A non-learnable class of E-pattern languages

Daniel Reidenbach [1]

Fachbereich Informatik, Technische Universität Kaiserslautern,
Postfach 3049, 67653 Kaiserslautern, Germany

Abstract

We investigate the inferrability of E-pattern languages (also known as extended or erasing pattern languages) from positive data in Gold's learning model. As the main result, our analysis yields a negative outcome for the full class of E-pattern languages – and even for the subclass of terminal-free E-pattern languages – if the corresponding terminal alphabet consists of exactly two distinct letters. Furthermore, we present a positive result for a manifest subclass of terminal-free E-pattern languages. We point out that the considered problems are closely related to fundamental questions concerning the nondeterminism of E-pattern languages.

Key words: pattern languages, inductive inference, learning theory

1 Introduction

In the context of this paper, a pattern – a finite string that consists of variables and terminal symbols – is used as a device for the definition of a formal language. Such a pattern generates a word by a uniform substitution of all variables with arbitrary strings of terminal symbols, and, accordingly, its language is the set of all words that can be constructed by suchlike substitutions. For instance, the language generated by the pattern $\alpha = x_1\, x_1\, \mathtt{a}\, \mathtt{b}\, x_2$ (with x_1, x_2 as variables and $\mathtt{a}$, $\mathtt{b}$ as terminals) includes all words where the prefix can be split in two occurrences of the same string, followed by the string $\mathtt{ab}$ and concluded by an arbitrary suffix. Thus, the language of α contains, among others, the words $w_1 = \mathtt{a}\,\mathtt{a}\,\mathtt{a}\,\mathtt{b}\,\mathtt{a}$, $w_2 = \mathtt{a}\,\mathtt{b}\,\mathtt{a}\,\mathtt{b}\,\mathtt{a}\,\mathtt{b}\,\mathtt{a}\,\mathtt{b}$, $w_3 = \mathtt{a}\,\mathtt{b}\,\mathtt{b}\,\mathtt{b}$, whereas the following examples are not covered by α: $v_1 = \mathtt{b}\,\mathtt{a}$, $v_2 = \mathtt{b}\,\mathtt{b}\,\mathtt{b}\,\mathtt{b}\,\mathtt{b}$,

Email address: reidenba@informatik.uni-kl.de (Daniel Reidenbach).
URL: http://www-agrw.informatik.uni-kl.de/home/reidenba/ (Daniel Reidenbach).

[1] Supported by the Deutsche Forschungsgemeinschaft (DFG), Grant Wi 1638/1-2

$v_3 = \mathtt{b\,a\,a\,b\,a}$. Consequently, numerous regular and nonregular languages can be described by patterns in a compact and "natural" way.

The investigation of patterns in strings may be seen as a classical topic in the research on word monoids and combinatorics of words, examined for instance by Thue [25], [26], Bean, Ehrenfeucht, McNulty [6], Keränen [11], and many more; the definition of *pattern languages* as described above goes back to Angluin [1]. Pattern languages have been the subject of several analyses within the scope of formal language theory, e.g. by Jiang, Kinber, Salomaa, Salomaa, Yu [9],[10] – for a survey see [21]. These examinations reveal that a definition disallowing the substitution of variables with the empty word – as given by Angluin – leads to a class of languages with particular features that significantly differ from the properties of the class that results from a definition which admits the empty substitution (cf. w_3 in our example, that can only be generated by α if the empty word is assigned to x_1). Languages of the latter type have been introduced by Shinohara in 1982 (cf. [22]); they are referred to as *extended*, *erasing*, or simply *E-pattern languages*, whereas those following Angluin's definition are called *NE-pattern languages*.

When dealing with pattern languages, manifold questions arise from the problem of computing a pattern that is common to a given set of words. Therefore pattern languages have been a focus of interest of algorithmic learning theory from the very beginning. In the elementary learning model of inductive inference – known as *learning in the limit* or *Gold style learning* (introduced by Gold in 1967, cf. [8]) – a class of languages is said to be *inferrable from positive data* if and only if a computable device (the so-called *learning strategy*) – that reads growing initial segments of any text (an arbitrary stream of words that, in the limit, fully enumerates the language) – after finitely many steps converges for every language and for every corresponding text to a distinct output exactly representing the given language. In other words, the learning strategy is expected to extract a single and complete description of a (potentially infinite) language from finite data. According to [8], this task is too challenging for many well-known classes of formal languages: All superfinite classes of languages – i.e. all classes that contain every finite and at least one infinite language – such as the regular, context-free and context-sensitive languages are not inferrable from positive data. Consequently, the number of rich classes of languages that are known to be learnable is rather small. Finally, it is worth mentioning that Gold's model has been complemented by several criteria on language learning (e.g. in [2] and [28]) and, moreover, that it has been transformed into a widely analysed learning model for classes of recursive functions (e.g. [5], [12], and [3]).

