INFERENCE OF REGULAR PATTERNS WITH ERASING AND NONERASING VARIABLES

Jin Uemura

Received December 20, 2001; revised April 16, 2002

Abstract. A *pattern* is a string of constants and variables, and is *regular* if each variable occurs in the pattern at most once. The present paper deals with the problem of learning languages generated by regular patterns with two kinds of variables, *erasing* and *nonerasing* from positive examples within Gold's model. First, we show an equivalence theorem of a semantic containment $L(p) \subseteq L(q)$ and a syntactic containment $p \preceq q$ for regular patterns p and q. Then we show that the class of regular pattern languages is polynomial time inferable from positive examples.

1 Introduction. A *pattern* is a string consisting of constants and variables. The language of a pattern is the set of all constant strings obtained from the pattern by substituting constant strings for variables in the pattern. A *nonerasing* substitution for a pattern is not allowed to replace a variable by the empty string whereas an *erasing* substitution may replace a variable by the empty string. In the Angluin's definition[1] of pattern languages, erasing substitutions are disallowed. Angluin[1] has proved the class of nonerasing pattern language is inductively inferable from positive examples in the framework of *identification in the limit* due to Gold[5]. Learnability of pattern languages first introduced by Angluin[1] has been extensively investigated (e.g., see Arimura, Shinohara and Otsuki[4], Mukouchi[8], Sato, Mukouchi and Zheng[10], Shinohara[11, 12], and Shinohara and Arimura[13]).

On the other hand, pattern languages allowing erasing substitutions has been first introduced by Shinohara[11] using the terminology of *extended* pattern languages. He has shown that the subclass consisting of *regular* erasing (or extended) pattern languages is polynomial time inferable from positive examples. A pattern is regular if each variable appears at most once in the pattern. The inferability of the class of erasing pattern languages is unsolved until now.

In the present paper, we introduce patterns consisting of constants and two kinds of variables: erasing and nonerasing. The language of a pattern p, denoted by $L(p)$, is the set of constant strings obtained by substituting constant strings (possibly the empty string) for erasing variables and *nonempty* constant strings for nonerasing variables in the pattern. The class $\mathcal{PL}$ of such pattern languages *properly* contains the union of the classes of erasing pattern languages and nonerasing pattern languages. In fact, for a pattern $p = yaz$ over $\Sigma = \{a, b\}$, the language $L(p)$ given by $\{uav \mid u \in \Sigma^*, v \in \Sigma^+\}$ can be generated by neither erasing nor nonerasing patterns, where y is an erasing variable and z is a nonerasing variable. As easily seen, a language generated by a pattern containing erasing variables can be expressed by union of finitely many nonerasing pattern languages. As well known, the membership problem for nonerasing pattern languages is NP-complete([1]) as well as for erasing, and thus so is for languages of $\mathcal{PL}$.

2000 *Mathematics Subject Classification.* 68T05, 68T35.

Key words and phrases. Inductive inference, Regular pattern language, Positive examples, Identification in the limit.

In this paper, we investigate the problem of efficient learning of regular pattern languages from positive examples within Gold's model of identification in the limit. The class of pattern languages is denoted by $\mathcal{RPL}$.

For patterns p and q, q is a generalization of p, denoted by $p \preceq q$, if p is obtained from q by substituting some patterns for variables in q, where patterns substituted for *nonerasing* variables are assumed to contain at least one constant or nonerasing variable. Then we first show an equivalence theorem of a semantic containment $L(p) \subseteq L(q)$ and a syntactic containment $p \preceq q$ for any regular patterns p and q, provided that the number of constants is more than two. Moreover, we show that for each regular pattern p, the finite set $S(p) \subseteq L(p)$ is a characteristic set of $L(p)$. Here $S(p)$ is the finite set of constant strings generated by substituting any constant to each nonerasing variable and any constant or the empty string ε to each erasing variable in p, and a finite set $S \subseteq L(p)$ is a characteristic set of $L(p)$ if $S \subseteq L(q)$ implies $L(p) \subseteq L(q)$ for any q. Thus if $S(p) \subseteq L(q)$ implies $L(p) \subseteq L(q)$ for any q. The notion of a characteristic set was introduced by Angluin[3]. A characteristic set is useful for building an efficient learning algorithm of languages.

