INFERRING UNIONS OF TWO PATTERN LANGUAGES

Shinohara, Takeshi
Computer Center, Kyushu University

https://doi.org/10.5109/13347

Bulletin of Informatics and Cybernetics Vol. 20, No. 3~4, 1983

INFERRING UNIONS OF TWO PATTERN LANGUAGES

By

Takeshi SHINOHARA*

Abstract

A pattern is a string of constant symbols and variable symbols. The language $L(p)$ of a pattern p is the set of all strings obtained by substituting any non-empty constant string for each variable symbol in p. In this paper we consider inference from positive data for unions of two pattern languages, that is, identification of $L(p) \cup L(q)$ when an enumeration of elements in the union is given.

1. Introduction

In general, two kinds of inferences have been known, inference from positive data and inference from positive and negative data. A general theory of inductive inference is found in literatures [3, 6]. Briefly, inference of languages from positive data is to identify a language when an enumeration of elements in the language is given.

Inference of unions from positive data, we are concerning with in this paper, is an inference of languages when given enumeration is not of one language but of two languages, and is found in the following problems.

1) Bilingual learning of children whose parents are internationally married.

2) Error detection when correct data and wrong data are shuffled.

Historically, the inference from positive data has been considered of little interest to study, since Gold [3] proved that any class of languages is not inferrable from positive data if it contains all finite languages and at least one infinite languages, and hence that even the class of regular sets is not inferrable from positive data. However, recently Angluin [1, 2] proved a theorem characterizing the class of languages inferrable from positive data and presented interesting classes including the class of pattern languages. Shinohara has found two subclasses of pattern languages applicable to practical problems [4], and has got a similar result for extended pattern languages [5]. Here a pattern is a string of constant symbols and variable symbols, and the language of a pattern p is the set of all strings obtained by substituting any non-empty constant string for each variable symbol in p.

In this paper we first show that the class of pattern languages is not closed under union. Without this property, considering inference for unions of pattern languages would make no sense. Then we show that the class of unions of two pattern languages is inferrable from positive data.

* Computer Center, Kyushu University 91, Fukuoka, 812 Japan.

2. Patterns and Their Languages

We begin with the definitions on patterns and their languages.

Let Σ be a finite set of symbols containing at least two symbols and let $X=\{x_1, x_2, \cdots\}$ be a countable set of symbols disjoint from Σ. Elements in Σ are called *constants* and elements in X are called *variables*. A *pattern* is any string over $\Sigma \cup X$. The set $(\Sigma \cup X)^*$ of all patterns is denoted by P.

Let f be a non-erasing homomorphism from P to P. If $f(a)=a$ for any constant a, then f is called a *substitution*. If f is a substitution, $f(x)$ is in X, and $f(x)=f(y)$ implies $x=y$ for any variables x and y, then f is called a *renaming of variables*. We define two binary relations on P as follows:

1) $p \equiv' q$ iff $p=f(q)$ for some renaming of variables f,

2) $p \leqq' q$ iff $p=f(q)$ for some substitution f.

The *language of a pattern* p is the set

$$L(p)=\{w \in \Sigma^* \mid w \leqq' p\},$$

and the *class of pattern languages* is

$$PL=\{L(p) \mid p \in P\}.$$

These syntactic relations $\leqq'$ and $\equiv'$ are characterized by the following lemma.

LEMMA 1. [2]

1) *For all patterns p and q, $p \equiv' q$ iff $L(p)=L(q)$.*

2) *For all patterns p and q, if $p \leqq' q$ then $L(p) \subseteq L(q)$, but the converse is not true in general.*

3) *If p and q are patterns such that $|p|=|q|$, then $p \leqq' q$ iff $L(p) \subseteq L(q)$, where $|p|$ is the length of p.*

We say that a pattern p is in *canonical form* if $x_1, x_2, \cdots, x_k$ are all variables in p and the leftmost occurrence of x_i is left to the leftmost occurrence of x_{i+1} for each $i=1, \cdots, k-1$. Clearly, for any pattern p, there uniquely exists a pattern $\hat{p}$ in canonical form such that $\hat{p} \equiv' p$.

3. Inference from Positive Data

In this section we briefly describe inference from positive data for later discussion.