The current state of knowledge concerning the learnability of pattern languages considerably differs when regarding NE- or E-pattern languages, respectively: The learnability of the class of NE-pattern languages was shown

by Angluin when introducing its definition in 1980 (cf. [1]). In the sequel there has been a variety of additional studies – e.g. by Lange, Wiehagen [13], Wiehagen, Zeugmann [27], Reischuk, Zeugmann [19] and many more (for a survey see [24]) – concerning the complexity of learning algorithms, consequences of different input data, efficient strategies for subclasses, and so on. The question, however, whether the class of E-pattern languages is inferrable from positive data, considered to be "one of the outstanding open problems in inductive inference" (Mitchell [15]), remained unresolved for two decades – apart from the positive results in [15] for the special terminal alphabets of unary and infinite size. The present paper, that has been given as a preliminary version in [16], examines this question with regard to the more significant binary terminal alphabets, and it provides a negative answer for that case. Thus, if alphabets with two distinct letters are considered, the small difference in the definitions of NE- and E-pattern languages causes the opposite results with regard to the learnability of both classes. Meanwhile, this negative finding has been extended on E-pattern languages over alphabets with three and with four letters as well (cf. [18]).

Up to the present, only very few non-trivial subclasses of E-pattern languages are known to be learnable. In detail, the class of E-pattern languages where the patterns contain at most m distinct variables (indirectly shown by Wright [28]) and the class of quasi-regular E-pattern languages, with every variable occurring exactly m times (first shown by Shinohara in [22] for $m = 1$, the general case shown by Mitchell in [15]), can be mentioned. The learnability of a third class has been claimed in [16] and is proven in Section 4. A fourth, recent positive finding on a subclass of E-pattern languages, that can be interpreted easier when the results of this paper are described completely, is noted in Section 5.

The considerations in the following sections focus on a particular subclass of E-Pattern languages, the so-called *terminal-free E-pattern languages*, which are generated by patterns that consist of variables only. These patterns, known as terminal-free or pure patterns, have been a subject of several publications within the scope of formal language theory, such as [7] and [10]. Our decision is motivated by two reasons – a rather abstract and a fairly pragmatical one. First, the approaches by Shinohara, Wright and Mitchell restrict the occurrences of variables, and therefore patterns only consisting of variables seem to allow an undisguised look at the difficulties that caused these restrictions. Second, the inclusion problem is not decidable for the full class of E-pattern languages, but it is decidable for terminal-free E-pattern languages, and this fact is a valuable aid when analysing learnability of formal languages; this circumstance is explained in the following, formal section. Since many problems considered in this paper may be interpreted as questions on the nondeterminism of pattern languages our focus on terminal-free patterns implies connections to so-called equality sets and, thus, to several examinations on words that

solve some instance of the Post Correspondence Problem (cf. [21]). Within the scope of this paper, however, these aspects are not discussed explicitly.

2 Preliminaries

In order to keep this paper largely self-contained we now introduce a number of definitions and basic properties. For standard mathematical notions and recursion-theoretic terms not defined explicitly, we refer to [20]; for unexplained aspects of formal language theory, [21] may be consulted.

We begin with some fundamental definitions on words and languages. $\mathbb{N}$ is the set of natural numbers, $\{0, 1, 2, \dots\}$. For an arbitrary set A of symbols, A^+ denotes the set of all non-empty words over A and A^* the set of all (empty and non-empty) words over A. Any set $L \subseteq A^*$ is a *language* over an alphabet A. We designate the *empty* word as e. For the word that results from the n-fold concatenation of a letter $\mathtt{a}$ or of a word w we write $\mathtt{a}^n$ or $(w)^n$, respectively. The size of a set A is denoted by $|A|$ and the length of a word w by $|w|$; $|w|_{\mathtt{a}}$ is the frequency of a letter $\mathtt{a}$ in a word w. The *Parikh vector* of a word w over a finite alphabet $A := \{\mathtt{a}_1, \mathtt{a}_2, \dots, \mathtt{a}_n\}$ is the vector $\langle |w|_{\mathtt{a}_1}, |w|_{\mathtt{a}_2}, \dots, |w|_{\mathtt{a}_n} \rangle$.