Next, we show that the class $\mathcal{RPL}$ has finite thickness as well as the classes of erasing and nonerasing pattern languages ([11, 12]). *Finite thickness* has been introduce by Angluin[2] as a sufficient condition for inferability from positive examples. Wright[7, 15] has introduced a more general notion of language classes called *finite elasticity* than finite thickness. Finite elasticity has a good property in the sense that it is not only a sufficient condition for inferability but also closed under various class operations such as union, intersection and so on ([6, 9, 15]). Thus the classes obtained by applying to these class operations to the class $\mathcal{RPL}$ is also inferable from positive examples.

Concerning to a class with finite elasticity, it is well known that if both of the membership problem and the MINL problem of the class are polynomial time computable, then the class is polynomial time inferable from positive examples. The MINL problem is a problem finding one of the minimal languages in the class containing a given nonempty finite set of Σ^*.

We finally show these two problems are polynomial time computable, and thus the class of regular pattern languages is polynomial time inferable from positive examples, as well as the classes of erasing and nonerasing pattern languages ([11, 12]).

2 Regular patterns with erasing and nonerasing variables.

We introduce patterns with erasing and nonerasing variables and languages defined by such patterns. Let N be the set of nonnegative integers, and let $\sharp S$ be the cardinality of a set S.

2.1 Pattern languages.

Let Y and Z be countable sets of variables. Assume that Σ, Y and Z are mutually disjoint. Elements of Y are *erasing* variables, denoted by $y, y_0, y_1, \cdots$, and those of Z are *nonerasing* variables, denoted by $z, z_0, z_1, \cdots$, while elements in Σ are *constants*. We put $X = Y \cup Z$, whose elements are denoted by $x, x_0, x_1, \cdots$. A *pattern* is a finite string (possibly the empty string) over $\Sigma \cup X$. The symbol ε denotes the empty string. For a pattern p, the *length* of p, denoted by $|p|$, is the number of symbols in p. The set of patterns is denoted by $\mathcal{P}$.

A *substitution* θ is a homomorphism from $\mathcal{P}$ to $\mathcal{P}$ which maps every constant to itself, every *nonerasing* variable to a pattern containing at least one element in $\Sigma \cup Z$ and every *erasing* variable to a pattern. A set of replacements $\{x_0 := p_0, x_1 := p_1, \cdots, x_n := p_n\}$ is the substitution that maps each variable x_i to the pattern p_i and any other variables to themselves. For a pattern p and a substitution θ, the image of p by θ is denoted by $p\theta$.

For a pattern p, we denote by $p_{n\varepsilon}$ the pattern obtained from p by substituting ε to every erasing variable in p. Thus the pattern $p_{n\varepsilon}$ is a string of $(\Sigma \cup Z)^*$.

For patterns p and q, q is a *generalization* of p or p is an *instance* of q, denoted by $p \preceq q$, if there is a substitution θ such that $p = q\theta$. Patterns p and q are *equivalent*, denoted by $p \equiv q$, if $p \preceq q$ and $q \preceq p$. Note that $p \equiv q$ does not always imply $p = q$. For instance, $y_0 \preceq y_0y_1$ and $y_0y_1 \preceq y_0$ imply $y_0 \equiv y_0y_1$ but $y_0 \neq y_0y_1$.

For a pattern p, the language generated by p is defined by $L(p) = \{w \in \Sigma^* \mid w \preceq p\}$. Clearly $|w| \geq |p_{n\varepsilon}|$ if $w \in L(p)$, and the shortest length of strings in $L(p)$ is given by $|p_{n\varepsilon}|$. And $p \preceq q$ implies $L(p) \subseteq L(q)$, and so $p \equiv q$ implies $L(p) = L(q)$. A language L over Σ is a *pattern language* if there is a pattern p such that $L(p) = L$. For instance, a language $L = \{uav \mid u \in \Sigma^*, v \in \Sigma^+\}$ over $\Sigma = \{a, b\}$ is a pattern language because $L = L(yaz)$. We should remark that the language L can be generated by neither erasing patterns nor nonerasing patterns. Thus the class $\mathcal{PL}$ *properly* contains union of the classes of erasing and nonerasing pattern languages.

2.2 Regular pattern languages. A pattern p is *regular* if each variable occurs at most once in p. The classes of regular patterns and regular pattern languages are denoted by $\mathcal{RP}$ and $\mathcal{RPL}$, respectively.

A regular pattern p containing at least one variable can be expressed as $p = w_0\alpha_0w_1\alpha_1 \cdots w_{n-1}\alpha_{n-1}w_n$ for some n where $w_0, w_n \in \Sigma^*, w_i \in \Sigma^+$ and $\alpha_j \in X^+$ for $i = 1, 2, \cdots, n-1$ and $j \in N$. A pattern p is of *canonical* form, if $\alpha_i \in Y \cup Z^+$ for any i or $p \in \Sigma^*$. In this paper, we do not distinguish two regular patterns if they are equal except renaming variables. However, we do not allow renaming variables between an erasing variable and a nonerasing variable. Thus $yaz = y_0az_1$, and $yaz \neq yy_0az$ although yaz is equivalent to yy_0az.

