a counterexample when $L(G_h) \neq L_t$.

Then all sub-words of the counterexample are added into W and the learner goes back to (step 2).

When W is extended with w, some rules in P_h are deleted. Formally, we claim the next lemma.

Lemma 11: Let (R, W, T, H) be an observation table, $G_h = (N_h, \Sigma, P_h, S_h)$ be the hypothesis grammar constructed from (R, W, T, H) and $L(G_h) \neq L_t$. Suppose that $w \in (L_t - L(G_h)) \cup (L(G_h) - L_t)$ and let $W' = W \cup \{w' \in \Sigma^+ \mid x \cdot w' \cdot y = w, where\ x, y \in \Sigma^*\}$. We assume that the observation table constructed from R and W' is denoted by (R, W', T', H'). Let $G'_h = (N'_h, \Sigma, P'_h, S'_h)$ be the hypothesis grammar constructed from (R, W', T', H') and let H' consist of $handle'(), body'()$ and $prt'(body'())$.

Then there exists $r \in R$ which satisfies one of the following conditions.

1. $$\bigcup_{row(r)=row(r')} prt'(body'(r')) \subset \bigcup_{row(r)=row(r')} prt(body(r'))$$
2. $body'(r) = 0$ whereas $body(r) \neq 0$

Proof : Assume that this lemma does not hold. Because $prt(s)$ for any $s \in R$ monotonically decreases, it holds that $body(s) = body'(s)$ and

$$\bigcup_{row(s)=row(s')} prt(body(s')) = \bigcup_{row(s)=row(s')} prt'(body'(s'))$$

for all $s \in R$. It implies that $N_h = N'_h$. Thus we can select a rule from $P(A,a)$ for every $A \in N_h$ and $a \in \Sigma$ so that G'_h is homomorphic to G_h.

For G'_h, it holds that $T'(s_0, w) = 1$ when $w \in L_t$ or $T'(s_0, w) = 0$ when $w \notin L_t$ where $s_0 \in R$ such that $row(s_0) = S_h$.

Now, Lemma 10 holds independently of the selection from $P(A, a)$. Then it holds that $w \in L(G'_h)$ if $w \in L_t$ or $w \notin L(G'_h)$ if $w \notin L_t$. It contradicts to $L(G'_h) = L(G_h)$. □

6. An Example Run

Let $L_t = \{a^i b^i \mid i \geq 1\}$ be the target language, and $G_t = (N_t, \Sigma, P_t, S_t)$ where $N_t = \{S_t, A, B\}$, $\Sigma = \{a, b\}$ and $P_t = \{S_t \to aA, A \to aAB, A \to b, B \to b\}$. Let the representative sample given to the learner be $Q_t = \{aabb\}$.

(step 1) The learner initializes R, W and LOG.

$$
\begin{aligned}
R &:= \{(x, y, z) \mid z \in \Sigma^*, x, y \in \Sigma^+, xyz \in Q\} \\
&\quad \cup \{(\varepsilon, w, \varepsilon) \mid w \in Q\} \\
&= \{(a, aab, \varepsilon), (aa, bb, \varepsilon), (aab, b, \varepsilon), \\
&\qquad (a, ab, b), (aa, b, b), (a, a, bb)\} \\
&\quad \cup \{(\varepsilon, aabb, \varepsilon)\}, \\
W &:= \{y \in \Sigma^+ \mid x, z \in \Sigma^*, x \cdot y \cdot z \in Q\} \\
&= \{a, b, aa, bb, ab, aab, abb, aabb\}, \ \textit{and} \\
LOG &:= \{\}.
\end{aligned}
$$

(step 2) The learner finds $T(r, w)$ for all $r \in R$ and $w \in W$.

The learner fills $T(r, w)$ and obtains the observation table in the left side of Table 1. For example, we confirm that $T(r_{1'}, aabb) = 1$ and $T(r_5, aab) = 0$.