Let $(L_i)_{i\in\mathbb{N}}$ be an infinite sequence of non-empty languages. Then the *membership problem* for $(L_i)_{i\in\mathbb{N}}$ is said to be *decidable*, provided there is a total computable function that, given any pair of an index $i \in \mathbb{N}$ and a word w, decides whether or not $w \in L_i$; we say that the *inclusion problem* is decidable if there exists a total computable function that, given any pair of indices $i, j \in \mathbb{N}$, decides whether or not $L_i \subseteq L_j$. If the membership problem for $(L_i)_{i\in\mathbb{N}}$ is decidable then we call it an *indexed family (of non-empty recursive languages)*. A class $\mathcal{L}$ of languages is *indexable* if and only if there exists an indexed family $(L_i)_{i\in\mathbb{N}}$ with $\mathcal{L} = \{L_i \mid i \in \mathbb{N}\}$ – in that case we say that the membership problem for $\mathcal{L}$ is decidable. Accordingly, for class $\mathcal{L}$ of non-empty languages the inclusion problem is said to be decidable if and only if there exists a sequence $(L_i)_{i\in\mathbb{N}}$ with $\mathcal{L} = \{L_i \mid i \in \mathbb{N}\}$ such that the inclusion problem is decidable for $(L_i)_{i\in\mathbb{N}}$.

We proceed with the pattern specific terminology. Σ is a finite or infinite alphabet of *terminal* symbols and $X = \{x_1, x_2, x_3, \dots\}$ an infinite set of *variable* symbols, $\Sigma \cap X = \emptyset$. Henceforth, we use lower case letters from the beginning of the Latin alphabet as terminal symbols; words of terminal symbols are named as u, v, or w. A *pattern* is a non-empty word over $\Sigma \cup X$, a *terminal-free pattern* is a non-empty word over X; naming patterns we use lower case letters from the beginning of the Greek alphabet. $\mathrm{var}(\alpha)$ denotes the set of all variables of a pattern α. We write Pat for the set of all patterns and $\mathrm{Pat}_{\mathrm{tf}}$ for the set of all terminal-free patterns.

A *substitution* is a morphism $\sigma : (\Sigma \cup X)^* \longrightarrow \Sigma^*$ such that $\sigma(\mathtt{a}) = \mathtt{a}$ for every $\mathtt{a} \in \Sigma$. We explicitly allow the substitution of variables with the empty word. An *inverse substitution* is a morphism $\bar{\sigma} : \Sigma^* \longrightarrow X^*$. The *E-pattern language* $L_\Sigma(\alpha)$ of a pattern α is defined as the set of all $w \in \Sigma^*$ such that $\sigma(\alpha) = w$ for some substitution σ. If α is a terminal-free pattern then we call $L_\Sigma(\alpha)$ a *terminal-free E-pattern language*. For any word $w = \sigma(\alpha)$ we say that σ *generates* w, and for any language $L = L_\Sigma(\alpha)$ we say that α generates L. If there is no need to give emphasis to the concrete shape of Σ we denote the E-pattern language of a pattern α simply as $L(\alpha)$. We use ePAT as an abbreviation for the full class of E-pattern languages and $\mathrm{ePAT}_{\mathrm{tf}}$ for the class of terminal-free E-pattern languages. For any class $\mathrm{ePAT}^\star$ of E-pattern languages we write $\mathrm{ePAT}^\star_\Sigma$ if the corresponding alphabet is of interest.

Clearly, both ePAT and $\mathrm{ePAT}_{\mathrm{tf}}$ are indexable since, first, every E-pattern language is non-empty, second, a recursive enumeration of all necessary patterns can be constructed with little effort and, third, the decidability of the membership problem for any pattern $\alpha \in \mathrm{Pat}$ and word $w \in \Sigma^*$ is guaranteed as the search space for a successful substitution of α is bounded by the length of w. With regard to the inclusion problem, however, ePAT and $\mathrm{ePAT}_{\mathrm{tf}}$ are different. In [10] it is shown that, in general, the inclusion problem for ePAT is undecidable, whereas for $\mathrm{ePAT}_{\mathrm{tf}}$ the opposite holds true. As this is of great importance for the following examinations, we now cite two corresponding theorems:

Fact 1 (Jiang, Salomaa, Salomaa, Yu [10]) *Let Σ be an alphabet, $|\Sigma| \geq 2$, and α, β arbitrarily given terminal-free patterns. Then $L_\Sigma(\beta) \subseteq L_\Sigma(\alpha)$ iff there exists a morphism $\phi : X^* \longrightarrow X^*$ such that $\phi(\alpha) = \beta$.*

Fact 2 (ibid.) *The inclusion problem for* $\mathrm{ePAT}_{\mathrm{tf}}$ *is decidable.*

We conclude our notions on E-pattern languages with the naming of some important properties of particular patterns. A pattern α is in *canonical form* if and only if, for some $n \geq 1$, $\mathrm{var}(\alpha) = \{x_1, x_2, \ldots, x_n\}$ and, additionally, for every x_i, $1 \leq i < n$, the leftmost occurrence of x_i in α is to the left of the leftmost occurrence of x_{i+1}; for instance, the pattern $x_1\, x_2\, x_1\, x_3\, x_2$ is in canonical form, whereas $x_1\, x_2\, x_4$ and $x_1\, x_3\, x_2\, x_3$ are not.

Following [15], we designate a pattern α as *succinct* if and only if $|\alpha| \leq |\beta|$ for all patterns β with $L(\beta) = L(\alpha)$. The pattern $\beta = x_1x_2x_1x_2$, for instance, generates the same language as the pattern $\alpha = x_1x_1$, and therefore β is not succinct; α is succinct because there does not exist any shorter pattern than α that exactly describes its language.

According to the studies of Mateescu and Salomaa on the nondeterminism of pattern languages (cf. [14]) we denote a word w as *ambiguous* (in respect of a pattern α) if and only if there exist two substitutions σ and σ' such

that $\sigma(\alpha) = w = \sigma'(\alpha)$, but $\sigma(x_i) \neq \sigma'(x_i)$ for some $x_i \in \mathrm{var}(\alpha)$. The word $w = \mathtt{aaba}$, for instance, is ambiguous in respect of the pattern $\alpha = x_1\mathtt{a}x_2$ since it can be generated by several substitutions, such as σ and σ' with $\sigma(x_1) = \mathtt{a}$, $\sigma(x_2) = \mathtt{ba}$ and $\sigma'(x_1) = e$, $\sigma'(x_2) = \mathtt{aba}$. We call a word *unambiguous* (in respect of a pattern α) if it is not ambiguous.

We proceed with the learning theoretical definitions. Our learning model goes back to Gold [8], but, since we restrict ourselves to considerations on indexable classes, we largely follow Angluin [2]. Our learner is expected to deal with positive data exclusively, given as *text*. A text for an arbitrary language L is any total function $t : \mathbb{N} \longrightarrow \Sigma^*$ satisfying $\{t(n) \mid n \in \mathbb{N}\} = L$. For any text t, any $n \in \mathbb{N}$ and a symbol $\diamond \notin \Sigma$, $t^n \in (\Sigma \cup \{\diamond\})^+$ is a coding of the first $n+1$ values of t, i.e. $t^n := t(0) \diamond t(1) \diamond t(2) \ldots \diamond t(n)$. The learner is any total computable function S (the so-called *learning strategy*) that, for a given text t, successively reads t^0, t^1, t^2, etc. and returns a corresponding stream of natural numbers $S(t^0)$, $S(t^1)$, $S(t^2)$, and so on. For a language L_j in an indexed family $(L_i)_{i\in\mathbb{N}}$ and a text t for L_j, we say that S *identifies* L_j *from* t if and only if there exist natural numbers n_0 and j' such that, for every $n \geq n_0$, $S(t^n) = j'$ and, additionally, $L_{j'} = L_j$. An indexed family $(L_i)_{i\in\mathbb{N}}$ is *learnable (in the limit)* – or: *inferrable from positive data*, or: $(L_i)_{i\in\mathbb{N}} \in$ LIM-TEXT for short – if and only if there is a learning strategy S identifying each language in $(L_i)_{i\in\mathbb{N}}$ from any corresponding text. Finally, we call an indexable class $\mathcal{L}$ of languages learnable (in the limit) or inferrable from positive data if and only if there is a learnable indexed family $(L_i)_{i\in\mathbb{N}}$ with $\mathcal{L} = \{L_i \mid i \in \mathbb{N}\}$. In this case we write $\mathcal{L} \in$ LIM-TEXT for short.