Theorem 2.1. The class of canonical regular patterns constitutes a partially ordered set under the relation $\preceq$.

Proof. Let p and q be canonical regular patterns. It is enough to show that if $p \equiv q$, then p is identical to q except renaming variables. If $p \in \Sigma^*$, it is obvious. We assume p contains at least one variable.

Let $p = w_0\alpha_0w_1\alpha_1 \cdots w_{n-1}\alpha_{n-1}w_n$ where $\alpha_i \in Y \cup Z^+$ for each i. By $p \equiv q$, there are substitutions θ and θ' such that $q = p\theta$ and $p = q\theta'$. Thus $p = p\theta\theta' = w_0(\alpha_0\theta\theta')w_1(\alpha_1\theta\theta') \cdots w_{n-1}(\alpha_{n-1}\theta\theta')w_n$ holds. It means that $\alpha_i\theta\theta' = \alpha_i$ must hold for every i. Clearly since p and q are in canonical forms, it implies that $\alpha_i\theta = \alpha_i$ for each i. Hence $p = q$. □

As easily seen, a regular pattern $x_0x_1 \cdots x_n$ is equivalent to y if $x_i \in Y$ for $i = 0, 1, \cdots, n$, Otherwise $x_0x_1 \cdots x_n$ is equivalent to $z_0z_1 \cdots z_{k-1}$, where k is the number of nonerasing variables of $x_0, x_1, \cdots, x_n$. Thus it follows immediately:

Theorem 2.2. For any regular pattern p, there uniquely exists the canonical regular pattern q such that $L(p) = L(q)$.

A pattern p is a *substring* of a pattern q if there are patterns r, r' such that $q = rpr'$. The following result can be shown similarly to that for nonerasing regular pattern languages given by Mukouchi[8]:

Lemma 2.3. Suppose $\sharp\Sigma \geq 3$. Let p and q be regular patterns. If $L(p) \subseteq L(q)$, then a constant substring of q is also a substring of p.

We should note that the condition $\sharp\Sigma \geq 3$ is necessary in the above theorem. Indeed, if $\sharp\Sigma = 2$, say $\Sigma = \{a, b\}$, $L(y_0ay_1by_2) \subseteq L(y_0aby_1)$ holds although the substring ab of y_0aby_1 is not a substring of $y_0ay_1by_2$.

Hereafter, we assume that $\sharp\Sigma \geq 3$ and patterns are regular.

3 Characteristic sets and completeness. For a regular pattern p, a finite set $S \subseteq \Sigma^*$ is a *characteristic set* of $L(p)$, if $S \subseteq L(q)$ implies $L(p) \subseteq L(q)$ for any $q \in \mathcal{RP}$.

For a regular pattern p, the set $S(p)$ denotes the set of all constant strings obtained from p by substituting any constant for each *nonerasing* variable in p, and any constant or ε for each *erasing* variable in p. Thus the lengths of the longest strings and the shortest strings in $S(p)$ equal $|p|$ and $|p_{n\varepsilon}|$, respectively.

Let p and q be regular patterns which satisfy with $p \preceq q$ and $p = p_0 r p_1$ for some patterns $r(\neq \varepsilon), p_0, p_1$. The substring r in $p_1 r p_2$ is *generable by variable substitution* for q if there is a variable x_i in q and a substitution $\theta = \{x_0 := r_0, x_1 := r_1, \cdots, x_i := r' r r'', \cdots, x_n := r_n\}$ such that $p_0 = (q_0\theta)r'$ and $p_1 = r''(q_1\theta)$ where $q = q_0 x_i q_1$. This variable x_i in q generates the substring r of p. As easily seen, $p_0 z p_1 \preceq q$ holds when x_i in q generates r, and particularly, $p_0 y p_1 \preceq q$ holds if $x_i \in Y$. Note that every variable in p is always generable by variable substitution for q when $p \preceq q$.

The following result can be shown similarly to that for nonerasing regular patterns given by Sato et al.[10]:

Lemma 3.1. Let p and q be regular patterns. If $p\{z := a_i\} \preceq q$ for distinct constants $a_0, a_1, a_2 \in \Sigma$, then $p \preceq q$.

