Minimal Multiple Generalization for Unions of Pattern languages

Arimura, Hiroki
Department of Artificial Intelligence, Kyushu Institute of Technology

Shinohara, Takeshi
Department of Artificial Intelligence, Kyushu Institute of Technology

Otsuki, Setsuko
Department of Artificial Intelligence, Kyushu Institute of Technology

出版情報：RIFIS Technical Report. 71, 1993-05-14. Research Institute of Fundamental Information Science, Kyushu University
バージョン：
権利関係：

RIFIS-TR-CS-71

RIFIS Technical Report

Minimal Multiple Generalization for Unions of Pattern languages

Hiroki Arimura
Takeshi Shinohara
Setsuko Otsuki

May 14, 1993

Research Institute of Fundamental Information Science
Kyushu University 33
Fukuoka 812, Japan

E-mail: arim@ai.kyutech.ac.jp Phone: 0948-29-7638

Minimal Multiple Generalization for Unions of Pattern Languages

Hiroki Arimura **Takeshi Shinohara** **Setsuko Otsuki**

Department of Artificial Intelligence
Kyushu Institute of Technology
Kawazu 680-4, Iizuka 820, JAPAN

連絡先：
有村博紀: 郵便番号 820, 飯塚市川津 680-4,
九州工業大学 情報工学部 知能情報工学教室
tel: (0948)29-7638
fax: (0948)29-7601
email: arim@ai.kyutech.ac.jp

Abstract

The k-minimal multiple generalization (k-mmg) is a natural extension of the least generalization (lg) given by Plotkin in 1970. The k-mmg generalizes given first order terms by at most k-terms, while the lg does by a single term. Thus, k-mmg gives a more precise approximation of a given set of examples. In this paper, we extend the algorithm for a more abstract class of objects by abstracting a generalization structure of first-order terms. We present a general design of a polynomial time k-mmg algorithm for the wider classes of objects, and prove the correctness. Using the algorithm, we prove the polynomial time inferability from positive data of unions of at most k languages in a subclass of pattern languages. One class is the class of one-variable pattern languages, and another is the class of regular pattern languages with a bounded number of variables. We also discuss the use of refinement operator and NC-learnability from positive data.

1 Introduction

The k-minimal multiple generalization (k-mmg) is a natural extension of the least generalization given by Plotkin in 1970. The k-minimal multiple generalization generalizes given first order terms by at most k-terms, while the least generalization does by a single term. Thus, k-mmg gives a more precise approximation of a given set of first-order terms.

In this paper, we extend the algorithm for a more abstract class of objects with a generalization structure. It enables us to apply multiple generalization algorithm to several classes of objects known in computer science such as pattern languages in inductive inference, complex database objects in database theory, and feature structures in computational linguistics.

We present a general design and the crrectness of a polynomial time k-mmg algorithm for such a class provided a least (or a minimal) single generalization algorithm. We also discuss the use of refinement operators for efficient computation.

Applying the algorithm, we show that two classes of unions of pattern languages are polynomial time identifiable in the limit by computing a minimal language. The first is the class of unions of at most k one-variable pattern languages, and the second is the class of unions of at most k regular pattern languages with a bounded number of variables. The result for the first class gives a positive answer to the open problem posed by K. Wright in 1989. For the second class, we give an NC algorithm to solve the problem.

The paper consists of the following sections. Section 2 gives basic definitions and a result on inductive inference. In Section 3, we review the properties of several generalization structure introducing the notion of generalization systems. Using this notion, we formalize multiple generalizations in Section 3. In Section 4, we present a design of a polynomial time algorithm MMG that computes a k-mmg of a given finite set of objects in an abstract style, prove the correctness, and apply it to inference of unions of one-variable pattern languages. In Section 6, we see that the use of refinement operators makes the construction of the algorithm significantly simple, and applies it to inference of unions of a subclass of regular pattern languages.

2 Pattern languages and identification in the limit

In this section, we introduce basic definitions on languages and structured objects, and give a brief survey of inductive inference.

For a set A, we denote by A^* the set of finite strings over A, by A^+ the set of nonempty finite strings over A, by Pow$A$ the *powerset* of A, the set $\{ B \mid B \subseteq A\}$ of all subsets of A, and by $|A|$ the number of elements in A. For a string w, $|w|$ denotes the length of w. If A is a finite set of strings $w_1, \ldots, w_n$, $||S||$ is the total length of strings in A, $|w_1| + \cdots + |w_n|$. For $a \in A$, $A \backslash a$ denotes the set $A - \{a\}$.

Let Σ be a finite alphabet and X be a set of variables such that $\Sigma \cap X = \emptyset$. A *pattern* is a nonempty string p over $(\Sigma \cup X)^+$. A *substitution* to p is a replacement of the form $\theta(x_i) = p_i$ by $\{x_1 := p_1, \ldots, x_n := p_n\}$, where $x_i \in v(p)$. We denote by $p\theta$ the pattern obtained by replacing the variables $x_1, \ldots, x_n$ with $p_1, \ldots, p_n$, respectively. The set $L(p)$ is the set of strings $\{\theta(p) \in \Sigma^+ \mid \theta$ is a substitution$\}$. If $L = L(p)$ for some pattern p, then L is a *pattern language*. For example, $p = xxbyb$ is a pattern over $\Sigma = \{a, b\}$ and p defines the language $\{aabab, bbbab, ababbabb, \ldots\} = \{ uubwb \mid u, w \in \Sigma^+\}$. A *one-variable pattern* is a pattern p such that at most one variables occurs in p [Ang80]. A *regular pattern* is a pattern p such that any variable occurs at most once in p [Shi82]. For example, $axxbxa$ is a one-variable pattern and $abxabyzb$ is a regular pattern. We denote by $\mathrm{P}, \mathrm{P}_1, \mathrm{P}_{\mathrm{reg}}$ the class of patterns, one-variable patterns, regular patterns, and by $\mathrm{PL}, \mathrm{PL}_1, \mathrm{PL}_{\mathrm{reg}}$ the class of the class of pattern languages, one-variable pattern langages, regular pattern languages, respectively.