As mentioned in the previous paragraph, this specific learning model is just a special case of Gold's learning model, which can be considered for more general applications as well. Indeed, there is a large number of publications where the elements of the above definition are modified or generalised, such as the objects to be learned (e.g., using arbitrary classes of languages instead of indexed families), the learning goal (e.g., asking for a semantic instead of a syntactic convergence), or the output of the learner (choosing some general *hypothesis space* instead of the indexed family itself). However, with regard to our negative main result (cf. Theorem 5), we state that it holds in many well-established, more general variants of Gold's learning model as well. For information on suchlike aspects, see [29] and [4].

Angluin has introduced some criteria on indexed families that reduce learnability to a particular language theoretical aspect (cf. [2]). For the proof of our main result, we use the following (combining Condition 2 and Corollary 1 of the referenced paper):

Fact 3 (Angluin [2]) *Let $(L_i)_{i\in\mathbb{N}}$ be an arbitrary indexed family of non-empty recursive languages. If $(L_i)_{i\in\mathbb{N}} \in$ LIM-TEXT then for every $j \in \mathbb{N}$*

there exists a set T_j such that

- $T_j \subseteq L_j$,
- T_j *is finite, and*
- *there does not exist a* $j' \in \mathbb{N}$ *with* $T_j \subseteq L_{j'} \subset L_j$.

If there exists a set T_j satisfying the conditions of Fact 3 then it is called a telltale *(for L_j) (in respect of $(L_i)_{i\in\mathbb{N}}$).*

The importance of telltales – that, at first glance, do not show any connection to the learning model – is caused by the need of avoiding *overgeneralisation* in the inference process, i.e. the case that the strategy outputs an index of a language which is a proper superset of the language to be learned and therefore, as the input consists of positive data only, is unable to detect its mistake. Thus, every language L_j in a learnable indexed family necessarily contains a finite set of words which, in the context of the indexed family, may be interpreted as a signal distinguishing the language from all languages that are subsets of L_j.

If the inclusion problem for the examined indexed family is decidable then this necessary condition for learnability is sufficient, too. This finding again derives from [2] (combining Condition 2, Condition 4 and Corollary 3):

Fact 4 (Angluin [2]) *Let $(L_i)_{i\in\mathbb{N}}$ be an arbitrary indexed family of non-empty recursive languages such that the inclusion problem for $(L_i)_{i\in\mathbb{N}}$ is decidable. Then* $(L_i)_{i\in\mathbb{N}} \in$ LIM-TEXT *iff for every $j \in \mathbb{N}$ there exists a set T_j such that*

- $T_j \subseteq L_j$,
- T_j *is finite, and*
- *there does not exist a* $j' \in \mathbb{N}$ *with* $T_j \subseteq L_{j'} \subset L_j$.

Consequently, Fact 4 can be used for analysing the learnability of ePAT_{tf} as the inclusion problem for ePAT_{tf} is decidable (cf. Fact 2).

With these criteria we can conclude the section of definitions and preliminary results and proceed to the main result of this paper.

3 The Main Result

As mentioned in Section 1, the full class of E-pattern languages is known to be learnable in case of a unary or an infinite alphabet (cf. [15]). However, since these special alphabets considerably ease the construction of telltales, the respective reasoning has not been extendable on finite alphabets of different

size (that, in turn, normally are considered to be more interesting). For our approach to the long-term open question of the learnability of ePAT over finite alphabets with more than one letter, we restrict ourselves to binary alphabets. This assumption facilitates the main result which provides a negative answer and thus remarkably contrasts with the outcome for NE-pattern languages (cf. Section 1):

Theorem 5 *Let Σ be an alphabet, $|\Sigma| = 2$. Then* $\mathrm{ePAT}_\Sigma \notin$ LIM-TEXT.

In the following section we give the proof of Theorem 5. For this purpose, we present a specific and simply structured terminal-free pattern $\alpha_{\mathtt{ab}}$ whose language, for $|\Sigma| = 2$, has no telltale in respect of ePAT_Σ, and we point out that this is caused by the ambiguity of some particular words in $L_\Sigma(\alpha_{\mathtt{ab}})$. Moreover, our reasoning implies that even the subclass of terminal-free E-pattern languages is not learnable in the limit for binary alphabets.

3.1 Proof of the main result

To begin with we name a special type of patterns that is as useful for the upcoming line of reasoning as it is inconvenient for the needs of inductive inference:

Definition 6 (Passe-partout) *Let α be a pattern and $W \subset L(\alpha)$ a finite set of words. Let β be a pattern, such that*

- *$W \subseteq L(\beta)$ and*
- *$L(\beta) \subset L(\alpha)$.*

We then say that β is a passe-partout (for α and W).