Note that, in the above lemma, $p \preceq q$ does not hold if we replace a nonerasing variable z with an erasing variable y. In fact, consider $p = a_0 y a_1$ and $q = a_0 z a_1$, where $\Sigma = \{a_0, a_1, a_2\}$. Clearly $p\{y := a_i\} \preceq q$ for $i = 0, 1, 2$ but $p \not\preceq q$. Concerning with the variable x in p to be erasing, we have the following result:

Lemma 3.2. Let p and q be canonical regular patterns. If $p\{y := z\} \preceq q$ and $p\{y := \varepsilon\} \preceq q$, then $p \preceq q$.

Proof. Clearly if the pattern p does not contain the variable y, our lemma is valid. Thus let $p = p_0 y p_1$ for some patterns p_0 and p_1. We prove only for the case of $p_i \neq \varepsilon$ for $i = 0, 1$. It can be shown similarly for the other cases.

Assume $p_i \neq \varepsilon$ for $i = 0, 1$. Since p is in canonical form, $p_0 = p_0' a$ and $p_1 = b p_1'$ for some $a, b \in \Sigma$ and for some patterns p_0', p_1'. Suppose that $p \not\preceq q$. Then by $(p\{y := \varepsilon\} =)$ $p_0'(ab)p_1' \preceq q$, there is a substitution θ such that $p_0'(ab)p_1' = q\theta$. If the substring a, b or ab in $p_0'(ab)p_1'$ is generable by variable substitution for q, then $p \preceq q$ holds, and a contradiction. Hence there are patterns q_0, q_1 such that

$$q = q_0(ab)q_1, \quad \text{and} \quad p_i' = q_i\theta \text{ for } i = 0, 1.$$

Similarly by $(p\{y := z\} =) p_0'(azb)p_1' \preceq q$, there is a substitution θ' satisfying $p_0'(azb)p_1' = q\theta'$. Let x be a variable in q to generate the variable z in $p_0'(azb)p_1'$.

Consider the case that q_0 contains x.

In this case, there are patterns q_0', q_0'' such that

$$q_0 = q_0' x q_0'', \qquad p_0' a = q_0'\theta' \qquad \text{and} \qquad b p_1' = q_0''(ab)q_1\theta'.$$

Let us put $r_0 = q_0'\theta, r_1 = x\theta$ and $r_2 = q_0''\theta$. Then we have $p_0'(ayb)p_1' = (r_0 r_1 r_2)(ayb)p_1' = r_0(r_1 r_2 a)(b y p_1') = (q_0'\theta)(x\{x := r_1 r_2 a y\})(q_0'' a b q_1\theta')$, and so $p \preceq q$.

We can prove similarly for the other case that q_1 contains x. □

The class $\mathcal{RP}$ is *complete* if the semantic containment $L(p) \subseteq L(q)$ is equivalent to the syntactic containment $p \preceq q$.

In terms of Lemma 3.1 and Lemma 3.2, we obtain the following equivalence theorem which plays an important role in learning regular patterns from positive examples:

Theorem 3.3. Let p and q be canonical regular patterns. The following three relations are equivalent:

$$\text{(i) } S(p) \subseteq L(q), \qquad \text{(ii) } L(p) \subseteq L(q), \qquad \text{(iii) } p \preceq q.$$

Proof. Clearly (iii) implies (ii) and (ii) implies (i). It suffices to show that (i) implies (iii), that is, $S(p) \subseteq L(q)$ implies $p \preceq q$. It is done by mathematical induction on the number n of variables in p.

In case of $n = 0$, it is obvious since p is a constant string. Assume that it is valid for any $k(\leq n)$. Let p be a canonical regular pattern with $n+1$ variables and $p = p_0 x p_1$ for some p_0, p_1 and for some variable x. Since $\sharp\Sigma \geq 3$, there are three distinct constants $a_0, a_1, a_2 \in \Sigma$. For each $i = 0, 1, 2$, the pattern $p_0 a_i p_1$ is of n variables and $S(p_0 a_i p_1) \subseteq S(p)$ holds. Thus by induction hypothesis, we have $p_0 a_i p_1 \preceq q$ for $i = 0, 1, 2$. It implies by Lemma 3.1 that $p_0 z p_1 \preceq q$. If $x \in Z$, clearly $p \preceq q$. Otherwise $S(p_0 p_1) \subseteq S(p)$ because of $x \in Y$. Since $p_0 p_1$ is also of n variables, it turns out by induction hypothesis that $p_0 p_1 \preceq q$. Hence both $p_0 p_1$ and $p_0 z p_1$ are instances of q. Appealing to Lemma 3.2, $p = p_0 y p_1 \preceq q$ holds. This completes our proof. □

By the above theorem, the following important results can be shown immediately:

Corollary 3.4. The class $\mathcal{RP}$ of regular patterns is complete.