Next, we introduce basic definitions on inductive inference according to [Ang80]. Let Π and Γ be alphabets for concepts and descriptions, respectively.

An *indexed family of recursive languages* or a *concept class* is a triple $C(C, D, L(\cdot))$, where $C \subseteq \mathrm{Pow}(\Pi^*)$ is the class of languages or concepts, $D \subseteq \Gamma^*$ is a recursively enumerable set of descriptions, and $L(\cdot)$ is a mapping from D to C such that $C = \{ L(p) \mid p \in D\}$. We call $L(p)$ the language or concept defined by p.

The set $U = \cup C$ is called the *universe* of words or of objects. We assume an algorithm that, given $w \in U_D$ and $p \in D$, effectively determines whether $w \in L(p)$. Sometimes, we do not distinguish $\mathcal{C}$ and C.

In this paper, we denote by $p, q, p_1, \ldots$ descriptions, by $P, Q, \ldots$ sets of descriptions, and by $L, \ldots$ languages. An example of a concept class is the class $(\mathrm{PL}, \mathrm{P}, L(\cdot))$ of pattern languages.

The inductive inference problem we study is *identification in the limit from positive data.* Let $\mathcal{C} = (C, D, L(\cdot))$ be a concept class. *Examples* are strings extracted from an unknown language $L_* \in C$. A *polynomial time inference machine* is a polynomial time algorithm M that for each stage $n = 0, 1, \ldots$, requests a new example $w_n \in L_*$ $(n \geq 0)$, computes a guess $g_n \in D$ from the set $S = \{w_0, \ldots, w_n\}$ of current examples in polynomial time in $||S||$, and outputs g_n. The concept class $\mathcal{C}$ is said to be *identifiable in the limit from positive data with consistent and conservative polynomial time updating* if there is a polynomial time inference machine such that

(1) For any $L_* \in C$ and any infinite sequence $w_0, w_1, \ldots$ such that $\{ w_n \mid n \geq 0\} = L_*$, the infinite sequence of guesses converges to some $p \in D$ with $L(p) = L_*$, and

(2) M outputs only a consistent guess g_n with S_n, that is, a guess satisfying $S_n \subseteq L(g_n)$, and whenever a guess g_n is consistent with S_n, M does not change the guess g_n, that is, $g_{n+1} = g_n$.

The next is the only lemma on inductive inferability that we use to show the inferability of classes of unions in the paper. This states rather stronger types of inferability, which is ideal for practical applications.

Lemma 1 ([Ang80, ASO93]) *If a concept class $\mathcal{C}$ has finite thickness (further, finite elasticity [Wri89]), and both search problems the membership problem of languages and the problem to compute a minimal language containing a given sample within $\mathcal{C}$ are polynomial time computable, then $\mathcal{C}$ is identifiable in the limit from positive data with consistent and conservative polynomial time updating, where $\mathcal{C}$ is of finite thickness if for any $w \in U$, the set $\{ L \in C \mid w \in L\}$ is finite.*

3 Generalization

Here, we introduce a special kind of concept classes, called generalization systems to capture properties common in most of systems which have an efficient generalization procedure.

Let $(D, \leq)$ be a set of descriptions partially ordered with a relation $\leq$. We may write D for $(D, \leq)$. If $p \leq q$ and $q \leq p$ then $p \equiv q$. If $\leq$ is only transitive, but is not partially ordered, we make a partial ordered set $(D_*, \leq)$ by taking the set D_* of representatives of the equivalent classes of D modulo $\equiv$. If $p \leq q$ but $q \not\leq p$ then we write $p < q$. If $p \leq q$ then we say p is a *refinement* of q or q is a *generalization* of p. We may say an *instance* for a refinement.

Definition 1 Let A be a subset of D. If $q \leq p$ for all $q \in A$, then p is a *common generalization* of A. If p is a common generalization of A and there is no common generalization q of A such that $q < p$, then p is a *minimal common generalization* (mcg) of A.

A *sample* is a finite set $S \subseteq U$. For a sample S, if $S \subseteq L(p)$, then we say p *covers* S, and p is a *covering* of S. In this case, it is equivalent to that p is a common generalization of S.

Now, we give a definition of generalization systems to capture concept classes where test of membership $w \in L(q)$ and computation of an mcg are efficiently computable. The condition on the size reflects the property that an instance is created from another description by replacing a component with a "larger" one.

Definition 2 A *generalization system* is a concept class $(C, D, L(\cdot))$ defined as follows.

1. The set $(D, \leq)$ is a partially ordered set with the greatest element $\top$.

2. We define the universe of words as the set $U = \{ w \in D \mid w$ is a minimal in D w.r.t. $\leq\}$. If p is in U, p is called ground. $L(\cdot)$ is defined as $L(p) = \{ w \in U \mid w \leq p\}$.

3. There is a totally computable function $size : D \to N$ satisfying that for any p and q in D, if $p \leq q$ then $size(p) \leq size(q)$.

Definition 3 A generalization system $\mathcal{C} = (C, D, L(\cdot))$ are *efficient* if it satisfies the following conditions.

