Finding a one-variable pattern from incomplete data

Sakamoto, Hiroshi
Department of Informatics, Kyushu University

出版情報：DOI Technical Report. 149, 1998-05-05. Department of Informatics, Kyushu University
バージョン：
権利関係：

DOI-TR-149

DOI Technical Report

Finding a one-variable pattern from incomplete data

by

Hiroshi Sakamoto

May 5, 1998

Department of Informatics
Kyushu University
Fukuoka 812-81, Japan

Email: hiroshi@i.kyushu-u.ac.jp Phone: +81-92-642-2697

Finding a one-variable pattern from incomplete data

Hiroshi SAKAMOTO
Department of Informatics, Kyushu University, Fukuoka 812-8581, Japan
hiroshi@i.kyushu-u.ac.jp

Abstract

The present paper deals with the problem of finding a consistent one-variable pattern from incomplete positive and negative examples. The studied problems are called an *extension*, a *consistent extension* and a *robust extension*, denoted by E, CE and RE, respectively. The E corresponds an ordinary decision problem of finding a consistent one-variable patterns for given positive and negative examples. On the other hand, an example string is allowed to contain some unsettled symbols that is potentially matching with every constant symbol. For CE, we must settle a suitable assignment for these symbols as well as we must find a one-variable pattern consistent with the examples w.r.t the assignment. The RE requires more universal consistency, i.e., a correct one-variable pattern must be consistent with the examples w.r.t every assignment. The decision problems are closely connected with the learnability of one-variable pattern languages from positive and negative examples. Then, we estimate the computational complexity of the problems.

1 Introduction.

A pattern is a string consisting of constant symbols and variable symbols. The language generated by a pattern π is the set of all strings obtained by substituting strings of constants for the variables of π [1]. During the last decade, a number of researchers have been widely investigated the learnability of pattern languages with respect to many paradigms (cf., e.g., [6, 10, 12, 13]).

Marron [13] considered a learning model in which a single positive example is given. He proposed a greedy learning algorithm and showed that k-variable patterns are learnable using polynomially many membership queries in his model.

Kearns and Pitt [10] investigated the PAC-learnability of k-variable pattern. They gave an algorithm that learn a target pattern by producing a polynomial sized disjunction of patterns from any product distribution. However, it is open whether their result holds for arbitrary distribution.

Lange and Wiehagen [12] studied learning all pattern languages in the limit. They provided a learner that may output inconsistent guesses, but it achieves polynomial

update time. Erlebach et al [6] investigated the learnability of one-variable pattern languages in the limit and they designed very efficient learning algorithm, using the concept of descriptive patterns, that achieves update time $O(n^2 \log n)$, where n is the size of the input sample. In particular, the best known algorithm to compute descriptive one-variable patterns requires $O(n^4 \log n)$ (cf. [1]). The readers should refer to as to other nice results of learnability of pattern languages (cf., e.g., [6, 15, 16]).

This continuous interest in pattern languages has many reasons. One of them is its potential applications (cf., e.g., [3, 14]). In these applications, efficiency of an algorithm becomes central issue. Then, various researchers studied the efficiency of learning algorithms.

On the other hand, another important aspect of pattern languages is its information theoretical interest. Because process of guessing descriptive patterns is a kind of recognition problems in *Formal Language Theory*, this aspect is very closed to the learnability of pattern languages and many attractive results are provided (cf., e.g., [4, 8, 9]).

Along this line of theoretical researches, recently Boros et al formalized the notion of monotone extensions (cf. [5]). Their motivation comes from an observation that real world data are often "monotone", and in such cases it is natural to build a monotone hypothesis.

Given a data, a suitable hypothesis is called an extension. An extension is required to express given data exactly. The problem of finding an extension is denoted by E. Since real world data sometime contains an indefinite value, in this case we should modify the value. There are two kinds of interpretations. One is that we consider an establishment of the value to be critical for our hypothesis. The other is that the value does not influence our hypothesis. Then, they also proposed the corresponding hypothesis spaces called consistent extension CE and robust extension RE, respectively.

Given data, CE is the problem of finding an extension and a modification such that the extension is consistent with the data with respect to the modification, and RE is the problem of finding an extension such that it is consistent with the data with respect to all possible modifications. Other hypothesis spaces for extensions are introduced in [5].