Corollary 3.5. For any regular pattern p, the set $S(p)$ is a characteristic set of $L(p)$.

4 Efficient learning of regular patterns.

In this section, we first give a framework of identification in the limit from positive examples due to Gold[5]. Then we show that the class of regular pattern languages is polynomial time inferable from positive examples.

4.1 Inductive inference from positive examples.

A language class $\mathcal{L} = L_0, L_1, \cdots$ over Σ is an *indexed class of recursive languages* if there is a computable function $f : N \times \Sigma^* \to \{0, 1\}$ such that $f(i, w) = 1$ if $w \in L_i$, otherwise 0. Hereafter we confine ourselves to indexed classes of recursive languages.

An infinite sequence of strings $w_0, w_1, \cdots$ over Σ is a *positive presentation* of a language L if $L = \{w_n \mid n \geq 0\}$ holds.

An *inference machine* is an effective procedure that requests examples from time to time and produces natural numbers, called hypotheses, from time to time. Note that hypotheses in the present paper are assumed to correspond to regular patterns. Let M be an inference machine and $\sigma = w_0, w_1, \cdots$ be an infinite sequence of constant strings. We denote by h_n, the hypothesis produced by M after the strings $w_0, w_1, \cdots, w_n$ are fed to M. The inference machine M is *polynomial time updating* if M produces h_n in time polynomial of the sum of lengths of strings so far received. The inference machine M is *consistent* if $\{w_0, w_1, \cdots, w_n\} \subseteq L_{h_n}$ for any n, and is *conservative* if $w_n \in L_{h_{n-1}}$ implies $h_n = h_{n-1}$ for any $n > 1$.

An inference machine M on input σ *converges* to h if there is an integer $n_0 \in N$ such that $h_n = h$ for every $n \geq n_0$. M *identifies in the limit* a language L *from positive examples*, if for any positive presentation σ of L, M on input σ converges to h which satisfies $L = L_h$.

An inference machine M infers the class $\mathcal{L}$ if M identifies in the limit a language L from positive examples for any $L \in \mathcal{L}$. A class of languages $\mathcal{L}$ is *inferable from positive examples* if there is an inference machine which infers the class $\mathcal{L}$. A class $\mathcal{L}$ is *polynomial time inferable from positive examples* if there is a consistent, conservative and polynomial time updating inference machine that infers the class $\mathcal{L}$.

Angluin[2] gave a characterizing theorem for language classes to be inferable from positive examples, and showed that a class with finite thickness is inferable from positive

examples. A class $\mathcal{L}$ has *finite thickness* if $\sharp\{L \in \mathcal{L} \mid S \subseteq L\}$ is finite for any nonempty finite set $S \subseteq \Sigma^*$.

Wright[15] gave a more general notion for a class called finite elasticity than finite thickness. A class $\mathcal{L}$ has *finite elasticity*, if there is no infinite sequence of constant strings $w_0, w_1, \cdots$ and no infinite sequence of languages $L_{i_0}, L_{i_1}, \cdots$ in $\mathcal{L}$ satisfying $\{w_0, w_1, \cdots, w_{n-1}\} \subseteq L_{i_n}$ but $w_n \notin L_{i_n}$ for every $n \in N$. Finite elasticity is a good property in a sense that it is not only a more general sufficient condition for inferability than finite thickness, but also closed under various class operations such as union, intersection and so on ([6, 15]).

For a class $\mathcal{L}$ with finite elasticity, an efficient inference algorithm was presented as given below if both of the membership problem and the MINL problem for the language class $\mathcal{L}$ are polynomial time computable. The membership problem for $\mathcal{L}$ is a problem deciding whether a constant string belongs to a language in $\mathcal{L}$. The MINL problem for $\mathcal{L}$ is a problem finding one of the minimal languages within $\mathcal{L}$ containing a nonempty finite set of constant strings. A problem is polynomial time computable if there is a procedure that computes the answer of the problem in polynomial time of the sum of lengths of input strings.

Theorem 4.1 (Angluin[1], Arimura et al.[4]). If a class $\mathcal{L}$ has finite elasticity and the MINL problem for $\mathcal{L}$ is computable, the procedure INFER below infers $\mathcal{L}$ from positive examples. Furthermore, if the membership problem and the MINL problem for $\mathcal{L}$ are polynomial time computable with respect to the sum of lengths of the pattern and the word, then $\mathcal{L}$ is polynomial time inferable from positive examples by the procedure INFER.