1. The relation $\cdot \leq \cdot$ is polynomial time decidable.
2. For "almost all" $p \in D$, the size $size(p)$ and the length $|p|$ of p are bounded by two polynomials $h(\cdot)$ and $h'(\cdot)$ each other (we say they are polynomially related).
3. There is an algorithm that, given a finite set $S \subseteq U$, computes an mcg of S in polynomial time in $||S||$.

Lemma 2 *Let $\mathcal{C} = (C, D, L(\cdot))$ be an efficient generalization system. Then, the following properties hold.*

1. *The system has finite thickness.*
2. *For any p and q in D, $p \leq q \Rightarrow L(p) \subseteq L(q)$.*
3. *The membership "$w \in L(p)$?" is polynomial time decidable.*

Proof. Finite thickness is derived from 2 in Definition 3, and a fact that for any $n > 0$, the number of distinct strings of length at most n is finite. □

The inverse direction $\Leftarrow$ for Property 2 in the lemma does not hold in general. Consider the following condition. Two generalization systems we will study in the paper have the condition.

Condition 1 For any p and q in D, $p \leq q \Leftarrow L(p) \subseteq L(q)$.

The next lemma is not new; it has been known for each particular classes of pattern languages and first-order terms [Ang80, LMM88, Wri89].

Lemma 3 *For a generalization system $\mathcal{C} = (C, D, L(\cdot))$, if it is efficient and it $\mathcal{C}$ satisfies Condition 1, then $\mathcal{C}$ is identifiable in the limit from positive data with consistent and conservative polynomial time updating.*

Fact 1 We give examples of generalization systems.

1. The following concept classes are efficient generalization systems: (1) One-variable patterns [Ang80]. (2) Regular patterns [Shi82]. (3) First-order terms (Tree patterns) [Plo70, Rey70]. (4) Complex database objects without set expression, or hierarchical tuples in relational databases [BK89, BJO91] (That are same as most simple kind of feature structures).

2. The following concept classes are generalization systems, but not efficient: (5) Patterns if $P \neq NP$ (Because, the membership decision is NP-complete [Ang80]). (6) Complex database objects with set expression [BK89, BJO91] (Because, it does not satisfy the condition for $size(\cdot)$; Moreover, it has no finite thickness).

4 Multiple generalization and inference of unions from positive data

Let $\mathcal{C} = (C, D, L(\cdot))$ be a generalization system with a partial ordering $(D, \leq)$. A *multiple description* is a subset $P \in \mathrm{Pow}D$ of D. We define the set $L_\cup(P) = \cup\{ L(p) \mid p \in P\}$. We extend a partial ordering $\leq$ on D for $\sqsubseteq$ on $\mathrm{Pow}D$.

Definition 4 (Hoare powerset ordering induced by $\leq$ [BJO91]) Let P, Q be subsets of D. Then, $P \sqsubseteq Q$ if for all $p \in P$, there exists $q \in Q$ such that $p \leq q$. If $P \sqsubseteq Q$ and $Q \sqsubseteq P$, then $P \equiv Q$. If $P \sqsubseteq Q$ but $Q \not\sqsubseteq P$, then $P \sqsubset Q$.

The *canonical version* of P is the set $P_* = \{p \in P |$ there is no $q \in Q$ such that $p > q\}$. Then, $P_* \equiv P$. Therefore, we can take the set $(\mathrm{Pow}D)_* = \{ P_* \mid P \in \mathrm{Pow}D\}$ to make $(\mathrm{Pow}D, \sqsubseteq)$ a partial order. Hereafter, we write D for D_*.

Next, we introduce k-multiple generalizations. In particular, we restrict our attention to multiple descriptions with a bounded number of members. Let $k \geq 1$. For D and $C \subseteq \mathrm{Pow}U$, we denote by D^k the set $\{ P \subseteq D \,|\, |D| \leq k\}$ and by C^k a class $C^k = \{ L_1 \cup \cdots \cup L_l \,|\, L_1, \ldots, L_l \in C, 1 \leq l \leq k\}$. We call an element P of D^k a *k-multiple description*. Then, $C^k = \{ L_\cup(P) \,|\, P \in D^k\}$. For example, $\mathrm{P_{reg}}^2$ is the class of pairs of regular patterns and $\mathrm{PL_{reg}}^2$ is the class of languages defined by pairs in $\mathrm{PL_{reg}}^2$. Using D^k and $L_\cup(P)$, we define concept classes for multiple descriptions.

Definition 5 Let $\mathcal{C} = (C, D, L(\cdot))$ be a generalization system and $k \geq 0$. A *k-multiple generalization system induced by $\mathcal{C}$* is a concept class $\mathcal{C}^k = (C^k, D^k, L_\cup(\cdot))$. We may write $L(\cdot)$ for $L_\cup(\cdot)$ if no confusion arises.

Definition 6 Let S be a set of ground descriptions, a sample, $k \geq 1$, and P, Q be k-multiple descriptions. If $S \subseteq L(P)$, then we say P is a *k-multiple covering* of S, or P covers S. If P covers S for $P \in D^k$, and there is no $Q \sqsubset P$ such that $Q \in D^k$ and Q covers S, then we say P is a *k-minimal multiple generalization* (k-mmg) of S.

Note that if we do not restrict the number k to a fixed constant, for all sample S there is a trivial minimal multiple generalization of S, the set S itself. The k-multiple generalization system $\mathcal{C}^k$ has the following properties inherited from $\mathcal{C}$

Lemma 4 Let $k \geq 0$, $\mathcal{C}$ be a generalization system, and $\mathcal{C}^k$ be a k-multiple generalization system induced by $\mathcal{C}$. If $\mathcal{C}$ is efficient, then $\mathcal{C}^k$ satisfies the following properties 1 – 4.