Boros et al assumed the class of monotone Boolean functions as the hypothesis space and analyzed the computational complexity of the above problems (cf. [5]). As was shown in [3], patterns are adaptable for management of real data. Then, we try to expand their protocol into the class of pattern languages. In general, by the intersection theorem in [1], possible hypotheses does not increase even if we get a data successively. Thus, the class of one-variable patterns is considered to be monotone, naturally. In this study we give the formal definitions of E, CE and RE for one-variable patterns and investigate the complexity of them.

2 Preliminaries.

An alphabet is a finite set of symbols, denoted by Σ. By Σ^*, we denote the free monoid over Σ (cf. [7]). The empty string of length zero is denoted by ε. Then, the

set of all non-empty strings is denoted by Σ^+. Let x be a symbol not contained in Σ. Every string $\pi \in (\Sigma \cup \{x\})^+$ is called a *one-variable* pattern and x is referred to as the variable of π. By Pat, we denote the class of one-variable patterns.

Let α and β be strings. By $\sharp(\alpha, \beta)$, we denotes the occurrences of β in α, e.g., $\sharp(abc, b) = 1$ and $\sharp(ababa, ab) = 2$. The length of α is denoted by $|\alpha|$. The i-th symbol of α from the left is denoted by $\alpha[i]$ and $\alpha(i)$ denotes the prefix of α in length i, e.g., $abbca[2] = abbca[3] = b$ and $abbca(3) = abb$. Moreover, for $1 \leq i \leq j \leq |\alpha|$, $\alpha[i,j]$ denotes the substring $\alpha[i]\alpha[i+1]\cdots\alpha[j]$, e.g., $abbca[2,4] = bbc$.

For $\pi \in Pat$ and $u \in \Sigma^+$, we denote by $\pi[x/u]$ the string $w \in \Sigma^+$ obtained by replacing all x in π by u. The string u is called a *substitution* for x of π. For every $\pi \in Pat$, we define the *language* of π by

$$L(\pi) =_{def} \{w \in \Sigma^+ \mid \exists u \in \Sigma^+, \ \ w = \pi[x/u]\}.$$

We assume finite sets of strings, denoted by $\mathcal{P}$ and $\mathcal{N}$, where $\mathcal{P} \cap \mathcal{N} = \emptyset$. A member of $\mathcal{P}$ is called a *positive example* and a member of $\mathcal{N}$ is called a *negative example*. A pattern $\pi \in Pat$ is called *consistent* with a string w if $w \in L(\pi)$ and w is a positive example, or $w \notin L(\pi)$ and w is a negative example. Similarly, π is said to be consistent with $\mathcal{P} \cup \mathcal{N}$ if it is consistent with every member of the set.

Let $\star$ be a special symbol not belonging to $\Sigma \cup \{x\}$. A string $w \in \Sigma \cup \{\star\}^+$ is called *incomplete* if $\sharp(w, \star) \geq 1$. Moreover, $\mathcal{P} \cup \mathcal{N}$ is also called incomplete if at least one member of $\mathcal{P} \cup \mathcal{N}$ is incomplete. For an alphabet Σ, Φ_Σ denotes the set of partial functions $\varphi : \Sigma^+ \times \mathbb{N} \mapsto \Sigma$ such that

$$\varphi(w,i) = \begin{cases} \text{an } a \in \Sigma & \text{if } w[i] = \star \\ w[i] & \text{if } w[i] \in \Sigma. \end{cases}$$

For each string $w \in (\Sigma \cup \{\star\})^+$ and $\varphi \in \Phi_\Sigma$, the string $\varphi(w,1)\varphi(w,2)\cdots\varphi(w,|w|)$ is denoted by $\varphi(w)$, for short.

In [5], the notion of finding monotone extensions from missing bits is introduced in order to find a concept and an assignment for missing bits such that the concept is consistent with given data with respect to the assignment. We apply this notion for our problem of one-variable patterns. Let $\mathcal{P}$ and $\mathcal{N}$ be the form $\mathcal{P} = \{u_i \mid 1 \leq i \leq n\}$ and $\mathcal{N} = \{v_j \mid 1 \leq j \leq m\}$.

