INFORMATIQUE THÉORIQUE ET APPLICATIONS

GILBERTO FILÉ

The relation of two patterns with comparable languages patterns

Informatique théorique et applications, tome 23, n° 1 (1989), p. 45-57.

<http://www.numdam.org/item?id=ITA_1989__23_1_45_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 23, n° 1, 1989, p. 45 à 57)

THE RELATION OF TWO PATTERNS WITH COMPARABLE LANGUAGES PATTERNS

by Gilberto FILÉ [1]

Abstract. – *A pattern is a string consisting of terminals and variables. The language defined by a pattern is the set of terminal strings obtained by substituting (consistently) terminal strings to its variables. A pattern simulates another pattern when its language contains that of the other one.*

If q simulates p, one may think that there must be a substitution that applied to q produces p itself. This hypothesis is considered under different assumptions. It results that it is true only for very restricted patterns (with variables only) and only when erasing substitutions are considered. The relation between two patterns is studied also in the case that the languages they produce are equal.

Résumé. – *Un motif est un mot formé de terminaux et de variables. Le langage défini par un motif est l'ensemble des mots terminaux obtenus en substituant (de façon consistante) les mots terminaux aux variables du motif. Un motif simule un autre motif si le langage du premier motif contient le langage du second motif.*

Si q simule p, on peut penser qu'il doit exister une substitution qui, appliquée à q, produit p. Ce problème est considéré sous différentes hypothèses. Il résulte de cette discussion que le problème a une réponse positive seulement pour des motifs très particuliers (contenant seulement des variables) et seulement lorsqu'on considère des substitutions effaçantes. On étudie aussi la relation entre deux motifs lorsque les langages produits sont égaux.

INTRODUCTION

A pattern is a word consisting of terminal symbols and of variables. The language defined by a pattern is the set of strings obtained substituting consistently terminal strings to all its variables.

Patterns were introduced in [1], see also [3], in the context of inductive inference. We consider patterns indipendently of this application. In the study of patterns it is natural to consider the following problem PD: for any two patterns p and q decide whether the language of one contains that of the

[1] C.N.R.S.-L.A. n° 226, Université de Bordeaux-I, France. Present address: Facoltà di Matematica, Università di Padova, via Belzoni 7, 35100, Padova, Italy.

Informatique théorique et Applications/Theoretical Informatics and Applications
0296-1598 89/01 45 13/$3.30/

other. Angulin [1] has left the decidability of PD as an open question. How would one attack such a question? Intuitively, the following Hypothesis H seems reasonable and, if verified, would immediately give a decision method for PD:

If the language of q contains that of p then there must be a substitution f such that $f(q)=p$.

Unfortunately, in [1] it is shown that H is false in the case that one considers only nonerasing substitutions in the definition of language of a pattern. We have considered whether *H* holds at least in some restricted case.

Namely, the following cases are considered:

(i) also erasing substitutions are allowed;

(ii) only pure patterns are considered i. e., patterns that contain only variables.

	erasing	nonerasing
pure patterns	1	2
any pattern	3	4

Figure 1.

This gives us the four cases shown in figure 1. Correspondingly, one has the four problems PD1-PD4 and the four hypothesis H1-H4. Only H4 is known to be false, we studied the remaining three cases.

The first result that we have obtained is that H1 is true. After this, we wanted to verify whether the conditions of pure patterns and erasing substitutions were both necessary. This is the case. Relatively simple counterexamples suffice show that H2 and H3 are false.

Therefore we have a decision method for the inclusion of pattern languages in only one of the four cases. Clearly, this does not imply that the other problems are undecidable. However, they are difficult problems: PD4 is shown to be NP-hard in [1] and it is easy to modify this proof for showing that the same is true for PD3.

The paper is organized as follows. First the necessary definitions are given. In section 2 we show that H1 is true and in section 3 that H2 and H3 are false. In section 4 the relation of two patterns defining equal languages is

studied. The paper is closed by a short conclusion in which some open problems are pointed out.