1. The set $(D^k, \sqsubseteq)$ is a partially ordered set with the greatest element $\{\top\}$. The relation $\cdot \leq \cdot$ is polynomial time decidable.
2. The membership decision for $L_\cup(\cdot)$ is polynomial time decidable.
3. For any P and Q in D^k, $P \sqsubseteq Q \;\Rightarrow\; L(P) \subseteq L(Q)$.
4. There is an effective algorithm that, given a finite set $S \subseteq U$, computes an mcg of S, although it may need exponential time.
5. The class $\mathcal{C}^k$ has finite elasticity, while it has no finite thickness.

Proof. Properties 1, 2 and 3 are immediate from the definition. In [Wri89], Wright proved that finite thickness implies finite elasticity and that finite elasticity is closed under language union. Thus, the finite thickness of $\mathcal{C}$ shows Property 5. For $\mathcal{C}^k$, we can show that if $P = \{p_1, \ldots, p_m\}$ $(m \leq k)$ is a k-mmg of a sample S, then there is a partition $S_1 + \cdots + S_m = S$ of S such that for all i, p_i is an mcg of S_i (Derived from Lemma 6). Thus, an algorithm can compute a k-mmg by enumerating all such $P \in D^k$ and by checking for each P whether it is minimal. It is effectively computable by the condition of $size(\cdot)$ in Definition 3. □

The converse of Property 3 does not hold in general. We introduce a condition that includes Condition 1

Condition 2 (Compactness with respect to containment) Let C^k be a class of languages for $k \geq 0$. Then, for any $L, M_1, \ldots, M_l \in C$ $(1 \leq m \leq k)$,

$$L \subseteq M_1 \cup \cdots \cup M_m \;\Rightarrow\; L \subseteq M_i \text{ for some } 1 \leq i \leq m.$$

The following theorem states that generalization and inference from positive data are closely related in $\mathcal{C}^k$. In later sections, we will show concept classes that satisfy Condition 2.

Theorem 5 Let $k \geq 0$ and $\mathcal{C}$ be an efficient generalization system, and let $\mathcal{C}^k$ be a k-multiple generalization system induced by $\mathcal{C}$. If $\mathcal{C}$ satisfies Condition 1, Condition 2, and for $\mathcal{C}^k$, there is an algorithm that computes a k-mmg in polynomial time, then $\mathcal{C}^k$ is identifiable in the limit from positive data with consistent and conservative polynomial time updating.

Proof. By Condition 1 and Condition 2, two relations $\cdot \sqsubseteq \cdot$ and $L(\cdot) \subseteq L(\cdot)$ coincide. From the argument above, the class $\mathcal{C}^k$ has finite elasticity and there is a polynomial time algorithm to compute a minimal language containing a sample. Hence, the result is immediate from Lemma 4. □

5 Polynomial time k-mmg algorithm

In this section, we investigate general sufficient conditions for the system has an algorithm to efficiently compute k-mmg of a sample, and using the result, we show the polynomial time inferability of the class $(\mathrm{PL}_1)^k$ of unions of at most k one-variable pattern languages.

As seen in the previous section, any k-multiple generalization system $\mathcal{C}^k$ induced by an efficient $\mathcal{C}$ has a k-mmg algorithm. However, it is not efficient. The reason of inefficiency of the method is that the algorithm enumerates exponentially many all the partitions of S into k subsets. Our polynomial time k-mmg algorithm works in quite different way. Starting from most general approximation, the algorithm proceeds from general to more specific candidates.

Before explaining our algorithm, we have to state two search problems for $\mathcal{C}$ that our algorithm uses as subprocedures. For a multiple covering P of S, if $S \not\subseteq L(P\backslash p)$ for all p, then we say P is *reduced* with respect to S. Let p be a covering of S. Then, a *k-division* of p with respect to S is a k-multiple covering P of S such that (1) $|P| > 1$; (2) each components are properly specific, $P \sqsubset \{p\}$; and (3) P is reduced with respect to S. If for some $p \in P$, there is a k-division of p with respect to S, then we call p *k-divisible* with respect to S.

We give an example of a k-division on first-order terms. Consider a covering $p = f(x, x)$ of the sample,

$$S = \left\{ \begin{array}{l} f(g_1(a), g_1(a)),\ f(g_2(a), g_2(a)),\ f(g_3(a), g_3(a)), \\ f(g_1(b), g_1(b)),\ f(g_2(b), g_2(b)),\ f(g_3(b), g_3(b)) \end{array} \right\}.$$

Then, $Q = \{f(g_1(x_1), g_1(x_1)), f(g_2(x_2), g_2(x_2)), f(g_3(x_3), g_3(x_3))\}$ is a 3-division of p with respect to S. Q represents a more specific concept than P.

Let S be a sample. If $\emptyset \subsetneq C \subsetneq S$ for a subset $C \subseteq S$ holds, then C is a *partial covering* of S. A set A of partial coverings of S is *complete* with respect to S if for any partial covering B of S, there is a partial covering C in A such that $B \subseteq C$. For a set A of partial covering of S and a set P of descriptions, if $A = \{ S \cap L(p) \mid p \in P \}$, then we say P *defines* A with respect to S. In this chapter, we assume that the concept class $\mathcal{C}^k$ satisfies the following conditions.

Condition 3 The following two search problems are polynomial time computable.