Then, for each $\varphi \in \Phi_\Sigma$, by $\varphi(\mathcal{P} \cup \mathcal{N})$, we denote the set

$$\{\varphi(u_i) \mid u_i \in \mathcal{P}\} \cup \{\varphi(v_i) \mid v_i \in \mathcal{N}\}.$$

Definition 1 Given $\mathcal{P}$ and $\mathcal{N}$ such that $\mathcal{P} \cup \mathcal{N} \subset \Sigma^+$, there exists an *extension* for them iff a $\pi \in Pat$ is consistent with all the strings in $\mathcal{P} \cup \mathcal{N}$. On the other hand, given incomplete $\mathcal{P}$ and $\mathcal{N}$, there exists a *consistent extension* for them iff a $\pi \in Pat$ is consistent with $\varphi(\mathcal{P} \cup \mathcal{N})$ for at least one $\varphi \in \Phi_\Sigma$. Furthermore, there exists a *robust extension* for them iff a $\pi \in Pat$ is consistent with $\varphi(\mathcal{P} \cup \mathcal{N})$ for every $\varphi \in \Phi_\Sigma$.

The extension problem is denoted by E, the consistent extension problem is denoted by CE and the robust extension problem is denoted by RE.

3 Comparing the difficulty of E and CE.

In this section, we study the complexity of the ordinary consistency problem E and its extended problem CE for one-variable patterns. For the problem E, given a string in Σ^+ and a one-variable pattern, there is a standard polynomial-time algorithm to check whether the string is generated by the pattern [1]. Then, we first consider the following membership problem of an incomplete string for a one-variable pattern.

Definition 2 For given $w \in (\Sigma \cup \{\star\})^+$ and $\pi \in Pat$, the *existential membership problem*, denoted by $\exists\mathrm{Mem}(\pi, w)$, is accepted if there exists a $\varphi \in \Phi_\Sigma$ such that $\varphi(w) \in L(\pi)$.

Lemma 1 $\exists Mem(\pi, w) \in \mathrm{P}$.

Proof. Let $\pi = w_1 x w_2 x \cdots w_n x w_{n+1}$. Since no erasing is allowed for any $\varphi \in \Sigma$, assuming $\varphi(w) \in L(\pi)$ for some $\varphi \in \Sigma$, we can decide substrings $s_1, s_2, \ldots, s_n$ of w beginning at positions $|w_1|+1, |w_1|+|s_1|+1, \ldots,$ and $\sum_{1\leq i\leq n} |w_i| + (n-1)|s_1|+1$, respectively, where $|s_1| = (|w| - \sum_{1\leq j\leq n+1} |w_j|)/n$. First, we compute the substrings and its length. Let $|s_1| = m$. Then, for each $1 \leq i \leq m$, we compare all $s_1[i], s_2[i], \ldots, s_n[i]$. If $n-1$ or all the symbols are $\star$, then there is a φ for $\varphi(s_1[i]) = \cdots = \varphi(s_n[i])$. Else if all $s_j[i] \in \Sigma$ $(1 \leq j \leq n)$ are equal each other, then such a φ also exists, and there is no such a $\varphi \in \Phi_\Sigma$ otherwise. Once the check is successfully finished, we set a $\varphi' \in \Phi_\Sigma$ such that $\varphi'(s_j) = \varphi(s_j)$ and it assigns all other $n+1$ substrings of w onto $w_1, w_2, \ldots, w_{n+1}$, respectively. Thus, we can decide polynomially whether there exists $\varphi \in \Phi_\Sigma$ such that $\varphi(w) \in L(\pi)$. □

Hence, in the sense of polynomial-time, any incomplete string does not make membership problem difficult. Moreover, this feature does not collapse in more tight class, indeed all the check in Lemma 1 can be done in NC^1. Then, we analyze the efficiency of incomplete strings for more difficult problem CE. The corresponding problem is E. Although E is clearly in NP, it is unknown the hardness for NP[1].

Now, we assume a restriction of CE such that a given $\mathcal{P}$ consists of only grand strings, i.e., $\mathcal{P} = \{u_i \mid 1 \leq i \leq n, u_i \in \Sigma^+\}$. The restricted problem is denoted by RCE. Then, the equivalence of E and RCE for Pat is concluded as follows.