Procedure INFER
begin
$S := \{w_0\}$; $h_0 :=$ MINL(S); $n := 1$;
repeat
read the next example w_n;
$S := S \cup \{w_n\}$;
if $w_n \in L_{h_{n-1}}$ **then** $h_n := h_{n-1}$ **else** $h_n :=$ MINL(S);
output h_n;
$n := n + 1$
forever
end

4.2 Learning regular patterns. Let $S \subseteq \Sigma^*$ be a nonempty finite set. If a regular pattern language $L(p)$ contains the set S, then $|p| \leq l_{\min}$ holds where $l_{\min}$ is the length of the shortest strings in S. It means that the number of regular pattern languages containing S is at most finite, that is, the class $\mathcal{RPL}$ has finite thickness. Thus we have the following result.

Theorem 4.2. The class $\mathcal{RPL}$ has finite thickness. Thus the class is inferable from positive examples.

Clearly, since a language class with finite thickness has finite elasticity, by the above theorem, the class $\mathcal{RPL}$ has finite elasticity. Moreover, since the property of finite elasticity is closed under various class operations such as union, intersection and so on([9, 15]), the classes obtained by applying to such operations finitely many times to the class $\mathcal{RPL}$ are also inferable from positive examples. As mentioned in the previous subsection, if both of the membership problem and the MINL problem of regular pattern languages are polynomial time computable, it implies that the class $\mathcal{RPL}$ is polynomial time inferable from positive examples by the procedure INFER.

The membership problem for regular pattern languages is polynomial time computable as well as that for erasing or nonerasing regular pattern languages([11, 12]) as follows:

Lemma 4.3. For any string $w \in \Sigma^*$ and any regular pattern p, whether $w \in L(p)$ or not is computable in time $O(|w| + |p|)$.

Let $v, w \in \Sigma^+$ and $w = a_0 a_1 \cdots a_n$ for $a_i \in \Sigma$ for $i = 0, 1, \cdots, n$. A constant string v is a *subsequence* of w, denoted by $v \leq w$, if $v = a_{i_0} a_{i_1} \cdots a_{i_k}$ for some $i_0, i_1, \cdots, i_k$ with $0 \leq i_0 < i_1 < \cdots < i_k \leq n$. We note that the empty string ε is a subsequence of any constant string. There is no confusion by using the same symbol as $\leq$ for a pair of numbers.

For a set $S \subseteq \Sigma^*$, we define the set of *common subsequences* CS and the set of *maximal common subsequences* MCS of S as follows:

$$\mathrm{CS}(S) = \{v \in \Sigma^* \mid \forall w \in S, v \leq w\}, \ \mathrm{MCS}(S) = \{v \in \mathrm{CS}(S) \mid \forall w \in \mathrm{CS}(S), v \not\prec w\}.$$

If $S \subseteq L(p)$ for a regular pattern p, then the constant string, denoted by $c(p)$, obtained from p by deleting all variables in p is a common subsequence of S. Conversely, if the constant string s is a common subsequence of S, then $S \subseteq L(p)$ holds, where $s = a_0 a_1 \cdots a_k$ and $p = y_0 a_0 y_1 y_1 \cdots y_k a_k y_{k+1}$. Note that this inclusion is not always valid for nonerasing variables z_i instead of erasing variables y_i. Then clearly $S \subseteq L(p)$ implies that $c(p)$ is a common subsequence of S. Shimizu[14] has given a procedure computing one of the maximal common subsequences of a finite set S of constant strings in time $O(l_{\min} \times \sum_{w \in S} |w|)$.

Lemma 4.4. Let $S \subseteq \Sigma^*$ and $s \in \mathrm{MCS}(S)$. The following procedure MINL computes a regular pattern that generates a minimal language containing the set S within the class $\mathcal{RPL}$ in time $O((l_{\min})^2 \times \sum_{w \in S} |w|)$.