> COMPLETE SET OF PARTIAL COVERS for $\mathcal{C}$ ($\mathrm{CPC}_{\mathcal{C}}$):
> Instance: a sample S and an mcg p of S.
> Question: find a set P of refinements of p that defines, with respect to S, a complete set A of partial coverings of S if it exists. Otherwise, output NO.
>
> MAXIMALLY SPECIFIC CONSISTENT REFINEMENT for $\mathcal{C}$ ($\mathrm{MSCR}_{\mathcal{C}}$):
> Instance: a sample S and a cover p of S.
> Question: find a maximally specific consistent refinement (mscr) of p with S, that is, a description $q \leq p$ which is an mcg of S.

We denote by $CPC(p, S)$ and by $MSCR(p, S)$, respectively, an answer that algorithms for the problems return.

The essential idea of our k-algorithm is due to K. Wright. In his study on inferability of unions of pattern languages from positive data [Wri89], Wright presented an efficient method to compute a 2-mmg for the class $(\mathrm{P}_1)^2$ of pairs of one-variable patterns. The essence of his method is to minimize each description $p \in P$ by computing an mcg of the members in P that any description other than p does not cover. We applied the idea to k-mmg for first-order terms for every $k \geq 1$ [ASO93]. Here, we extend the method for more wider classes $\mathcal{C}^k$ of patterns and structured objects for every $k \geq 1$.

The original algorithms mainly consist of two parts. The first part computes a rough approximation P of a sample. Then, the second part refines each $p_1, \ldots, p_m \in P$ ($m \leq k$) by computing an mcg to fit to the sample more.

Algorithm 1: Computing a normal form of P with respect to a sample S, where P is a reduced multiple covering of S.

---

procedure $NF(P, S)$
begin
  while there is an unmarked member $p \in P$ **do**
    Compute a $MSCR(q, S|p)$, where $S|p = S - L(P \backslash p)$;
    Mark q and replace p in P by q;
  end while ;
  Output P;
end ;

The first part is to compute a reduced k-multiple description. It seems to be difficult to generalize this part for wider classes of objects. Because, methods already proposed heavily depends on the structure of objects. Therefore, we deal with this problem in the next section. We only note here that if we can efficiently compute a $CPC(p, S)$ (complete set of partial covers), we can also effectively find a reduced k-multiple description from the sample.

The second part can be easily extended to k-multiple descriptions for various classes of objects. It only need the existence of an efficient single generalization procedure that, given p and S, can find more specific mcg of S than p.

We give an algorithm $NF(P, S)$ for the second part in Algortihm 2. Whenever a reduced k-description P is given, $NF(P, S)$ correctly computes a k-mmg of S by refining P If for every $p \in P$, p is an mcg of $S - L(P \backslash p)$, then we say that P is of normal form with respect to S. We sometimes write $S|p$ for the set $S - L(P \backslash p)$ if P is clear from context. Note that $q \leq p$, for any covering P of S and an mgc q of $S|p$.

Lemma 6 *Let S be a sample. If P is a covering of S that is reduced with respect to S, then P is of normal form with respect to S iff P is a k-mmg of S, where $k = |P|$.*

Lemma 7 *If P is a covering of S that is reduced with respect to S, then the set $NF(P, S)$ is of normal form with respect to S.*

For example, let us consider how the algorithm $NF(P, S)$ works in the case of $k = 2$. See a finite set D in Figure 1, where we write an arrow from q to p if $p \leq q$. Let $S = \{r_1, \ldots, r_6\}$ be a sample and $P = \{p_1, p_2\}$ be a cover of S.

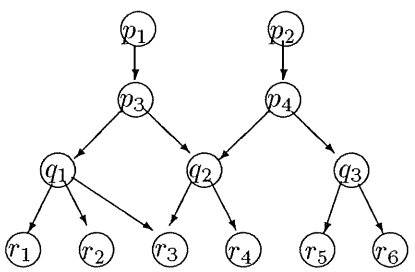

Figure 1: Computing a normal form $NF(P, S)$ on a partial ordered set of descriptions $D = \{p_1, p_2, \ldots, q_1, \ldots, r_1, \ldots, r_6\}$, where $P = \{p_1, p_2\}$ and $S = \{r_1, \ldots, r_6\}$

First, we refine p_1 by taking an mcg q_1 of $S - L(p_2) = \{r_1, r_2\}$. Then, we get $P' = \{q_1, p_2\}$. Next, we refine p_2 by taking an mcg p_4 of $S - L(q_1) = \{r_4, r_5, r_6\}$. Thus, $P'' = \{q_1, p_4\}$, which is a 2-mmg of S. Note that we can not refine both of p_1 and p_2 simultaneously. Then, we get $\{q_1, q_3\}$, but it is not a 2-mmg of S.

Now, we present the algorithm MMG in Algorithm 2. We prove that k-mmg is polynomial time computable for any multiple generalization class $\mathcal{C}^k$ that satisfies conditions we already discussed.

Theorem 8 *Assume that the class $(D, \leq)$ of descriptions satisfies Condition 1 and Condition 3. Then, the algorithm MMG, given a sample S, computes a k-mmg of S in polynomial time in $||S||$.*

Proof. Assume $k \geq 1$ and a sample S are given to the algorithm. First we show by induction on $i \geq 1$ that whenever the algorithm enters the while loop (from Line 4 to Line 8) at i-th time, the current P is an m-mmg of S for $m = |P|$. Note that if the algorithm enters the while loop at the first time, P must be an mcg of S because the greatest element $\{\top\}$ of D^k covers any generalization of S. Thus, the base case holds. Assume that it enters the while loop. And then, assume that P is a l-mmg of S for $l = |P|$, and that it finds $p \in P$ that is divisible. The following claim says that the algorithm can find a Δk-division ΔP of p by picking up all the subset $P \subseteq CPC(p, S)$ with $|P| \leq k$ in polynomial time in $||S||$ because $|CPC(p, S)|$ and the time to test the membership are bounded by some polynomial in $||S||$.