It is easy to check that $short(r_{1'}) = b$ and the membership queries about $a \cdot aabb \cdot b$, $a \cdot aab \cdot b$, $a \cdot aa \cdot b$ and $a \cdot a \cdot b$ are replied "no" except for $a \cdot aabb \cdot b$. Thus $T(r_{1'}, aabb) = 1$.

For r_5, we find $short(r_5) = bb$ because $a \cdot a \cdot b \notin L_t$. Now, it holds that $a \cdot aab \cdot bb \in L_t$ but $T(r_5, a) = 1$ and a is a proper prefix of aab. Thus $T(r_5, aab) = 0$.

Then we have $N_h = \{S, A, B, C, D\}$ and S_h such that $S = row(r_1) = row(r_{1'})$, $A = row(r_2) = row(r_{2'})$, $B = row(r_4)$, $C = row(r_3)$, $D = row(r_5)$, and $S_h = S$.

(step 3) The learner finds $handle(r)$, $body(r)$ and $prt(body(r))$ for all $r \in R$.

For example, we are concerned with r_2 in Table 1.

Obviously, $handle(r_2) = a$. It holds that r_2 and r_3 are in head-body relation because

- $aa = a \cdot a$, $\varepsilon = \varepsilon$ and $abb = a \cdot bb$, then it implies that the first condition of Definition 7 holds,
- $T(r_2, a \cdot bb) = 1$ and $T(r_3, bb) = 1$, then it implies that the second condition of Definition 7 holds.

It holds that r_3 is partitionable into $r_{2'}$ and r_4 because

- $aa = aa$, $\varepsilon = \varepsilon$, $b = b \cdot \varepsilon$, $aab = aa \cdot b$ and $b \cdot b = bb$, then it implies that the first condition of Definition 8 holds,
- $T(r_3, bb) = 1$, $T(r_{2'}, b) = 1$ and $T(r_4, b) = 1$, then it implies that the second and the third conditions of Definition 8 hold.

In addition, except for the above pair of $r_{2'}$ and r_4, r_3 is not partitionable into any pair of $s_1 \in R$ and $s_2 \in R$. Thus $prt(body(r_2)) = \{r_{2'} r_4\}$.

When the learner finds $handle(r)$, $body(r)$ and $prt(body(r))$ for all $r \in R$, the right side of Table 1 is obtained.

(step 4) The learner makes P_h.

It holds that $T(r_3, bb) = 1$ and $handle(r_3) = b$ then r_3 satisfies that $\Sigma_T(r_3) = \Sigma_{H_0}(r_3) = \{b\}$. For $r_{2'}$, it holds that $T(r_{2'}, b) = 1$, $T(r_{2'}, abb) = 1$, $handle(r_{2'}) = b$ and $handle(r_2) = a$ then $r_{2'}$ satisfies that $\Sigma_T(r_{2'}) = \Sigma_{H_0}(r_{2'}) = \{a, b\}$. We can check that $\Sigma_T(r) = \Sigma_{H_0}(r)$ for all $r \in R$ thus we can conclude that every $r \in R$ is valid.

Next, the learner makes P_h. It holds that $P_2(r_2) = \{row(r_2) \to a \cdot row(r_{2'}) \cdot row(r_4)\}$, $P_1(r_2) = \{row(r_2) \to a \cdot row(r_3)\}$ and $P_0(r_2) = \{\}$. Similarly, for $r_{2'}$, it holds that $P_2(r_{2'}) = \{\}$, $P_1(r_{2'}) = \{\}$ and $P_0(r_{2'}) = \{row(r_{2'}) \to b\}$. It implies that $P_2(row(r_2), a) = P_2(r_2)$, $P_1(row(r_2), a) = P_1(r_2)$, $P_0(row(r_2), b) = P_0(r_{2'})$ and $P_2(row(r_2), b) = P_1(row(r_2), b) = P_0(row(r_2), a) = \{\}$. Then $P(row(r_2), a) = \{row(r_2) \to a \cdot row(r_{2'}) \cdot row(r_4)\}$ and $P(row(r_2), b) = \{row(r_{2'}) \to b\}$.