Procedure MINL
Inputs: a finite set of constant strings S
and a maximal common subsequence $s = a_0 a_1 \cdots a_k$ of S where $|s| = k + 1$
Output: a regular pattern
begin
let l be the length of the shortest strings in S;
$m := l - (k + 1)$;
$q_0 := y_0 a_0 y_1 a_1 \cdots y_k a_k y_{k+1}$;
for $i = 0$ **to** $k + 1$ **do**
begin
if $S \subseteq L(q_{i-1}\{y_i := \varepsilon\})$ **then begin** $q_i := q_{i-1}\{y_i := \varepsilon\}$; **goto** E **end**;
for $j = m$ **downto** 1 **do**
if $S \subseteq L(q_{i-1}\{y_i := z_{i,1} z_{i,2} \cdots z_{i,j}\})$ **then**
begin $q_i := q_{i-1}\{y_i := z_{i,1} z_{i,2} \cdots z_{i,j}\}$; **goto** E **end** ;
$q_i := q_{i-1}$
E: **end**;
output q_{k+1}
end

Proof. By the procedure MINL, since $s \in \mathrm{MCS}(S)$ and $q_0 = y_0 a_0 y_1 a_1 \cdots y_k a_k y_{k+1}$, $S \subseteq L(q_0)$ holds. Moreover, as easily seen, $S \subseteq L(q_i)$ for $i = 0, 1, \cdots, k + 1$. We denote by q the output q_{k+1} of the procedure for short. Suppose to the contrary that there is a regular pattern p such that $S \subseteq L(p) \subset L(q)$. By Theorem 3.3, we can assume that p is of canonical form. Put $q = \beta_0 a_0 \beta_1 a_1 \cdots \beta_k a_k \beta_{k+1}$ for some $\beta_i \in Y \cup Z^+ \cup \{\varepsilon\} (i = 0, 1, \cdots, k + 1)$, where $s = a_0 a_1 \cdots a_k$ and each $a_i \in \Sigma$. Since $L(p) \subset L(q)$, by Theorem 3.3, $p \prec q$, that is,

$p \preceq q$ and $p \neq q$. Thus $p = p_0 a_0 p_1 a_1 \cdots p_k a_k p_{k+1}$ for some patterns p_i with $p_i \preceq \beta_i$ for $i = 0, 1, \cdots, k+1$. We first note that for each i, p_i contains no constants. In fact, if p_i contains constants, then the constant string $c(p)$ defined above is a common subsequence of S because of $S \subseteq L(p)$. Moreover, the maximal subsequence s of S is a subsequence of $c(p)$ and $s \neq c(p)$. It leads a contradiction to the choice of s. Thus $p_i \in Y \cup Z^+ \cup \{\varepsilon\}$ for each i.

Next, we note that by $p_i \preceq \beta_i$ for each i, $\beta_i = \varepsilon$ implies $p_i = \varepsilon$, $\beta_i \in Z^+$ implies $p_i \in Z^+$ and $p_i = y_i$ implies $\beta_i = y_i$. By $p \prec q$, it follows that $p_i \prec \beta_i$ for some $i \leq k+1$. Let i_0 be the minimum integer of such i's. By the above, $\beta_{i_0} \neq \varepsilon$ and $p_{i_0} \neq y_{i_0}$.

Let us put $q' = \beta_0 a_0 \beta_1 a_1 \cdots \beta_{i_0-2} a_{i_0-2} \beta_{i_0-1} a_{i_0-1}$ and $q'' = a_{i_0} \beta_{i_0+1} a_{i_0+1} \cdots \beta_k a_k \beta_{k+1}$. Then we have

$$q = q' \beta_{i_0} q'', \quad p = q' p_{i_0} p'', \text{ for some } p'' \text{ such that } p'' \preceq q''$$
$$\text{and} \qquad p_{i_0} \prec \beta_{i_0} \preceq y_{i_0}, \quad p'' \preceq q'' \preceq a_{i_0} y_{i_0+1} a_{i_0+1} \cdots y_k a_k y_{k+1}$$

Let us consider the β_{i_0} determined at the i_0-th stage in the first *for* loop of the procedure. Since the prefix q' of q is determined at the (i_0-1)-th stage, $q_{i_0-1} = q' y_{i_0} a_{i_0} y_{i_1} a_{i_1} \cdots y_k a_k y_{k+1}$. At the i_0-th stage, the nonempty string ε is first substituted to the variable y_{i_0}. As mentioned above, since $\beta_{i_0} \neq \varepsilon$, $S \not\subseteq L(q_{i_0-1}\{y_{i_0} := \varepsilon\})$. As easily seen, $q'p'' \preceq q'q'' \preceq q_{i_0-1}\{y_{i_0} := \varepsilon\}$ holds, and so $S \not\subseteq L(q'p'')$. It means that $p_{i_0} \neq \varepsilon$. Moreover, since $p_{i_0} \neq y_{i_0}$ as mentioned above, it implies $p_{i_0} \in Z^+$.