1. PRELIMINARIES

For any set S, $|S|$ is the number of elements of S and for any string s, $|s|$ is its length. A is a finite set of terminal symbols; $A=\{a, b, c, \ldots\}$. $V=\{x_1, x_2, x_3, \ldots\}$ is a set of variables. A *pattern* p is a word in $(A\cup V)^*$. $\mathrm{Var}(p)=\{x \mid x$ is a variable appearing in $p\}$; $\mathrm{Term}(p)=\{a \mid a$ is a terminal appearing in $p\}$. A pattern is *pure* if it contains only variables. A substitution σ is a function: $V\to(A\cup V)^*$. A substitution σ is *nonerasing* if, for every x in V, $\sigma(x)\neq\lambda$. A substitution is said to be a *variable renaming* if it defines a bijection from V to V. For a pattern p and a substitution σ, $\sigma(p)$ is the string obtained from p by substituting each variable x in it with the string $\sigma(x)$. The *language generated by a pattern* p is the set $L(p)=\{w \mid w$ in A^* and $w=\sigma(p)$ for some substitution $\sigma\}$. The set of all terminal strings that can be generated from p, by means of nonerasing substitutions only, is denoted by $\mathrm{LN}(p)$.

For a pattern p, the i-th position of p, $1\leqq i\leqq|p|$, is denoted by $\langle p, i\rangle$. If the symbol occurring in $\langle p, i\rangle$ is x then $\langle p, i\rangle$ *is an occurrence of* x *in* p. When the pattern under consideration is clear from the context, a position $\langle p, i\rangle$ is denoted with i only. For $x\in\mathrm{Var}(p)$, the sequence of occurrences of x in p is denoted by $\mathrm{Occ}(p, x)$ and is the sequence $\langle i_1, \ldots, i_h\rangle$ such that $1\leqq i_1<i_2<\ldots<i_h\leqq|p|$ and such that $i_1, \ldots, i_h$ are all and only the occurrences of x in p.

As already explained in the introduction, see also figure 1, we want to show the truth or the falsity of the following four hypothesis H1 to H4. Given any two pure patterns p and q,

H 1: $L(q)\supseteq L(p)\Rightarrow$there is a substitution f such that $f(q)=p$.

H 2: $L(q)\supseteq L(p)\Rightarrow$there is a nonerasing substitution f such that $f(q)=p$.

The hypothesis H 3 and H 4 are obtained from H 1 and H 2, respectively, by dropping the hypothesis that p and q are pure.

The falsity of H 4 has been shown in [1] by means of the following counterexample. Let $A=\{0, 1\}$, $p=0x10xx1$ and $q=xxy$.

Similar counterexamples can be found for any finite A, see [1].

It is important to remark the role of the size of the terminal alphabet A for the probleme under consideration. On the one hand, if $|A|=1$, then it is easy to show that H 1 to H 4 are all false. For instance, the following

counterexample suffices for showing that H 1 and H 2 are not verified:

$$p = xyyx \qquad \text{and} \qquad q = xx.$$

On the other hand, if $|A| \geqq |\text{Var}(p)| + |\text{Term}(p)|$, then H 1 to H 4 are trivially verified: substitute each variable of p with a distinct symbol of A that is not in $\text{Term}(p)$, let w be the word obtained, since $L(q) \supseteq L(p)$, there is a substitution σ such that $\sigma(q) = w$; this σ trivially gives a substitution σ' such that $\sigma'(q) = p$. Thus, when considering two patterns p and q we will always assume that $2 \leqq |A| < |\text{Var}(p)| + |\text{Term}(p)|$.

2. H1 IS TRUE

The goal of this section is to show the following theorem.

THEOREM 1: *For an alphabet A containing at least two symbols, H 1 is verified.*

The proof of the theorem is quite long and it is split in several lemmas. Throughout the rest of the section the following notation is used.

NOTATION : p and q are patterns such that $L(q) \supseteq L(p)$; $k = |\text{Var}(p)|$, $k' = |\text{Var}(q)|$, $n = |p|$. □

The idea of the proof of Theorem 1 is that of defining a substitution π that associates to each variable of p a word that has "nothing" in common with the words of the other variables. Through π we obtain an effect similar to that of having an alphabet A such that $|A| \geqq |\text{Var}(p)|$, see the observations at the end of section 1.