When the learner finds $P(row(r), a)$ for all $row(r) \in N_h$ and $a \in \Sigma$, the following rules are obtained.

$$
\begin{aligned}
&row(r_1) \to a \cdot row(r_2) \\
&row(r_2) \to a \cdot row(r_{2'}) \cdot row(r_4) \\
&row(r_3) \to b \cdot row(r_4) \\
&row(r_4) \to b \\
&row(r_{2'}) \to b \\
&row(r_5) \to a
\end{aligned}
$$

Finally, the learner strips the above rules and obtains the hypothesis grammar such that $G_h = (\{S, A, B\}, \Sigma, \{S \to aA, A \to aAB, A \to b, B \to b\}, S)$.

(step 5) The learner guesses the grammar.

This hypothesis grammar does not contradict to any results of membership queries then the learner asks an equivalence query with this grammar. It holds that $L_t = L(G_h)$ then the learner obtains "yes" and terminates.

7. The Correctness and the Time Complexity of the Learning Algorithm

This learning algorithm terminates only when a correct hypothesis grammar has been guessed. It ensures the correctness of the algorithm.

Next, we are concerned with the time complexity of the learning algorithm.

Let l_q be the length of the longest word in the representative sample Q and l_c be the length of the longest counterexample returned by equivalence queries.

To initialize R and W, it takes $O(l_q^2|Q|)$ time. It implies that the time complexity of (step 1) and $|R|$ are bounded by $O(l_q^2|Q|)$. It holds that $|prt(body(r))| \leq |R|^2$ for all $r \in R$ from the definition of $prt()$. Then from Lemma 11, at most $|R|^2 + 2|R|$ times equivalence queries are enough to terminate the learning algorithm. Next, we analyze the time complexity of each step.

(step 2) : Filling T for all $r \in R$ and $w \in W$ takes $O(|R||W|l_w)$ time where l_w is the length of the longest word in W. Let $l = \max\{l_q, l_c\}$. Then l_w is bounded by $O(l(|R|^2+2|R|))$ because the increase of l_w is at most l when W is extended, and l_w is equal to l_q at the initial step. In addition, $|W|$ is bounded by $O(l_w^2(|R|^2+2|R|))$ because W is extended only in (step 5).

It holds that $|LOG|$ is bounded by the size of the observation table which is $O(|R||W|)$. Finally, the time complexity to make N_t is bounded by $O(|R|^2|W|)$ and $O(|R|)$ time is needed to define S_h.

(step 3) : For $r \in R$, finding $handle(r)$ takes $O(1)$ time. For fixed $r \in R$ and every $s \in R$, checking whether r and s satisfy the first condition of Definition 7 or not takes $O(|R|l_q)$ time. Moreover, checking whether they satisfy the second condition of Definition 7 or not takes $O(|R||W|)$ time. Thus finding $body(r)$ takes $O(|R|(l_q + |W|))$ time.

For fixed $r \in R$ and every pair of $s_1 \in R$ and $s_2 \in R$, the analysis of the time complexity to check whether $s_1 s_2 \in prt(body(r))$ or not is as follows.

- Checking the first condition of Definition 8 takes $O(|R|^2 l_q)$ time.
- Checking the second condition of Definition 8 takes $O(|R|^2|W|l_w)$ time.
- Checking the third condition of Definition 8 takes $O(|R|^2|W|^3)$ time.

Then finding $prt(body(r))$ takes $O(|R|^2(l_q + l_w|W| + |W|^3))$ time.

It is clear that the total time complexity of (step 3) is at most $|R|$ times of the above complexities.

(step 4) : For $r \in R$, $\Sigma_T(r)$ can be found in $O(|W|)$ time and finding $\Sigma_{H_0}(r)$ takes $O(|R|)$ time. To find $\Sigma_{H_n}(r)$ with $\Sigma_{H_{n-1}}(s)$ for all $s \in R$ takes $O(|R|(1 + 1 + |R|^2)) = O(|R|^3)$ time. Now, n is at most $|R|$ to distinguish whether r is valid or not. Thus, for all $r \in R$, we can decide whether r is valid or not in $O(|R|(|W| + |R| + |R|^4))$ time.