Consider the case of $\beta_{i_0} \in Z^+$. Clearly $|p_{i_0}| > |\beta_{i_0}| (= j_0)$. The second *for* loop runs substitutions $\{y_{i_0} := z_{i_0,1} z_{i_0,2} \cdots z_{i_0,j}\}$ for the pattern q_{i_0-1} from $j = m$ down to 1, and fails to substitute for any $j > j_0$. It means $S \not\subseteq L(q_{i_0-1} p_{i_0} a_{i_0} y_{i_0+1} a_{i_0+1} y_{i_0+2} a_{i_0+2} \cdots y_k a_k y_{k+1})$. Hence $L(p) \subseteq L(q_{i_0-1} p_{i_0} a_{i_0} y_{i_0+1} a_{i_0+1} y_{i_0+2} a_{i_0+2} \cdots y_k a_k y_{k+1})$, it leads a contradiction.

The other case is $\beta_{i_0} = y_{i_0}$. In this case, the procedure at the i_0-th stage fails a substitution $\{y_{i_0} := z_{i_0}\}$. It means that $S \not\subseteq L(q' p_{i_0} a_{i_0} y_{i_0+1} \cdots a_k y_{k+1})$ holds because of $p_{i_0} \in Z^+$. This implies that $S \not\subseteq L(p)$, and it is a contradiction.

Finally since $k, m \leq l$ and the inclusion problem for S at each stage is computable in time $O(\sum_{w \in S} |w|)$, the procedure MINL runs in time $O(l^2 \times \sum_{w \in S} |w|)$.

This completes our proof. □

By Theorem 4.1, Lemma 4.3 and Lemma 4.4, we obtain the following:

Theorem 4.5. The class of regular pattern languages is polynomial time inferable from positive examples.

Acknowledgements. I am grateful to Professor Masako Sato and Associate Professor Yasuhito Mukouchi for our fruitful discussions.

References

[1] D. Angluin: *Finding patterns common to a set of strings*, Journal of Computer and System Sciences, **21**, 46-62, (1980).

[2] D. Angluin: *Inductive inference of formal languages from positive data*, Information and Control, **45**, 117-135, (1980).

[3] D. Angluin: *Inference of Reversible Language*, Journal of the Association for Computing Machinery, **29**, 741-765, (1982).

[4] H. Arimura, T. Shinohara and S. Otsuki: *Finding minimal generalizations for unions of pattern languages and its application to inductive inference from positive data*, Lecture Notes in Computer Science, **775**, 646-660, (1994).

[5] E. M. Gold: *Language identification in the limit*, Information and Control, **10**, 447-474, (1967).

[6] T. Moriyama and M. Sato: *Properties of language classes with finite elasticity*, IEICE Transactions on Information and Systems, **E78-D**, (**5**), 532-538, (1995).

[7] T. Motoki, T. Shinohara and K. Wright: *The correct definition of finite elasticity: Corrigendum to identification of unions*, Proceedings of the 4th Annual Workshop on Computational Learning Theory, 375, (1991).

[8] Y. Mukouchi: *Containment problems for pattern languages*, IEICE Transactions on Information and Systems, **E75-D**, (**4**), 420-425, (1992).

[9] M. Sato: *Inductive Inference of Formal Languages*, Bulletin of Information and Cybernetics, **27**, (**1**), 85-106, (1995).

[10] M. Sato, Y. Mukouchi and D. Zheng: *Characteristic sets for unions of regular pattern languages and compactness*, Lecture Notes in Artificial Intelligence, **1501**, 220-233, (1998).

[11] T. Shinohara: *Polynomial time inference of extended regular pattern languages*, Proceedings of RIMS Symposia on Software Science and Engineering, Lecture Notes in Computer Science, **147**, 115-127, (1982).

[12] T. Shinohara: *Polynomial Time Inference of Pattern Languages and Its Application*, Proceedings of the 7th IBM Symposium on Mathematical Foundations of Computer Science, 191-209, (1982).

[13] T. Shinohara and H. Arimura: *Inductive inference of unbounded unions of pattern languages from positive data*, Proceedings of the 7th International Workshop on Algorithmic Learning Theory, Lecture Notes in Artificial Intelligence, **1160**, 256-271, (1996).

[14] K. Shimizu: *A polynomial time algorithm for computing a shortest characteristic sequence and maximal common subsequence of a set of sequences*, Master thesis, University of Osaka, (1993).

[15] K. Wright: *Identification of unions of languages drawn from positive data*, Proceedings of the 2nd Annual Workshop on Computational Learning Theory, 328-333, (1989).

Department of Mathematics and Information Sciences, Gratitude School of Science,
Osaka Prefecture University
1-1 Gakuen-cho, Sakai, Osaka 582-8582, JAPAN