SUBSTITUTION π : The notation introduced above is used. Fix an arbitrary total order among the variables of p, i. e., fix a bijection ord: $\text{Var}(p) \to [1, k]$. For each x in $\text{Var}(p)$, π is as follows: let $A = \{a, b\}$ and $\text{ord}(x) = i$,

$$\pi(x) = as_1 as_2 \ldots as_L a,$$

where

$$L = 6\,mk \qquad \text{and} \qquad s_j = b^{(i-1)L+j} \qquad j \in [1, L].$$

A subword ab^ia of $\pi(x)$ is called a *module* of $\pi(x)$. In what follows π' is a substitution such that $\pi'(q) = \pi(p)$. Such a substitution exists because $L(q) \supseteq L(p)$. □

The reason for making π depend on L (and thus on q) is technical and it will become clear in Lemma 1 and 3 below. The following is an important propriety of π.

PROPERTY (*) : For any x in $\mathrm{Var}(p)$ consider a decomposition $\pi(x)=\alpha w \beta$, where w contains at least a module of $\pi(x)$. There is no other decomposition $\alpha' w \beta'$ of $\pi(x)$ where $\alpha \neq \alpha'$. □

If i is in $\mathrm{Occ}(p, x)$ then with $\pi(\langle p, i \rangle)$ we will denote $\pi(x)$. Similarly for π'. This notation is extended to sequences of positions as follows: $\pi(p, i, i+1, \ldots, i+h)$ denotes $\pi(\langle p, i \rangle) \ldots \pi(\langle p, i+h \rangle)$.

Let us define the following two relations:

(1) A position $\langle p, i \rangle$ is *simulated* by the positions $\langle q, j \rangle, \ldots, \langle q, j+h \rangle$ if,

(*a*) $\pi'(q, 1, \ldots, j-1)$ is a prefix of $\pi(p, 1, \ldots, i-1)$ and

(*b*) $\pi'(q, 1, \ldots, j+h)$ contains $\pi(p, 1, \ldots, i)$ as a prefix.

With *Issim* (i) we denote the sequence $\langle j, \ldots, j+h \rangle$.

(2) A position $\langle q, j \rangle$ *simulates* the positions $\langle p, i \rangle, \ldots, \langle p, i+h \rangle$ when $\mathrm{Issim}(i-1)$ (if it exists) does not contain j, $\mathrm{Issim}(i), \ldots, \mathrm{Issim}(i+h)$ all contain j, and $\mathrm{Issim}(i+h+1)$ does not. The sequence $\langle i, \ldots, i+h \rangle$ is denoted by $\mathrm{Sim}(j)$.

It is useful to be able to be more precise about who simulates what: we want to specify also what part of a string is simulated.

Consider two positions $\langle p, i \rangle$ and $\langle q, j \rangle$, such that the first is simulated by the second one. It is easy to understand that in this case $\pi(\langle p, i \rangle)$ and $\pi'(\langle q, j \rangle)$ must have a common substring w. Figure 2 shows one possible situation of the simulation of $\langle p, i \rangle$ by $\langle q, j \rangle$. Obviously, there are other

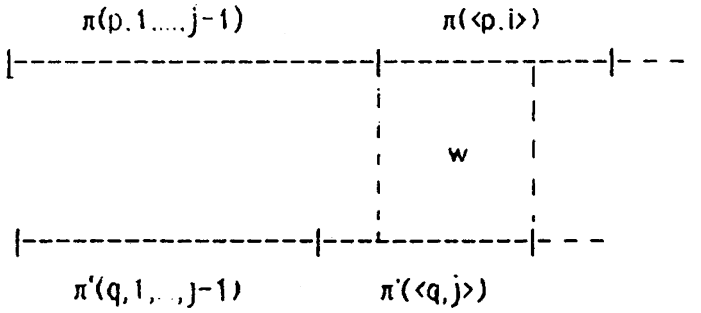

Figure 2.

cases, but for each the above statement remains true. Assume that the substring w common to $\langle p, i \rangle$ and $\langle q, j \rangle$ starts and ends in the positions $h1$ and $h2$ of $\pi(\langle p, i \rangle)$, i.e., $\pi(\langle p, i \rangle).\alpha w \beta$, where $|\alpha| = h1 - 1$ and $|w| = h2 - h1 + 1$. In this case we say that $\langle q, j \rangle$ *simulates* $\langle p, i \rangle$ *from* $h1$ to $h2$. In the case that w contains at least one module of $\pi(\langle p, i \rangle)$, one says that $\langle q, j \rangle$ is *principal* for $\langle p, i \rangle$. If this is the case and $j \in \mathrm{Occ}(q, y)$ and $i \in \mathrm{Occ}(p, x)$, y is said to be *principal* for x.