Inference machine is an effective procedure which requires inputs from time to time and produces outputs from time to time. Let $s=s_1, s_2, \cdots$ be an arbitrary infinite sequence, and let $g=g_1, g_2, \cdots$ be a sequence of outputs produced by an inference machine M when inputs in s are successively given to M on request. Then we say that M *on input s converges to g_0* iff g is a finite sequence ending with g_0 or all but finitely many elements of g are equal to g_0.

Let $L=L_1, L_2, \cdots$ be a class of recursive languages, and let $s=s_1, s_2, \cdots$ be an arbitrary enumeration of some language L_i. Then we say that a machine M *infers L from positive data* if M on input s converges to an index j with $L_j=L_i$. We say that a class L is *inferrable from positive data* if there exists a machine which infers L from

positive date.

To prove inferrability from positive data for unions of two pattern languages, we use the following theorem.

THEOREM 1. [2] *A class $L=L_1, L_2, \cdots$ of languages is inferrable from positive data iff L satisfies the following condition.*

CONDITION. *There exists an effective procedure which enumerates elements in a set T_i for any index i, where*

1) *T_i is finite,*
2) *$T_i \subseteq L_i$, and*
3) *for no index j, $T_i \subseteq L_j \subsetneq L_i$.*

T_i in Theorem 1 is called a *tell-tale finite subset* of L_i. Note that L_i is a minimal language containing T_i.

4. Inference for Unions of Two Pattern Languages

Let $L=L_1, L_2, \cdots$ be a class of recursive languages and $I=\{1, 2, \cdots\}$ be the set of indexes. Then we define $I^2=\{(i, j) | i, j \in I\}$, $L_{(i,j)}=L_i \cup L_j$, and L^2 is the class of languages whose index set is I^2. Then inference for unions of two pattern languages is nothing less than the inference for PL^2. In our case, the set of indexes is the set P^2 of all pairs of two patterns and $L(i) \cup L(j)$ is denoted by $L(i, j)$.

First we show that the class of pattern languages is not closed under union. The following lemma is used in the proof.

LEMMA 2. [2] *If $|p|=|q|$, then*

$$L(p) \cap \Sigma^{|p|} \subseteq L(q) \Longrightarrow L(p) \subseteq L(q) \quad \textit{and} \quad p \leqq' q .$$

THEOREM 2. *If $card(\Sigma) \geqq 3$, then*

$$L(p) \cup L(q) = L(r) \Longrightarrow p \equiv' r \quad \textit{or} \quad q \equiv' r$$

for any patterns p, q, and r, where $card(S)$ is the number of elements in S.

PROOF. Assume $L(p) \cup L(q) = L(r)$, $|p| \leqq |q|$, $p \not\equiv' r$, and $q \not\equiv' r$. The minimum lengths of strings in $L(p) \cup L(q)$ and $L(r)$ are equal to $|p|$ and $|r|$, respectively, therefore

$$|p| = |r| .$$

$L(p) \subseteq L(r)$ from our assumption and hence by Lemma 1

$$p \leqq' r .$$

By Lemma 2, there exists a string $w \in (L(r) \cap \Sigma^{|r|}) - L(p)$. Since the string w must be in $L(q)$, $|r| = |w| \geqq |q|$. Therefore

$$|p| = |q| .$$

Again by Lemma 1

$$q \leqq' r .$$

Here we should note that if $|s| = |t|$, $s \leqq' t$, and $s \not\equiv' t$, then

$$s\langle i \rangle \in \Sigma \quad \text{and} \quad t\langle i \rangle \in X \qquad \text{for some } i, \text{ or}$$

$$s\langle i\rangle,\ s\langle j\rangle,\ t\langle i\rangle,\ t\langle j\rangle \in X,\quad s\langle i\rangle = s\langle j\rangle,\quad \text{and}\quad t\langle i\rangle \neq t\langle j\rangle \qquad \text{some } i \text{ and } j,$$

where $v\langle i\rangle$ denotes the i-th symbol in v from the left. A typical example is $s=012$, $t=0x2$ or $s=x1x$, $t=x1y$, where $\Sigma=\{0, 1, 2\}$ and $X=\{x, y, \cdots\}$.