Note that if there exists a passe-partout β for a pattern α and a set of words W, then W is not a telltale for $L(\alpha)$ in respect of any class of E-pattern languages that contains both $L(\alpha)$ and $L(\beta)$.

Definition 6 allows us to formulate the following lemma, that is crucial for the proof of Theorem 5:

Lemma 7 *Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ be an alphabet. Then for the pattern*

$$\alpha_{\mathtt{ab}} := x_1\, x_1\, x_2\, x_2\, x_3\, x_3$$

and for any finite $W \subset L_\Sigma(\alpha_{\mathtt{ab}})$ there exists a terminal-free passe-partout.

PROOF. If W is empty then the claim of Lemma 7 is trivially true. Given an arbitrary non-empty $W = \{w_1, w_2, \ldots, w_n\} \subset L(\alpha_{\mathtt{ab}})$, the following procedure constructs a passe-partout β:

As an inverse substitution we define for every w_i a morphism $\bar{\sigma}_i : \Sigma^* \longrightarrow X^*$ by

$$\bar{\sigma}_i(\mathtt{c}) := \begin{cases} x_{2i-1} & , \quad \mathtt{c} = \mathtt{a}, \\ x_{2i} & , \quad \mathtt{c} = \mathtt{b}. \end{cases}$$

As $W \subset L(\alpha_{\mathtt{ab}})$, for every w_i, $1 \leq i \leq n$, there exists a substitution σ_i satisfying $\sigma_i(\alpha_{\mathtt{ab}}) = w_i$. Constructing a set of $3n$ strings $\gamma_{i,k} \in X^*$ we now identify the necessary elements of β.

Case (i) $\sigma_i(x_3)$ contains a letter exactly once and w_i contains this letter exactly twice.
Formally, that is $\sigma_i(x_1), \sigma_i(x_2) \in \{\mathtt{b}\}^*$ and $\sigma_i(x_3) = v_1 \mathtt{a} v_2$ with $v_1, v_2 \in \{\mathtt{b}\}^*$ or $\sigma_i(x_1), \sigma_i(x_2) \in \{\mathtt{a}\}^*$ and $\sigma_i(x_3) = v_1\, \mathtt{b}\, v_2$ with $v_1, v_2 \in \{\mathtt{a}\}^*$. In this case we define

$$\begin{aligned} \gamma_{i,1} &:= \bar{\sigma}_i\left(\sigma_i(x_1)\, \sigma_i(x_2)\right), \\ \gamma_{i,2} &:= \bar{\sigma}_i\left(\sigma_i(x_3)\right), \\ \gamma_{i,3} &:= e. \end{aligned}$$

Note that w_i necessarily is ambiguous in the present case, and therefore the above definition provides a pattern $\gamma_i := \gamma_{i,1}\, \gamma_{i,1}\, \gamma_{i,2}\, \gamma_{i,2}\, \gamma_{i,3}\, \gamma_{i,3}$ with $w_i \in L(\gamma_i)$.

Case (ii) Not (i).
In other words, $\sigma_i(x_3)$ is empty or w_i contains every letter of $\sigma_i(x_3)$ at least four times. In this case we simply define

$$\gamma_{i,k} := \bar{\sigma}_i\left(\sigma_i(x_k)\right), \ 1 \leq k \leq 3.$$

Obviously, (ii) also provides a pattern $\gamma_i := \gamma_{i,1}\, \gamma_{i,1}\, \gamma_{i,2}\, \gamma_{i,2}\, \gamma_{i,3}\, \gamma_{i,3}$ with $w_i \in L(\gamma_i)$.

Combining the fragments of all γ_i in an appropriate manner we now compose the resulting pattern of the procedure:

$$\beta := \underbrace{\gamma_{1,1}\, \gamma_{2,1}\, \cdots\, \gamma_{n,1}}_{\sim x_1}\ \underbrace{\gamma_{1,1}\, \gamma_{2,1}\, \cdots\, \gamma_{n,1}}_{\sim x_1}\ \underbrace{\gamma_{1,2}\, \gamma_{2,2}\, \cdots\, \gamma_{n,2}}_{\sim x_2}\ \underbrace{\gamma_{1,2}\, \gamma_{2,2}\, \cdots\, \gamma_{n,2}}_{\sim x_2}$$
$$\underbrace{\gamma_{1,3}\, \gamma_{2,3}\, \cdots\, \gamma_{n,3}}_{\sim x_3}\ \underbrace{\gamma_{1,3}\, \gamma_{2,3}\, \cdots\, \gamma_{n,3}}_{\sim x_3}.$$

Note that, in general, β is neither in canonical form nor succinct.