In what follows we will prove three lemmas that will enable us to show Theorem 1. Before doing this let us describe intuitively the line of thought that is followed. What we want to do is the following:

First, in Lemma 1 it is shown that each position $\langle p, i \rangle$ is simulated by at least one position $\langle q, j \rangle$ that is principal for it ([clearly, j is in $\mathrm{Issim}(i)$)].

In Lemma 2 it is shown that if an occurrence $\langle q, j \rangle$ of y is principal for an occurrence $\langle p, i \rangle$ of x, then every other occurrence of y must be principal for some other occurrence of x.

Finally, in Lemma 3 we show that each position $i \in [1, n]$ can "choose" a position $\langle q, j \rangle$ of $\mathrm{Issim}(i)$ that is principal for $\langle p, i \rangle$, and in such a way that the following holds: let $\langle p, i \rangle$ and $\langle q, j \rangle$ be occurrences of x and y, respectively, since $\langle p, i \rangle$ has chosen $\langle q, j \rangle$, every other occurrence $\langle p, i' \rangle$ of x such that an occurrence $\langle q, j' \rangle$ of y is in $\mathrm{Issim}(i')$, chooses $\langle q, j' \rangle$.

Once this is shown, it is easy to construct a substitution f such that $f(q) = p$ (thus showing Theorem 1) as follows:

(1) for all variables y of q that are never "choosen" in the above process, $f(y) = \lambda$;

(ii) for every other variable z of q, consider an occurrence $\langle q, j \rangle$ of z and let $\langle i_h, \ldots, i_{h+v} \rangle$ be the elements of $\mathrm{Sim}(j)$ that have "chosen" $\langle q, j \rangle$, if $x_0, \ldots, x_v$ are the variables occurring in $i_h, \ldots, i_{h+v}$, respectively, then $f(z) = x_0 \ldots x_v$.

LEMMA 1: *For every* $i \in [1, n]$, $\mathrm{Issim}(i)$ *contains at least one element* j *such that* $\langle q, j \rangle$ *is principal for* $\langle p, i \rangle$.

Proof: A variable y in $\mathrm{Var}(q)$ that is principal for no variable of p is such that $|\pi'(y)| \leqq 2(kL + 1)$. In fact, $\pi'(y)$ can have the forms $b^t a a b^{t''}$ or $b^t a b^{t'}$, where t and t' are at most kL, see the definition of π. Now, since $|q| = m$, $\mathrm{Issim}(i)$ is at most $\langle 1, \ldots, m \rangle$; in this case, since all its variables are non principal, the string that q can generate for simulating $\pi(\langle p, i \rangle)$ has length at most $2m(kL + 1)$. This cannot be sufficient because the length of $\pi(\langle p, i \rangle)$

is as follows: let $r = \text{ord}(x)$, where x is the variable in $\langle p, i \rangle$, then,

$$|\pi(x)| = \sum_{j=1}^{L} ((r-1)L + j + 1) + 1 = L^2(r-1) + (L(L+1)/2) + L + 1.$$

Because $|\pi(x)|$ depends on the square of L it is easy to prove that $|\pi(x)| > 2\,m(kL+1)$. Recall that $L = 6\,mk$; it suffices to consider the second term only (the first may be equal to 0 if $r=1$): we want to show that $(L(L+1)/2) > 2\,m(kL+1)$.