Theorem 1 E is NP-complete iff RCE is NP-complete.

Proof. By Lemma 1, it follows that RCE is in NP. The remained part is the reducibility of E under the assumption that RCE is NP-complete. Let $\mathcal{P} = \{u_i \mid 1 \leq i \leq n, u_i \in \Sigma^+\}$ and $\mathcal{N} = \{v_j \mid 1 \leq j \leq m, v_j \in (\Sigma \cup \{\star\})^+\}$. Then, we set $\mathcal{P}' = \mathcal{P}$ and $\mathcal{N}' = \{v'_j \mid 1 \leq j \leq m, v'_j \in \Sigma^+\}$ such that $v'_j[k] = v_j[k]$ if $v_j[k] \in \Sigma$ and $v'_j[k] = a_j^k$ if $v_j[k] = \star$ for each $1 \leq k \leq |v_j|$, where all a_j^k are not contained in Σ and they are distinct each other.

[1]General pattern consistency is Σ_2^p-complete and regular pattern consistency is NP-complete.

Next, we show the consistency. Assume that there exists $\varphi \in \Phi_\Sigma$ and $\pi \in Pat$ such that π is consistent with $\varphi(\mathcal{P} \cup \mathcal{N})$. Since $\mathcal{P} = \mathcal{P}'$ has no string containing $\star$, the π is consistent with $\mathcal{P}'$. If a $v_j \in \mathcal{N}$ contains no $\star$, then π is also consistent with the corresponding v'_j because $v'_j = v_j$. The critical case is that a string v_j contains at least one $\star$. If a $\star$ appears in a k-th position of v_j such that it is not corresponding to any substitution for x of π, then since $|v_j| = |v'_j|$, the symbol $a^k_j = v'_j[k]$ is not contained in any possible substitution for x, yet. Note that any a^k_j is not in Σ. Thus, $v'_j \notin L(\pi)$. The last case is that all $\star$ are removed by substitutions for x. Since all a^k_j are distinct, clearly $v'_j \notin L(\pi)$.

Conversely, let for any $\pi \in Pat$ and $\varphi \in \Phi_\Sigma$, there exists $v_j \in \mathcal{N}$ such that $v_j \in L(\pi)$. Let the v_j be of $w\alpha w'$, where $w, w' \in \Sigma^*$ and $\sharp(\alpha, \star) \geq 1$. If a π contains two or more variables, one of them must correspond to a substring of v_j containing $\star$. Then, by selecting suitable φ, we can avoid v_j. This is a contradiction. Thus, each π must be of the form wxw'. Hence, $v'_j \in L(\pi)$. The converse reduction is trivial. Therefore, we obtain the result. □

From this Theorem, it seems that incomplete strings as negative examples causes no efficiency for consistency problem. However, it is open whether there is a gap between RCE and CE. Moreover, there is a chance that E and RCE are members in inside of NP, e.g., NP ∩ co-NP, randomized P and P.

4 Analyzing the difficulty of RE.

The aim of this section is to show the NP-completeness of another problem RE. Then, we begin with the membership problem closed to RE defined as follows. Similarly to Definition 2, we assume $w \in (\Sigma \cup \{\star\})^+$ and $\pi \in Pat$.

Definition 3 The *universal membership problem*, denoted by $\forall\mathrm{Mem}(\pi, w)$, is accepted if $\varphi(w) \in L(\pi)$ for all $\varphi \in \Phi_\Sigma$.

Lemma 2 $\forall Mem(\pi, w) \in$ P.