In order to conclude the proof we now show that β indeed is a passe-partout for $\alpha_{\mathtt{ab}}$ and W:

(1) We define a substitution $\sigma_i' : X^* \longrightarrow \Sigma^*$ by

$$\sigma_i'(x_j) := \begin{cases} \mathtt{a} & , \quad j = 2i-1, \\ \mathtt{b} & , \quad j = 2i, \\ e & , \quad \text{else.} \end{cases}$$

Obviously $\sigma_i'(\beta) = w_i$, and thus $W \subseteq L(\beta)$.

(2) $\alpha_{\mathtt{ab}}$ and β both are terminal-free, and, because of the above depicted shape of these patterns, there exists a morphism $\phi : X^* \longrightarrow X^*$ with $\phi(\alpha_{\mathtt{ab}}) = \beta$, namely $\phi(x_j) = \gamma_{1,j}\ \gamma_{2,j}\ \cdots\ \gamma_{n,j}$ for every $x_j \in \mathrm{var}(\alpha_{\mathtt{ab}})$. Thus, $L(\beta)$ is a subset of $L(\alpha_{\mathtt{ab}})$ (according to the inclusion criterion described in Fact 1).
We now prove that $L(\beta)$ is a proper subset of $L(\alpha_{\mathtt{ab}})$. For that purpose, assume to the contrary there is a morphism $\psi : X^* \longrightarrow X^*$ such that $\psi(\beta) = \alpha_{\mathtt{ab}}$. As, due to the existence of ϕ, every variable of β occurs at least twice, there exist two morphisms ψ' and ψ'' such that $\psi(\beta) = \psi''(\psi'(\beta)) = \alpha_{\mathtt{ab}}$ and

$$\psi'(x_j) = \begin{cases} e & , \quad |\beta|_{x_j} > 2, \\ x_j & , \quad |\beta|_{x_j} = 2, \end{cases}$$

for $x_j \in \mathrm{var}(\beta)$. Consequently, ψ' replaces – possibly among others – all variables in $\phi(x_3)$ with the empty word since these variables occur at least four times in β (cf. definitions of cases (i) and (ii)), and therefore

$$\beta' := \psi'(\beta) = \overbrace{x_{j_1}\ x_{j_2}\ \cdots\ x_{j_p}}^{\psi'(\phi(x_1))}\ \overbrace{x_{j_1}\ x_{j_2}\ \cdots\ x_{j_p}}^{\psi'(\phi(x_1))}\ \overbrace{x_{j_{p+1}}\ x_{j_{p+2}}\ \cdots\ x_{j_{p+q}}}^{\psi'(\phi(x_2))}\ \overbrace{x_{j_{p+1}}\ x_{j_{p+2}}\ \cdots\ x_{j_{p+q}}}^{\psi'(\phi(x_2))}\ \overbrace{e}^{\psi'(\phi(x_3))}\ \overbrace{e}^{\psi'(\phi(x_3))}$$

with $p, q \geq 0$ and $x_{j_k} \neq x_{j_l}$ for $k \neq l$, $1 \leq k \leq p+q$, $1 \leq l \leq p+q$. However, when regarding all patterns α that are in canonical form and that can be derived from β' by any morphism such that the Parikh vector of α equals that of $\alpha_{\mathtt{ab}}$, we obviously receive the following list: $x_1x_2x_3x_1x_2x_3$ (for $p+q \geq 1$), $x_1x_2x_1x_2x_3x_3$, and $x_1x_1x_2x_3x_2x_3$ (for $p \geq 1$ and $q \geq 1$). Consequently, since $\alpha_{\mathtt{ab}}$ is in canonical form and since no pattern in the above list equals $\alpha_{\mathtt{ab}}$, there is no morphism ψ'' such that $\psi''(\beta') = \alpha_{\mathtt{ab}}$; this is a contradiction. Thus, $L(\alpha_{\mathtt{ab}})$ is not a subset of $L(\beta)$, and therefore $L(\beta) \subset L(\alpha_{\mathtt{ab}})$. $\square$

Clearly, the proof of Lemma 7 can be adapted to infinitely many succinct terminal-free patterns such as $x_1^2\,x_2^2\,x_3^2\,x_4^2$, $x_1^2\,x_2^2\,x_3^2\,x_4^2\,x_5^2$, and so on. However, we claim that there is no shorter pattern than $\alpha_{\mathtt{ab}}$ with the feature described above. Furthermore, it seems worth mentioning that the procedure given in the proof of Lemma 7 is not the only way to construct a passe-partout: With little effort the cases (i) and (ii) can be modified such that the variable $x_1 \in \mathrm{var}(\alpha_{\mathtt{ab}})$ takes the role of x_3 and vice versa, leading to a different passe-partout β' for every set of words W that contains at least one element satisfying the condition of case (i).