This is true if $L^2 > 4\,m(kL+1)$, now, $L^2 > 5\,mkL > 4\,m(kL+1)$. Thus we have a contradiction and the Lemma is true. □

LEMMA 2: *Consider two positions $\langle p, i \rangle$ and $\langle q, j \rangle$ such that the second simulates the first from $h1$ to $h2$ and it is principal for it. Let j be an element of* $\text{Occ}(q, y) = \langle j_1, \ldots, j_h \rangle$ *and i be in* $\text{Occ}(p, x)$; *for every $f \in [1, h]$, there is an element i' of* $\text{Occ}(p, x)$ *such that $\langle q, j_f \rangle$ simulates it from $h1$ to $h2$ and is principal for it.*

Proof: By definition of π, only occurrences of x produce in $\pi(p)$ a module of $\pi(x)$. Hence, if y is principal for x, every occurrence of y in q must partecipate to the simulation of an occurrence of x. This, together with property (*) shows the lemma. □

The following concept is very important for the sequel of the proof.

DEFINITION OF CHOICE : Let x be in $\text{Var}(p)$ and $\text{Occ}(p, x) = \langle i_1, \ldots, i_h \rangle$. A choice for x is a sequence $C_x = \langle j_1, \ldots, j_h \rangle$ of positions of q such that the following two conditions are satisfied:

(1) j_r is in $\text{Issim}(i_r)$ and $\langle q, j_r \rangle$ is principal for $\langle p, i_r \rangle$;

(2) let y be the variable in position $\langle q, j_r \rangle$ and assume that $\langle q, j_r \rangle$ simulates $\langle p, i_r \rangle$ from $h1$ to $h2$; for any other $i_z z, \in [1, h]$, such that $\langle p, i_z \rangle$ is simulated from $h1$ to $h2$ by an occurrence $\langle p, j \rangle$ of y, it must be that j_z is equal to j. □

The second point of the above definition may appear mysterious. Its goal is explained intuitively as follows. From a choice for each variable of p we intend to construct the substitution f such that $f(q) = p$. To this end we need that once a simulation task, e. g., simulate $\langle p, i_r \rangle$, is given to one occurrence of y, e. g., $\langle q, j_r \rangle$, that same task must be assigned to every other occurrence of y (and thus, in the above definition, $\langle q, j_z \rangle$ must simulate $\langle p, i_z \rangle$). Intuitively, this condition can be met because of Lemma 2; the formal proof is given in the following lemma.

LEMMA 3: *For each variable x in* $\text{Var}(p)$ *there is a choice for x.*

Proof: Let $\mathrm{Occ}(p, x)=\langle i_1, \ldots, i_h\rangle$ and $H=|\pi(x)|$. For each f in $[1, H]$, $\mathrm{Cut}(f)$ is the sequence $\langle j_1, \ldots, j_h\rangle$ such that for each r in $[1, h]$, $\langle q, j_r\rangle$ simulates $\langle p, i_r\rangle$ from $h1$ to $h2$ and $h1 \leqq f \leqq h2$. For proving the lemma it suffices to show that there is at least one f such that for each $r \in [1, h]$, $\langle q, j_r\rangle$ is principal for $\langle p, i_r\rangle$. That such $\mathrm{Cut}(f)$ is a choice for x is shown as follows.

$\mathrm{Cut}(f)$ satisfies trivially condition (1) of the definition of choice. It satisfies also condition (2) because otherwise the following would be true: $\mathrm{Cut}(f)$ contains two elements i_{v1} and i_{v2} of $\mathrm{Occ}(p, x)$ such that,

(i) $\langle q, j_v\rangle$ simulates $\langle p, i_{v1}\rangle$ from $h1$ to $h2$; let j_{v1} be in $\mathrm{Occ}(q, y)$;

(ii) $\langle q, j_{v2}\rangle$ simulates $\langle p, i_{v2}\rangle$ from $h1'$ to $h2'$ and j_{v2} is not in $\mathrm{Occ}(q, y)$;

(iii) there is an occurrence $\langle q, j\rangle$ of y that simulates $\langle p, i_{v2}\rangle$ from $h1$ to $h2$.

It is easy to see that this cannot be true because f is both in $[h1, h2]$ and in $[h1', h2]$ and hence, if (ii) and (iii) would be true at the same time, the f-th symbol of $\langle p, i_{v2}\rangle$ would be "simulated twice"!