Proof. Assume that $\varphi(w) \in L(\pi)$ for every $\varphi \in \Phi_\Sigma$. Let s is the substitution for x. If $w[i] = \star$ and it is not contained in s, then we can take other $\varphi' \in \Phi_\Sigma$ such that $\varphi'(w[i]) \neq \varphi(w[i])$ and $\varphi'(w[j]) = \varphi(w[j])$ for all $j \neq i$. It follows that $\varphi'(w) \notin L(\pi)$. Thus, every $\star$ must be contained in the substitution s. In this case, if $\sharp(\pi, x) \geq 2$, then we can lead a contradiction analogously. Hence, the required π and w must be of $\pi = w_1 x w_2$ and $w = w_1 w' w_2$, where $w_1, w_2 \in \Sigma^*$ and $\sharp(w', \star) \geq 1$. Conversely, if π and w are of these forms, for all $\varphi \in \Phi_\Sigma$, $\varphi(w) \in L(\pi)$. □

We recall the problem of the robust extension (RE) for one-variable patterns in Definition 1. This problem requires the universal consistency of a one-variable pattern for every $\varphi \in \Phi_\Sigma$. Now, we give a log-space reduction from the 3-SAT to RE below.

Lemma 3 RE is log-space reducible from 3-SAT.

Proof. Let $C = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ be a 3-CNF of n variables $x_1, x_2, \ldots, x_n$ such that $C_i = (\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$, where each ℓ_{i_j} denotes a positive or negative literal of x_{i_j}, i.e., $\ell_{i_j} \in \{x_{i_j}, \bar{x}_{i_j}\}$, $1 \leq i \leq n$ and $j \in \{1,2,3\}$. Let us set an alphabet consisting of $n+3$ symbols such that $\Sigma = \{a_1, a_2, \ldots, a_{n+1}\} \cup \{A, B\}$. First, we compute the strings α_1, α_2, α_3 and α_4 such that

$\alpha_1 = a_1 A^2 a_2 B a_n A^2 a_{n+1}$,

$\alpha_2 = a_1 A a_2 A a_3 \cdots a_n A a_{n+1}$,

$\alpha_3 = a_1 A^2 a_2 A^2 a_3 \cdots a_n A^2 a_{n+1}$ and

$\alpha_4 = a_1 A^3 a_2 A^3 a_3 \cdots a_n A^3 a_{n+1}$.

Next, for each $C_i = (\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$, we compute a string $\beta_i = a_1 \gamma_1 a_2 \gamma_2 a_3 \cdots a_n \gamma_n a_{n+1}$ such that for each $1 \leq j \leq n$, $\gamma_j = BA$ if $x_j \in \{\ell_{i_1}, \ell_{i_2}, \ell_{i_3}\}$, $\gamma_j = AB$ if $\bar{x}_j \in \{\ell_{i_1}, \ell_{i_2}, \ell_{i_3}\}$ and $\gamma_j = \star\star$ otherwise. Finally, we output $\mathcal{P} = \{(\alpha_3, 1), (\alpha_4, 1)\}$ as the positive examples and $\mathcal{N} = \{(\alpha_1, 0), (\alpha_2, 0), (\beta_k, 0) \mid 1 \leq k \leq m\}$ as the negative examples.

Claim: Each one-variable pattern π consistent with $(\alpha_1, 0), (\alpha_2, 0), (\alpha_3, 1)$ and $(\alpha_4, 1)$ must be of the form $\pi = a_1 X_1 a_2 X_2 a_3 \cdots a_n X_n a_{n+1}$, where $X_i \in \{Ax, xA\}$ and $1 \leq i \leq n$.

We first prove this claim. Suppose to the contrary that there exists a substitution other than $[x/A]$ for π that is consistent with the examples. Then, we can consider the following cases.

Case 1. A substitution contains $a_i \in \{a_1, a_2, \ldots, a_{n+1}\}$. In this case, π must be of the form wxw' such that $w \in \{a_1, a_1 A, a_1 AA\}$ and $w' \in \{a_{n+1}, A a_{n+1}, AA a_{n+1}\}$. For each possible w and w', π is inconsistent with the negative example $(\alpha_1, 0)$.

Case 2. The substitution is $[x/AA]$. In this case, the pattern π must be of the form $a_1 X_1 a_2 X_2 a_3 \cdots a_n X_n a_{n+1}$, where $X_i \in \{AA, x\}$ for each $1 \leq i \leq n$. If $X_j = AA$ for some $1 \leq j \leq n$, then π must contains $a_j AA a_{j+1}$ as its a substring. Since $\sharp(\alpha_4, a_j AA a_{j+1}) = 0$, it is a contradiction. Thus, in this case, π must be of the form $a_1 x a_2 x a_3 \cdots a_n x a_{n+1}$. This pattern is inconsistent with the negative example $(\alpha_2, 0)$.