Claim1. Let $k > 1$. Assume that p is an mcg of S. If p is k-divisible with respect to S, then there is a k-division that is a subset P of $CPC(p, S)$.

Next, by Lemma 7, it can find in polynomial time a Δk-mmg of $S|p$ by computing a normal form $(\Delta P, S)$, where $\Delta k = k - |P| + 1$. Let P' be $P \backslash p \cup NF(\Delta P, S)$. Then, the next claim holds.

Claim2. Let $k, m \geq 1$, P be a k-multiple covering of S that is of normal form. Assume that for some $p \in P$, there is an m-division Λ of p that is of normal form. Then, $P' = P \backslash p \cup \Lambda$ is again of normal form.

Therefore, combining Lemma 7, P' is a m'-mmg of S for $m' = |P'|$. On the other hand, we can show the following.

Claim3. Let P be an m-mmg of S. If $k > m$ and $Q \sqsubset P$ for some k-multiple covering Q of S, then some member $p \in P$ is Δk-divisible with respect to $S|p$, where $\Delta k = k - |P| + 1$ and $S|p = S - L(P \backslash p)$.

Combining Claim 1, Claim 2 and Claim 3, we know that at any time it satisfies the condition of the while loop, it can computes the next value($= P'$) from P increasing the cardinality $|P|$ by at least one. At last, if there is no Δk-divisible one in P with respect to S for $\Delta k = k - |P| + 1$, then it terminates. Then, Claim 3 shows that P is a k-mmg of S. Since $|P|$ must be at most k, it enters the while loop at most k times. It is not hard to see that the algorithm runs in polynomial time in $||S||$. □

Corollary 9 Let $k \geq 0$ and $\mathcal{C}$ be an efficient generalization system satisfying Condition 1, Condition 2, and Condition 3. If $\mathcal{C}^k$ be a k-multiple generalization system induced by $\mathcal{C}$, then $\mathcal{C}^k$ is identifiable in the limit from positive data with consistent and conservative polynomial time updating. □

Using this algorithm, we show that the class $(\mathrm{PL}_1)^k$ is polynomial time inferable from positive data. Let Σ be an alphabet.

Lemma 10 (Angluin [Ang80]) *There is an algorithm that, given a sample S, computes an mcg of S in polynomial time in $||S||$. The class P_1 satisfies Condition 1.*

Lemma 11 (Wright [Wri89]) *Let $k \geq 1$. If $|\Sigma| > k$, then the class $(\mathrm{PL}_1)^k$ satisfies Condition 2.*

Theorem 12 Let Σ be an alphabet with $|\Sigma| > k$. Then, the class $(\mathrm{PL}_1)^k$ of unions of at most k one-variable patterns is identifiable in the limit from positive data with consistent and conservative polynomial time updating.

Proof. Let $p \in \mathrm{P}_1$, x be the variable in p and let $p^{-1}(S)$ be the set $\{ u \in \Sigma^+ \mid w = p\{x := u\}, w \in S\}$. Wright showed that if $q \in \mathrm{P}_1$ is an mcg of $p^{-1}(S)$, then $r = p\{x := q\}$ is an mcg of S. Then, r is $MSCR(p, S)$, and it is polynomial time computable by Lemma 10. For $w_1, w_2 \in \Sigma^+$ and integers n_1, n_2, let $pat(w_1, w_2, n_1, n_2)$ be

Algorithm 2: Computing a k-mmg of a sample S

---

procedure $MMG(k, S)$
begin
1 Initialize $P = \{\top\}$; /* the greatest member in D^k */
2 Set $P = NF(P, S)$;
3 Set $\Delta k = k$;
4 **while** there is a Δk-divisible $p \in P$ w.r.t. $S|p$ **do**
/* where $S|p = S - L(P\backslash p)$ */
5 Compute a Δk-division ΔP of p w.r.t. $S|p$;
6 Set $P = (P\backslash p) \cup NF(\Delta P, S)$;
7 Set $\Delta k = k - |P| + 1$;
8 **end while** ;
9 Output P;
end ;

---

the set of one-variable patterns $p \in \mathrm{P}_1$ such that for some $s_1, s_2 \in \Sigma^+$ (1) $p\{x := s_i\} = w_i$ for all $i = 1, 2$, and (2) the s_1 and s_2 start with distinct letters and $|s_1| = n_1$ and $|s_2| = n_2$, respectively. The set is computed in polynomial time (in $O(||S||^5)$). Consider the set

$$\mathcal{H}(S) = \{\, cx \mid c \in \Sigma\} \cup \{\, pat(w_1, w_2, n_1, n_2) \mid w_1, w_2 \in S, |w_1| \leq n_1, |w_2| \leq n_2\}.$$

By an argument similar to one in [Wri89], we can show that the set $\{\, p\{x := q\} \mid q \in \mathcal{H}(S)\}$ is a complete set $CPC(p, S)$ and is polynomial time computable. Therefore, P_1 satisfies Condition 3 and the result follows from Lemma 10, Lemma 11 and Corollary 9. □

More precisely, the algorithm for $(\mathrm{PL}_1)^k$ described in the proof of Theorem 12 runs in time $O(n^{4k+1})$ for $k > 1$, where $n = ||S||$ and the time bound includes the time $O(n^5)$ to find an mcg using Angluin's algorithm.