An f, such that $\mathrm{Cut}(f)$ has the propriety specified above, exists because, otherwise, the non principal variables of q should generate more than H symbols and in the proof of Lemma 1 we have shown that this is not possible. □

We are finally in the condition of proving Theorem 1. For this proof we need the following notation. Consider a variable x of $\mathrm{Var}(p)$, let $\mathrm{Occ}(p, x)=\langle i_1, \ldots, i_h\rangle$ and let $C_x=\langle j_1, \ldots, j_h\rangle$ be a choice for it; let $\langle p, i\rangle$ be an occurrence of x, i.e., $i=i_r$ for some r in $[1, h]$, then with $C_x(\langle p, i\rangle)$ we denote the element j_r of C_x. Intuitively, $C_x(\langle p, i\rangle)$ is the position in q that has been chosen for simulating $\langle p, i\rangle$.

Proof of Theorem 1: Let, for x in $\mathrm{Var}(p)$, C_x be a choice for x. The definition of the substitution f such that $f(q)=p$ is as follows:

DÉFINITION OF f: For each $y \in \mathrm{Var}(q)$ one needs first to fix the notation (a),

(a) consider any occurrence $\langle q, j\rangle$ of y and let $S=\langle\langle p, i\rangle, \ldots, \langle p, i+h\rangle\rangle$ be all the positions that have chosen $\langle q, j\rangle$; formally, S is the maximal sequence of positions of p such that, for each r in $[i, j+h]$, if x is the variable occuring in $\langle p, r\rangle$, then $C_x(\langle p, r\rangle)=j$.

Now, if S is empty, then $f(y)=\lambda$, otherwise, if $x_0, \ldots, x_h$ are the variables of p occurring in the positions $i, \ldots, i+h$ of p, then $f(y)=x_0 \ldots x_h$. □

Notice that S consists of continguous positions: this is the case because if $\langle q, j \rangle$ is chosen by $\langle p, r \rangle$ and $\langle p, r+2 \rangle$, then it is the only principal position of $\langle p, r+1 \rangle$ and hence it must be chosen by $\langle p, r+1 \rangle$. Since for defining $f(y)$ just any occurrence of y is taken, the reader may wonder whether the above definition characterizes a unique substitution. This is the case because of the following reasoning (*) :

(*) When an occurrence $\langle q, j \rangle$ of y is chosen for simulating $\langle p, i \rangle$ from $h1$ to $h2$, then, by Lemma 2, every other occurrence $\langle q, j' \rangle$ of y must simulate from $h1$ to $h2$ an occurrence $\langle p, i' \rangle$ of x and then, by point (2) of the definition of Choice, $C_x(\langle p, i' \rangle) = j'$. Hence, considering j or j' for defining $f(y)$ is strictly the same.

It remains to show that $f(q) = p$. To this end remark that p can be cut into h pieces, $h \geqq 1$,

$$\langle 1, \ldots, i(1) \rangle, \langle i(1)+1, \ldots, i(2) \rangle, \langle i(h-1)+1, \ldots, i(h) \rangle$$

such that every positions in each piece has chosen the same position of q (each piece is like the sequence S in the definition of f above). Let j_r be the position of q that is chosen by the r-th piece, r in $[1, h]$. For obtaining the desired result, it suffices to observe that the definition of f and reasoning (*) imply the following two points:

(1) The positions $\langle j_1, \ldots, j_h \rangle$ are all and only the positions in q of the variables y such that $f(y) \neq \lambda$;

(2) If y is the variable in $\langle q, j_r \rangle$, r in $[1, h]$, $f(y)$ is equal to the sequence of variables corresponding to the positions in the r-th piece of p, i. e., $\langle i(r-1)+1, \ldots, i(r) \rangle$ (we assume that $i(0) = 0$). □

This result gives an exponential test for the inclusion of the languages of two pure patterns under erasing substitutions.

3. H 2 AND H 3 ARE FALSE

These negative results are easier to present than the first one because it suffices to give a counterexample for each of them.

Counterexample for H 2 (pure patterns and nonerasing substitutions): Let the terminal alphabet be $A = \{a, b\}$: $p = xyzwkmr$ and $q = xyzyw$.