Clearly, from our observation, there exists a string $w \in L(r) \cap \Sigma^{|r|}$ satisfying the following conditions:

1) If $p\langle i\rangle \in \Sigma$ and $r\langle i\rangle \in X$ for some i, then

$$w\langle i\rangle \neq p\langle i\rangle, \qquad \text{or}$$

if $p\langle i\rangle,\ p\langle j\rangle,\ r\langle i\rangle,\ r\langle j\rangle \in X$, $p\langle i\rangle = p\langle j\rangle$, and $r\langle i\rangle \neq r\langle j\rangle$ for some i and j, then

$$w\langle i\rangle \neq w\langle j\rangle, \qquad \text{and}$$

2) If $q\langle i'\rangle \in \Sigma$ and $r\langle i'\rangle \in X$ for some i', then

$$w\langle i'\rangle \neq q\langle i'\rangle, \qquad \text{or}$$

if $q\langle i'\rangle,\ q\langle j'\rangle,\ r\langle i'\rangle,\ r\langle j'\rangle \in X$, $q\langle i'\rangle = q\langle j'\rangle$, and $r\langle i'\rangle \neq r\langle j'\rangle$ for some i' and j', then

$$w\langle i'\rangle \neq w\langle j'\rangle.$$

Then $w \notin L(p)$ by condition 1) and $w \notin L(q)$ by condition 2). Hence, by contradiction, our proof is completed. □

The condition $card(\Sigma) \geqq 3$ is necessary in Theorem 2 because if $p\langle i\rangle,\ q\langle i\rangle \in \Sigma$, $p\langle i\rangle \neq q\langle i\rangle$, and $r\langle i\rangle \in X$, then we need at least three constant symbols to satisfy two conditions

$$w\langle i\rangle \neq p\langle i\rangle \quad \text{and} \quad w\langle i\rangle \neq q\langle i\rangle.$$

In fact, $L(0x) \cup L(1x) = L(xy)$ when $\Sigma=\{0, 1\}$.

Now we state our main theorem and prove it.

THEOREM 3. *PL^2 is inferrable from positive data.*

PROOF. We show that PL^2 satisfies Condition of Theorem 1. We present an effective procedure which enumerates elements in a tell-tale finite subset of $L(p, q)$ for any pair (p, q) of patterns given. Consider the following procedure, where (p, q) is given pair of patterns, $|p| \leqq |q|$, and assignments to T is the enumeration.

```
init: T := L(p) ∩ Σ^|p|;                                          (1)
      F := {(r, s); |p| = |r| = |s|, r and s are canonical, T ⊆ L(r, s)};   (2)
      start enumeration of Σ+;                                    (3)
      w := the first element of Σ+;                               (4)
      while w ∈ L(p) or w ∉ L(q) do                               (5)
          begin
              if w ∈ L(p) then                                    (6)
                  begin
                      F1 := {(r, s) ∈ F; w ∉ L(r, s)};            (7)
                      if F1 ≠ φ then T := T ∪ {w};                (8)
                      F := F − F1                                 (9)
                  end;
              w := the next element of Σ+                         (10)
```

end;

reset: $T:=T\cup(L(q)\cap\Sigma^{|q|})\cup\{w\}$; (11)

$F:=\{(r, s);\ |p|=|r|\leqq|s|\leqq|w|,\ r \text{ and } s \text{ are canonical}, T\subseteq L(r, s)\}$; (12)

restart enumeration of Σ^+; (13)

for each $w\in\Sigma^+$ **do** (14)

if $w\in L(p, q)$ **then** (15)

begin

$F2:=\{(r, s)\in F;\ w\notin L(r, s)\}$; (16)

if $F2\neq\phi$ **then** $T:=T\cup\{w\}$; (17)

$F:=F-F2$ (19)

end

The value of T at (1), the value of F at (2), the value of T at (11), and the value of F at (12) are all finite. It is easily shown that this procedure enumerates elements in a subset of $L(p, q)$. We must show that T^*, the finite subset of $L(p, q)$ enumerated by this procedure, is a tell-tale.

Assume T^* is not a tell-tale finite subset of $L(p, q)$. Then, by the definition, there exists a pair (r, s) of patterns satisfying

(*) $T^*\subseteq L(r, s)\subsetneqq L(p, q)$, $|r|\leqq|s|$, r and s are canonical.

Since we assume $|p|\leqq|q|$, $L(p)$ is not properly contained in $L(q)$ and hence we consider two cases according to whether $L(q)-L(p)=\phi$ or not.

Case 1. Assume $L(q)-L(p)=\phi$. Then the condition $w\in L(p)$ or $w\notin L(q)$ of the while statement (5) is always satisfied. The while statement never terminates and statements under "reset", (11)-(19), are never executed. Note that $L(p, q)=L(p)$ holds in this case.