6 Refinement operators for k-mmg

To construct a k-mmg algorithm for a particular concept class $\mathcal{C}^k$, we require algorithms for solving two search problems $\mathrm{CPC}_{\mathcal{C}}$ and $\mathrm{MSCR}_{\mathcal{C}}$. However, the construction of an algorithm for $\mathrm{CPC}_{\mathcal{C}}$ much depends on the structure of $\mathcal{C}$. In this section, we present a construction of an algorithm for $\mathrm{CPC}_{\mathcal{C}}$ in more uniform way. Then, we show the polynomial time inferability of the class $(\mathrm{PL}_{\mathrm{reg(m)}})^k$ of unions of at most k regular pattern languages with bounded numbers of variables.

First we introduce the notion of refinement operator [Lai88, Sha81]. For a binary relation R, $R(a)$ denote the set $\{\, b \mid (a, b) \in R\}$ and R^+ denote the transitive closure of R. Let $\mathcal{C} = (C, D, L(\cdot))$ is an efficient generalization system. A *refinement operator* for $\mathcal{C}$ is a subrelation ρ of the strict relation $(D, <)$. A refinement operator ρ is *subsumption complete* if $p < q \iff p \in \rho^+(q)$ for any $p, q \in D$. A refinement operator is *efficient* if the set $\rho(p)$ is polynomial time computable.

Theorem 13 *Let $\mathcal{C}$ be an efficient generalization system, and ρ be a refinement operator for $\mathcal{C}$. If ρ is subsumption complete and efficient, then $\mathcal{C}$ satisfies Condition 3.*

Proof. Let S be a sample and $p \in D$ be a cover of S. Then, the following procedure computes $MSCR(p, S)$.

```
begin
   while there is some q ∈ ρ(p) such that S ⊆ L(q) do
      p := q
   end while ;
   Output p;
end ;
```

If $q \in \rho(p)$ then $size(q) > size(p)$. If $S \subseteq L(p)$, then $size(p)$ does not exceed $size(w)$. Therefore, the while loop must be executed at most n times, where $n = max\{\, size(w) \mid w \in S\}$. Recall that $size(p)$ and $|p|$ are polynomially related by Definition 3. Thus, the procedure runs in polynomial time in $|p|$ and $||S||$. On the other hand, the subsumption completeness shows that $\rho(p)$ is actually $CPC(p, S)$. Because, any $q \in \rho(r)$ is properly smaller refinement of u. Thus, q can not cover the whole S since p is an mcg of S. Therefore, $\rho(p)$ is a $CPC(p, S)$ and the result follows from that ρ is efficient. □

Now, we apply the result to polynomial time inference of unions of regular patterns. Let $m \geq 0$ and $\mathrm{P}_{\mathrm{reg(m)}}$ be the class of *regular patterns with at most m variables* and $\mathrm{PL}_{\mathrm{reg(m)}}$ is the class of pattern languages defined by patterns in $\mathrm{P}_{\mathrm{reg(m)}}$.

Definition 7 We define the *size* of a pattern p by $size(p) = 2 \times |p| - |v(p)|$.

Definition 8 A substitution θ is *basic* for $p \in \mathrm{P}_{\mathrm{reg(m)}}$ if $v(q\theta) \leq m$ and θ satisfies one of the followings:

(1) $\theta = \{x := a\}$ where $x \in v(p)$ and $a \in \Sigma$.
(2) $\theta = \{x := ay\}$ where $x \in v(p)$, $y \notin v(p)$ and $a \in \Sigma$.
(3) $\theta = \{x := ya\}$ where $x \in v(p)$, $y \notin v(p)$ and $a \in \Sigma$.
(4) $\theta = \{x := yz\}$ where $x \in v(p)$, $y \notin v(p)$ and $z \notin v(p)$.

Definition 9 Let $m \geq 0$ and $p, q \in \mathrm{P}_{\mathrm{reg(m)}}$. Then, q is in $\rho_m(p)$ iff $q = p\theta$ for a basic substitution θ for $\mathrm{P}_{\mathrm{reg(m)}}$.

Lemma 14 *ρ_m is a subsumption complete and efficient refinement operator for* $\mathrm{P}_{\mathrm{reg(m)}}$.

To prove the inferability of $\mathrm{PL}_{\mathrm{reg(m)}}$, we need some lemmas. By a similar discussion in Mukouchi [Muk92], we can show the following lemma.

Lemma 15 *Let $k \geq 1$ and $m \geq 0$. If $|\Sigma| > 2km$, then the class $(\mathrm{PL}_{\mathrm{reg(m)}})^k$ satisfies Condition 1 and Condition 2.*

Proof. Assume that for $p, q_1, \ldots, q_m$ $(1 \leq m \leq k)$, $L(p) \subseteq L(q_1) \cup \cdots \cup L(q_m)$. Let $p = \pi_0 x \pi_1 x \cdots \pi_{m-1} x \pi_m$. Then, there is some $c \in \Sigma$ that is distinct from any letter in p that occurs previous or next to a variable $x \in v(p)$, that is, positions at both ends of π_i for every $1 \leq i \leq m$, the right end of π_0, and the left end of π_m. Therefore, there is a substitution θ that maps each variable to sufficiently long sequence of c's and that $p\theta \leq q_i$ for some i means that $p \leq q_i$. Also the case of $k = 1$ shows Condition 1. Hence, the result follows. □

Combining Corollary 9, Theorem 13 and lemmas proved above, we can prove the following theorem.