$L(q)$ contains all words of length at least 5 and that can be decomposed into $w_1 w_2 w_3 w_2 w_4$, such that all w_i are non empty. For showing that $L(q) \supseteq L(p)$ observe that, if $L5$ is the set of all words of length at least 5 on

A, any w in $L5$ can be decomposed in $w_1 w_2 w_3 w_2 w_4$, where w_1 and w_4 may be empty. Since $|p|=7$, every word w of $L(p)$ can be decomposed into $w_1 w_2 w_3$, where w_2 is in $L5$ and w_1 and w_3 are not empty. Hence, w is an element of $L(q)$. It is evident that no nonerasing substitution σ exists such that $\sigma(q)=p$.

It is not difficult to generalize this example to larger alphabets A. □

Counterexample for $H3$ (any pattern and erasing substitutions): The case that $A=\{a, b\}$ is very simple: $p=xaybz$ and $q=xaby$.

Clearly, there is no erasing substitution σ such that $\sigma(q)=p$, but $L(p)=L(q)$: they both contain all words in A^* containing ab. This example is due to Ch. Codognet, [Cod].

We were not able to generalize this example to larger alphabets. A quite different counterexample is needed, for disproving $H3$, if $A=\{a, b, c\}$. For simplicity we write p and q using the extra symbol @ to denote the string *abc:*

$p=@\,aa\,@\,ba\,@\,ca\,@@\,ab\,@\,bb\,@\,cb\,@@\,tatb;$

$q=x\,@\,y\,@\,z\,@\,w\,@\,r\,@\,kykw.$

Let us show that $L(q)\supseteq L(p)$. Intuitively, the idea is that k of q cannot produce both *ta* and *tb* and hence, it must be "helped" by y and w, but y and w cannot be just a and b: they are strings of length at least 2. The last character of the string generated by t (under any substitution) can be a, b or c, thus y must have the possibility of becoming, according to the need *aa*, *ba* and *ca*, whereas w must be able to become *ab*, *bb*, *cb*. This can be done by varying accordingly the values of the variables x, z and r. More formally, consider any substitution σ and let the last character of $\sigma(t)$ be, for instance, c. Then one can define a substitution $\sigma'(q)=p$ as follows:

$\sigma'(x)=@\,aa\,@\,ba;$

$\sigma'(y)=ca;$

$\sigma'(z)=@\,ab\,@\,bb;$

$\sigma'(w)=cb;$

$\sigma'(r)=\lambda;$

$\sigma'(k)=\sigma(t)$ to which the last letter has been deleted.

Assume now that there is a substitution σ such that $\sigma(q)=p$. Any such σ must satisfy the condition that, $\sigma(kykw)=tatb$ and hence, $\sigma(k)=y$, $\sigma(y)=a$, and $\sigma(w)=b$. But, considering q, one sees that this is possible only if the first part of p contains $@\,a\,@$ and $@\,b\,@$. With $@=abc$ this is not possible. □

Observe that these negative results do not imply the undecidability of the inclusion of pattern languages in the conditions of H 2 and H 3. However, they seem to imply that any method for deciding these problems will not be simple. In [Ang] it is proved that whether $L(q) \supseteq L(p)$ for any patterns p and q under nonerasing substitutions is NP-hard. It is simple to modifiy this proof for showing the NP-hardness of the problem also in the case that erasing substitutions are considered. This proof is not included here because it is a straightforward modification of that of [1].

4. ABOUT PATTERN EQUIVALENCE

Based on the results of the previous sections, one may say that, in general, the condition that $L(q) \supseteq L(p)$ is not sufficient for showing a strict relationship between p and q. It is natural to wonder whether the condition that $L(p) = L(q)$ would then be strong enough.

Angulin in [Ang] shows the following result (*a*):

(*a*) For any two patterns p and q, $\mathrm{LN}(p) = \mathrm{LN}(q)$ iff p and q are equal modulo a variable renaming.

The proof of this result uses the obvious fact that if $\mathrm{LN}(p) = \mathrm{LN}(q)$, then $|p| = |q|$. Therefore, this proof breaks down if erasing substitutions are considered. In this case, the following results can be shown:

(*b*) If p and q are pure patterns then $L(p) = L(q)$ iff there are two substitutions σ and γ such that $\sigma(p) = q$ and $\gamma(q) = p$.