Case 3. The substitution is $[x/A]$. Then, $\pi = a_1 X_1 a_2 X_2 a_3 \cdots a_n X_n a_{n+1}$, where $X_i \in \{Ax, xA, AA\}$ for each $1 \leq i \leq n$. If $X_j = AA$ for some $1 \leq j \leq n$, π contains $a_j AA a_{j+1}$ as its substring, that is a contradiction for $(\alpha_4, 1)$. Thus, the claim is true.

In the further discussion, we denote

$$PI = \{a_1 X_1 a_2 X_2 a_3 \cdots a_n X_n a_{n+1} \mid X_i \in \{Ax, xA\}, 1 \leq i \leq n\}.$$

By this claim, the remained part of the proof is reduced to show that the 3-CNF C is satisfiable iff there exists a $\pi \in PI$ such that $\varphi(\beta_j) \notin L(\pi)$ for all $1 \leq j \leq m$ and $\varphi \in \Phi_\Sigma$.

Assume that the CNF C is satisfiable, i.e., there exists a truth assignment $f : \{x_1, x_2, \ldots, x_n\} \mapsto \{0,1\}$ such that for each clause $C_i = (\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$, an $\ell_j \in \{\ell_{i_1}, \ell_{i_2}, \ell_{i_3}\}$ fulfills exactly one of the following conditions.

Case I. $\ell_j = x_j$ and $f(x_j) = 1$, or

Case II. $\ell_j = \bar{x}_j$ and $f(x_j) = 0$,

where $1 \leq j \leq n$. On the other hand, there is a corresponding $\pi \in PI$ such that $X_j = Ax$ if $f(x_j) = 1$ and $X_j = xA$ if $f(x_j) = 0$.

Recall the construction of γ_i $(1 \leq i \leq m)$. Each γ_i is of $a_1Y_1a_2Y_2a_3 \cdots a_nY_na_{n+1}$, where $Y_j = BA$ if $x_j \in \{\ell_{i_1}, \ell_{i_2}, \ell_{i_3}\}$, $Y_j = AB$ if $\bar{x}_j \in \{\ell_{i_1}, \ell_{i_2}, \ell_{i_3}\}$ and $Y_j = \star\star$ otherwise.

In *Case I*, π and γ_i are of the following forms:

$\pi = a_1X_1a_2X_2a_3 \cdots a_kAxa_{k+1} \cdots a_nX_na_{n+1}$ and

$\gamma_i = a_1Y_1a_2Y_2a_3 \cdots a_kBAa_{k+1} \cdots a_nY_na_{n+1}$.

We note that $|\pi| = |\gamma_i|$ for each $1 \leq i \leq m$. Then, $pi[x/w] \neq \varphi(\gamma_i)$ for any substitution $[x/w]$ and $\varphi \in \Phi_\Sigma$. Similarly in *Case II*, $\pi[3k-2, 3k+1] = xA$ and $\gamma_i[3k-2, 3k+1] = AB$. Thus, there is no chance for $\varphi(\gamma_i) \in L(\pi)$.

Conversely, let the CNF C be unsatisfiable. Then, for each $\pi \in PI$, there exists γ_i corresponding to the clause $C_i = (\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$ such that

$\pi = a_1X_1a_2X_2a_3 \cdots a_nX_na_{n+1}$,

$\gamma_i = a_1Y_1a_2Y_2a_3 \cdots a_nY_na_{n+1}$ and

for each $j \in \{i_1, i_2, i_3\}$ $Y_j = AB$ if $X_j = Ax$ and $Y_j = BA$ if $X_j = xA$.

We set φ for γ_i by

$$\varphi(\gamma_i, k) = \begin{cases} \pi[k]; & \text{if } \pi[k] \in \{a_1, a_2, \ldots, a_{n+1}\} \cup \{A\}, \\ B & \text{if } \pi[k] = x. \end{cases}$$

It follows $\varphi(\gamma_i) \in L(\pi)$ for the substitution $[x/B]$. Hence, the CNF C is satisfiable iff there exists a $\pi \in Pat$ such that for all $\varphi \in \Phi_\Sigma$, π is consistent with $\varphi(\mathcal{P})$ and $\varphi(\mathcal{N})$. □

Theorem 2 RE is NP-complete, even if $||\Sigma|| = 3$.