Theorem 16 *Let $k \geq 1$, $m \geq 0$, and Σ be an alphabet with $|\Sigma| > 2km$. Then, the class $(\mathrm{PL}_{\mathrm{reg(m)}})^k$ of unions of at most k regular patterns with at most m variables is identifiable in the limit from positive data with consistent and conservative polynomial time updating.*

From Lemma 4, the class $\mathrm{PL}_{\mathrm{reg(m)}}$ has finite thickness. By the bound of maximum number of variables in a pattern, we can strengthen this. For any $w \in \Sigma^+$, the set $H(w) = \{\, L \in \mathrm{PL}_{\mathrm{reg(m)}} \mid w \in L\}$ is of polynomial cardinality $O(l^{2m})$, where $l = |w|$. Then, we can construct a trivial algorithm that computes k-mmg of a

sample S by enumerating elements in the set $\bigcup\{ H(w_1) \times \cdots \times H(w_k) \mid w_1, \ldots, w_k \in S\}$. Thus, it also runs in polynomial time $O(n^{2km+k+1})$ in $||S|| = n$.

We compare the complexity of the trivial algorithm with that of ours. Our algorithm runs more efficiently than a trivial one. Using a linear time matching algorithm in [Shi82], we can compute $MSCR(p, S)$ in time $O(|p| \cdot (|p| + ||S||))$. Since the number of variables is bounded by m, $|\rho_m(p)| = O(m)$. Thus, our algorithm runs in time $O((4m)^k \cdot n^2)$ in $||S|| = n$. Hence, we can conclude that our method to compute k-mmg significantly improves the efficiency.

The algorithm is also efficient in the sense of parallel computation. By the bound m of variables, we can specify a generalization p of w by at most $2m$ positions on w. This is done by a fixed number of binary counters that point a letter on an input string. We also compute pattern matching and enumeration of $\rho_m(p)$ with some binary counters. Therefore, we can construct a deterministic turing machine that compute k-mmg of S using space $O(\log n)$. Since DLOG $\subseteq$ NC^2, we can say that the class $(\mathrm{PL}_{\mathrm{reg(m)}})^k$ is *NC-identifiable in the limit* from positive data consistently and conservatively if $|\Sigma| > 2km$. Note that this construction is also possible for the trivial algorithm.

References

[Ang80] D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, Vol. 21, pp. 46–62, 1980.

[ASO93] H. Arimura, T. Shinohara, and S. Otsuki. A polynomial time algorithm for finding finite unions of tree pattern languages. In *Proceedings of the Second International Workshop on Nonmonotonic and Inductive Logic*, pp. 118–131, 1993. Lecture Notes in Artificial Intelligence 659.

[BJO91] P. Bunemann, A. Jung, and A. Ohori. Using powerdomains to generalize relational databases. *Theoretical Computer Science*, Vol. 91, pp. 23–55, 1991.

[BK89] F. Bancilhon and S. Khoshafian. A calculas for complex objects. *Journal of computer and system science*, Vol. 38, pp. 326–340, 1989.

[Lai88] P. D. Laird. *Learning from good and bad data.* Kluwer Academic, 1988.

[LMM88] J-L. Lassez, M.J. Maher, and K. Marriott. Unification revisited. In J. Minker, editor, *Foundations of Deductive Databases and Logic Programming*, pp. 587–625. Morgan Kaufmann, 1988.

[Muk92] Y. Mukouchi. Containment problems for pattern languages. *IEICE Trans. Inf. and Syst.*, Vol. E75-D, No. 7,, 1992.

[Plo70] G. Plotkin. A note on inductive generalization. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence*, volume 5, pp. 153–163. Edinburgh University Press, 1970.

[Rey70] J. Reynolds. Transformational systems and the algebraic structure of atomic formulas. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence*, volume 5, pp. 135–152. Edinburgh University Press, 1970.

[Sha81] E. Y. Shapiro. Inductive inference of theories from facts. Technical Report 192, Yale University, Department of Computer Science, 1981.

[Shi82] T. Shinohara. Polynomial time inference of pattern languages and its applications. In *Proceedings of the 7th IBM Symposium on Mathematical Foundations of Computer Science*, pp. 191–209, 1982.

[Wri89] K. Wright. *Inductive Inference of Pattern Languages.* PhD thesis, University of Pittsburgh, 1989.

About the Author

Hiroki Arimura (有村博紀) was born in Fukuoka on June 7, 1965. He received the B.S. degree in 1988 in Phisics, and the M.S. degree in 1990 in Information Systems from Kyusyu University. Presently, he is an Assistant of Kyushu Institute of Technology. His research interests are in logic programming and computational learning theory.

Takeshi Shinohara (篠原　武) was born in Fukuoka on January 23, 1955. He received the B.S. in 1980 from Kyoto University, and the M.S. degree and the Dr. Sci. from Kyushu University in 1982, 1986, respectively. Currently, he is an Associate Professor of Department of Artificial Intelligence, Kyushu Institute of Technology. His present interests include information retrieval, string pattern matching algorithms and computational learning theory.

Setsuko Otsuki (大槻説乎) was born in Tokushima on July 8, 1932. She graduated from Department of Physics, Kyoto University in 1955, and received the Dr. Eng. degree in 1971 in Department of Engineering from Kyushu University. Presently, she is a Professor of Department of Artificial Intelligence, Kyushu Institute of Technology. Her research interests include intelligent tutoring systems, knowledge information processing and natural language understanding.

Research Institute of Fundamental Information Scinece, Kyushu University 33, Fukuoka 812, Japan.