(*c*) For any two patterns p and q and for an alphabet A containing at least 3 symbols, if $L(p) = L(q)$, then p and q must be as follows:

$p = w_1 \alpha_1 w_2 \ldots w_k \alpha_k w_{k+1}$;

$q = w_1 \beta_1 w_2 \ldots w_k \beta_k w_{k+1}$;

where, for each $i \in [1, k+1]$. w_i is in A^* and for each $i \in [1, k]$, α_i β_i are in $\mathrm{Var}(p)^+$ and in $\mathrm{Var}(q)^+$, respectively. When two patterns respect the above condition, they are said to have *the same structure.*

Point (*b*) is an immediate consequence of the fact that H 1 is true. Point (*c*) is somebow a weaker version of (*a*).

Point (*c*) can be proved, roughly, as follows (this proof was suggested by [2]). First, remark that the hypothesis that A contains more than two symbols is necessary: the first counterexample for H 3, where $A = \{a, b\}$, contradicts (*c*). Consider two patterns p and q such that $L(p) = L(q)$. It is easy to see that if $t(p)$ and $t(q)$ denote the terminal strings obtained from p

and q, by deleting the variables, then $t(p)=t(q)$. It is easy to see that, since $L(p)=L(q)$, the patterns start either both with a variable or both with a terminal. Thus, if p and q contradict (c), the following situation (or the symmetric one) takes place:

$$p=\Omega w \alpha w' \Omega' \quad \text{and} \quad q=\Pi w w' \Pi',$$

$$\text{where } \Omega, \Omega', \Pi \text{ and } \Pi' \in (A \cup V)^*, \quad \alpha \in V^+, w \text{ and } w' \in A^+.$$

Assume that this is the left-most such situation. Let a be a symbol in A that is different from the last symbol of w and from the first one of w'. Let β be the substitution sending every variable of p to a. There is no β' such that $\beta'(q)=\beta(p)$. Assume, in fact, that such β' exists. Since β sends all variables of p to a, by the fact that $t(p)=t(q)$, β' must do the same. From the assumption that Ω and Π respect point (c), it follows that Ωw and Πw contain the same number (at least one) of symbols different from a; β and β' must be such that these symbols occur in corresponding places of $\beta(p)$ and $\beta'(q)$. Thus, in particular, $|\beta(\Omega w)|=|\beta'(\Pi w)|=k$. Observe now that the $k+1$-th symbol of $\beta(p)$ is a, whereas the $k+1$-th symbol of $\beta'(q)$ is the first symbol of w' that is different from a by construction. Hence, $\beta(p) \neq \beta'(q)$. □

5. CONCLUSIONS AND OPEN QUESTIONS

We have studied the problem of whether, for 2 patterns p and q, the fact that the language of q contains that of p implies the existence of a substitution f such that $f(q)=p$. This is true only in the case that p and q are pure patterns and that erasing substitutions are considered. Thus, only in this case we have an (exponential) method for deciding the inclusion of pattern languages.

The stronger hypothesis that $\mathrm{LN}(p)=\mathrm{LN}(q)$ implies the equality (modulo renaming) of p and p, whereas, under the hypothesis that $L(p)=L(q)$, we are able to prove only the equality of the structures of p and q, see point (c) in the previous section.

Several problems must still be answered:

(1) Can a stronger result than that of point (c) of the previous section be shown for any two patterns p and q such that $L(p)=L(q)$?

(2) Are there methods for deciding the inclusion of two pattern languages when the two patterns are not pure or if one considers nonerasing substitutions?

(3) In the case of erasing substitutions, define a minimal set of rules for transforming a given pattern into one of minimal length and still defining the same language.

(4) Can the results of this paper be extended to tree-patterns?

ACKNOWLEDGMENTS

I would like to tank Christian Codognet and Michel Billaud of the University of Bordeaux for stimulating conversations and helpful discussions.

REFERENCES

1. D. ANGULIN, *Finding Patterns Common to a Set of Strings*, Proc. of the 11-th ACM Symp. on the Theory of Computing, Atlanta, 1979.
2. Ch. CODOGNET, *Personal Communication*, 1986.
3. T. SHINOHARA, *Polynomial Time Inference of Pattern Languages and its Applications*, Proc. of the 7-th IBM Symp. on Math. Found. of Comp. Sci., 1982.