Proof. For the length n of a shortest positive example, by enumerating all possible $\pi \in Pat$ at most length n nondeterministically, we can check the universality of π for all $\varphi \in \Phi_\Sigma$ in a polynomial time. Then, by Lemma 3, the NP-completeness is concluded. Moreover, even if we restrict an alphabet to be $\Sigma = \{a, A, B\}$, the reduction preserves the consistency for 3-SAT. □

It is an open question whether the above hardness are concluded even if $||\Sigma|| = 2$. In this reduction, there exist four possible substitutions Ax, xA, xx and AA. Can we remove only the last two cases by using no dividing symbol?

There are other related works on one-variable patterns not solved in this paper. In the problems E, CE and RE, if no correct pattern is found, then our answer is just 'no'. However, in this case, we often need an approximate hypothesis. Thus, one is the problem of finding an optimal hypothesis for the given data. For the formalization of the problem, we would need the average-case analysis technique in [6].

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. Queries and concept learning. *Machine Learning*, 2:319-342, 1988.

[3] S. Arikawa, S. Miyano, A. Shinohara, S. Kuhara, Y. Mukouchi and T. Shinohara. A machine discovery from amino acid sequences by decision trees over regular patterns. *New Generation Computing*, 11:361-375, 1993.

[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. In *Proc. STACS'94*, LNCS 775, pp. 649-660, 1994. Springer-Verlag.

[5] E. Boros, T. Ibaraki and K. Makino. Monotone extensions of Boolean data sets. In *Proc. 8th International Workshop on Algorithmic Learning Theory*, LNAI 1316, pp. 161-175, Berlin, 1997. Springer-Verlag.

[6] T. Erlebach, P. Rossmanith, H. Stadtherr, A. Steger and T. Zeugmann. Learning one-variable pattern languages very efficiently on average, in parallel, and by asking queries. In *Proc. 8th International Workshop on Algorithmic Learning Theory*, LNAI 1316, pp. 260-276, Berlin, 1997. Springer-Verlag.

[7] J.E. Hopcroft and J.D. Ullman. Introduction to automata theory, languages, and computation. Addison-Wesley Publ., 1979.

[8] H. Ishizaka, H. Arimura and T. Shinohara. Finding tree patterns consistent with positive and negative examples using queries. In *Proc. 5th International Workshop on Algorithmic Learning Theory*, LNAI 872, pp. 317-332, Berlin, 1994. Springer-Verlag.

[9] T. Jiang, A. Salomaa, K. Salomaa and S. Yu. Inclusion is undecidable for pattern languages. In *Proc. 20th ICALP*, LNCS 700, pp. 301-312, Berlin, 1993. Springer-Verlag.

[10] M. Kearns and L. Pitt. A polynomial-time algorithm for k-variable pattern languages from examples. In *Proc. 2nd Ann. ACM Workshop on Computational Learning Theory*, pp. 57-71, Morgan Kaufmann Publ., San Mateo, 1989.

[11] C. H. Papadimitriou. Computational complexity. Addison-Wesley Publ., 1994.

[12] S. Lange and R. Wiehagen. Polynomial-time inference of arbitrary pattern languages. *New Generation Computing*, 8:361-370, 1991.

[13] A. Marron. Learning pattern languages from a single initial example and from queries. In *Proc. 1st Ann. Conference on Computational Learning Theory*, pp. 311-325, 1988.

[14] T. Shinohara and S. Arikawa. Pattern inference. In *Algorithmic Learning for Knowledge-Based Systems*, LNAI 961, pp. 259-291, Berlin, 1995. Springer-Verlag.

[15] T. Shinohara and H. Arimura. Inductive inference of unbounded unions of pattern languages from positive data. In *Proc. 7th International Workshop on Algorithmic Learning Theory*, LNAI 1160, pp. 257-271, Berlin, 1996. Springer-Verlag.

[16] T. Zeugmann. Learning 1-variable pattern languages in linear average time. In *Proc. 11st Ann. Conference on Computational Learning Theory*, 1998, to appear.