Let (r, s) be a pair of patterns satisfying (*). Then

$$min\{|w|;\ w\in L(p, q)\}=|p|\leqq min\{|w|;\ w\in L(r, s)\}=|r|.$$

Since $L(p)\cap\Sigma^{|p|}\subseteq T^*\subseteq L(r, s)$, $|p|\geqq|r|$. Therefore

$$|p|=|r|.$$

By Lemma 2, there exists a string $w\in(L(p)\cap\Sigma^{|p|})-L(r)$. The string w must be in $L(s)$ and $|w|=|p|=|s|$. Hence

$$|p|=|r|=|s|.$$

The pair (r, s) appears in F at (2) and never appears in $F1$ at (9). However any string $w'\in L(p, q)-L(r, s)$ will appear in the enumeration of Σ^+ in time and then the pair (r, s) will appear in $F1$ at (9). This is a contradiction.

Case 2. Assume $L(q)-L(p)\neq\phi$. Then the while statement terminates and statements under "reset", (11)-(19), are executed. If we can show that any pair, (r, s), satisfying, (*), appears in F at (12), then a contradiction is easily derived in a similar way to Case 1. Let (r, s) be a pair of patterns satis-fying (*). Note that

$$T\supseteq(L(p)\cap\Sigma^{|p|})\cup(L(q)\cap\Sigma^{|p|})\cup\{w\}$$

holds at (11). Obviously $|p|=|r|$.

There are two cases to consider. Let w be the w at (11).

1) Assume $L(p)\cap\Sigma^{|p|}\not\subseteq L(r)$. Then clearly $|s|=|p|\leqq|w|$.

2) Assume $L(p)\cap\Sigma^{|p|}\subseteq L(r)$. Then $L(p)\subseteq L(r)$ by Lemma 2. If $L(p)=L(r)$, then $w\notin L(r)$ and hence $w\in L(s)$ and $|s|\leqq|w|$. Otherwise, if, $L(p)\subsetneqq L(r)$, then there exists a string v such that $v\in(L(r)\cap\Sigma^{|r|})-L(p)$. Since the string v must be in $L(q)$, $|p|=|q|$. Clearly, by Lemma 2, $L(q)\cap\Sigma^{|p|}\not\subseteq L(r)$ and hence there exists a string v' satisfying $v'\in L(q)\cap\Sigma^{|q|}$ and $v'\in L(s)$. Therefore

$$|s|=|q|\leqq|w|.$$

In either 1) or 2), $|p|=|r|\leqq|s|\leqq|w|$ is satisfied and hence the pair (r, s) appears in F at (12). □

5. Concluding Remarks

we have discussed inference from prositive data for unions of two pattern languages. Unions of other classes than pattern languages should be considered on their inferrabilities. For example, whether the class of unions of two extended pattern languages is inferrable from positive data or not should be an interesting problem. The extended language of a pattern p is the set of all strings obtained by substituting any (possibly empty) constant string, instead of non-empty constant string, for each variable in p.

To find practical problems, which can be modeled by inference for unions, is another interesting future subject. The computational complexity problem, we have not referred in this paper, should be considered in practical applications of inference.

Acknowledgements

The author wishes to thank Prof. S, Arikawa for his constant supoort and encouragement. He also acknowledges helpful suggestions by S. Miyano in the beging of this study.

References

[1] ANGLUIN, D.: *Finding Patterns Common to a Set of Strings*, in "Proceedings, 11th Annual ACM Symposium on Theory of Computing," (1979), 130-141.

[2] ANGLUIN, D.: *Inductive Inference of Formal Languages from Positive Data*, Inform. Contr. **45**, (1980), 117-135.

[3] GOLD, E.M.: *Language Identification in the Limit*, Inform. Contr. **10**, (1967), 447-474.

[4] SHINOHARA, T.: *Polynomial Time Inference of Pattern Languages and Its Application*, in "Proceedings, 7th IBM Symposium on Mathematical Foundation of Computer Science," (1982).

[5] SHINOHARA, T.: *Polynomial Time Inference of Extended Regular Pattern Languages*, in "Proceedings, SSE Symposium, Kyoto," (1982).

[6] SOLOMONOFF, R.: *A Formal Theory of Inductive Inference*, Inform. Contr. **7**, (1964), 1-22.

Communicated by S. Arikawa
Received October 12, 1982