To construct $P_2(r)$, $P_1(r)$ and $P_0(r)$ takes $O(|R|^2)$, $O(1)$ and $O(1)$ time, respectively. $P_i(A, a)$ for $i = 0, 1, 2$ can be constructed in $|R||P_i(r)|$ time and $P(A, a)$ can be found in $O(1)$ time from $P_i(A, a)$ for $i = 0, 1, 2$. In addition, P_h is constructed in $|N_h||\Sigma|$ time from $P(A, a)$ for $A \in N_h$ and $a \in \Sigma$.

(step 5) : Obviously, it takes $O(l^2)$ time to update W where l is the length of the counterexample or the longest word in LOG.

Now, summering the above all, we have proved Theorem 4.

8. Conclusion

In this paper, we showed that simple deterministic languages are polynomial time learnable from MAT and a representative sample. The time complexity of the learning algorithm is bounded by a polynomial in the size of the grammar which generates the target language, the cardinality of the representative sample and the length of the longest word in the representative sample and counterexamples.

When we are concerned with generalization of learning via queries, studies of learner's environment becomes more important. If there exists an algorithm which enumerates a representative sample effectively then it implies that there exists a learning algorithm of simple deterministic languages from MAT. For the other classes of languages, whether there exists such a subset or not is an interesting problem.

Acknowledgement

The authors are grateful to the reviewers for their careful reading of the manuscript and their very useful comments.

References

[1] D. Angluin, "A note on the number of queries needed to identify regular languages," Inf. Control, vol.51, no.1, pp.76–87, Oct. 1981.
[2] D. Angluin, "Learning regular sets from queries and counterexamples," Inf. Computation, vol.75, no.2, pp.87–106, Nov. 1987.
[3] J.E. Hopcroft and J.D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, Reading MA, 1979.
[4] H. Ishizaka, "Polynomial time learnability of simple deterministic languages," Machine Learning, vol.5, no.2, pp.151–164, June 1990.
[5] A.J. Korenjak and J.E. Hopcroft, "Simple deterministic languages," Proc. IEEE 7th Annual Symposium on Switching and Automata Theory, pp.36–46, Oct. 1966.

Yasuhiro Tajima was born in Tokyo, Japan, 1971. He received the B.E and M.E degrees in Communications and Systems Engineering from the University of Electro-Communications in 1994 and 1996, respectively. From 1996 through 1998, he joined Ishikawajima-Harima Heavy Industries Co. Ltd. Currently he is a Ph. D. candidate at the University of Electro-Communications. His research interests are in computational learning theory and formal language theory. He is a member of Japanese Society for Artificial Intelligence.

Etsuji Tomita was born in 1942. He received his D.E. degree from Tokyo Inst. Tech. in 1971. He was appointed Assoc. Professor in the Dept. of Comm. Eng., Fac. of Elec.-Comm. UEC in 1976, and became Professor in 1986. Since Apr. 1999, he has been a Professor in the Dept. of Inform. and Comm. Eng., Fac. of Elec.-Comm. and also in the Dept. of Comm. and Systems Eng., Grad. School of Elec.-Comm., UEC. His current research interests include the computational learning theory, the design and analysis of algorithms, and others. He received the Yonezawa Award from IECE in 1971. He served as a member of the Editorial Board of IECE, and is presently in charge of a Chairman of SIG-MPS of IPSJ. Dr. Tomita is a member of ACM, EATCS, JSAI, IPSJ, and JNNS.

Mitsuo Wakatsuki was born in Tokyo, Japan on December 17, 1965. He received his B.E. degree in Communication Engineering in 1988, and his M.E. and D.E. degrees in Communications and Systems Engineering in 1990 and 1993, respectively, from the University of Electro-Communications. He is now a Research Associate in the Department of Information and Communication Engineering, Faculty of Electro-Communications at the University of Electro-Communications. His research interests include formal language theory and computational learning theory. Dr. Wakatsuki is a member of the Information Processing Society of Japan and Japanese Society for Artificial Intelligence.