The following example illustrates the most relevant elements of the proof:

Example 8 *Let* $W := \{w_1, w_2, w_3, w_4\} \subseteq L(\alpha_{\mathtt{ab}})$ *be given by*

$$w_1 := \underbrace{\mathtt{a}}_{\sigma_1(x_1)}\ \underbrace{\mathtt{a}}_{\sigma_1(x_1)}\ \underbrace{\mathtt{b}}_{\sigma_1(x_2)}\ \underbrace{\mathtt{b}}_{\sigma_1(x_2)}\ \underbrace{\mathtt{b}}_{\sigma_1(x_3)}\ \underbrace{\mathtt{b}}_{\sigma_1(x_3)},$$

$$w_2 := \underbrace{\mathtt{bb}}_{\sigma_2(x_1)}\ \underbrace{\mathtt{bb}}_{\sigma_2(x_1)}\ \underbrace{\mathtt{ab}}_{\sigma_2(x_2)}\ \underbrace{\mathtt{ab}}_{\sigma_2(x_2)}\ \underbrace{\mathtt{b}}_{\sigma_2(x_3)}\ \underbrace{\mathtt{b}}_{\sigma_2(x_3)},$$

$$w_3 := \underbrace{\mathtt{b}}_{\sigma_3(x_1)}\ \underbrace{\mathtt{b}}_{\sigma_3(x_1)}\ \underbrace{\mathtt{bb}}_{\sigma_3(x_2)}\ \underbrace{\mathtt{bb}}_{\sigma_3(x_2)}\ \underbrace{\mathtt{bab}}_{\sigma_3(x_3)}\ \underbrace{\mathtt{bab}}_{\sigma_3(x_3)},$$

$$w_4 := \underbrace{\mathtt{ab}}_{\sigma_4(x_1)}\ \underbrace{\mathtt{ab}}_{\sigma_4(x_1)}\ \underbrace{\mathtt{bb}}_{\sigma_4(x_2)}\ \underbrace{\mathtt{bb}}_{\sigma_4(x_2)}\ \underbrace{\mathtt{bab}}_{\sigma_4(x_3)}\ \underbrace{\mathtt{bab}}_{\sigma_4(x_3)}.$$

Evidently, w_3 *satisfies the condition of case (i), whereas the other words satisfy case (ii). Consequently, the pattern fragments* γ_i *have the following shape:*

$$\gamma_1 = \overbrace{\underbrace{x_1}_{\bar\sigma_1(\mathtt{a})}}^{\gamma_{1,1}}\ \overbrace{\underbrace{x_1}_{\bar\sigma_1(\mathtt{a})}}^{\gamma_{1,1}}\ \overbrace{\underbrace{x_2}_{\bar\sigma_1(\mathtt{b})}}^{\gamma_{1,2}}\ \overbrace{\underbrace{x_2}_{\bar\sigma_1(\mathtt{b})}}^{\gamma_{1,2}}\ \overbrace{\underbrace{x_2}_{\bar\sigma_1(\mathtt{b})}}^{\gamma_{1,3}}\ \overbrace{\underbrace{x_2}_{\bar\sigma_1(\mathtt{b})}}^{\gamma_{1,3}},$$

$$\gamma_2 = \overbrace{\underbrace{x_4x_4}_{\bar\sigma_2(\mathtt{bb})}}^{\gamma_{2,1}}\ \overbrace{\underbrace{x_4x_4}_{\bar\sigma_2(\mathtt{bb})}}^{\gamma_{2,1}}\ \overbrace{\underbrace{x_3x_4}_{\bar\sigma_2(\mathtt{ab})}}^{\gamma_{2,2}}\ \overbrace{\underbrace{x_3x_4}_{\bar\sigma_2(\mathtt{ab})}}^{\gamma_{2,2}}\ \overbrace{\underbrace{x_4}_{\bar\sigma_2(\mathtt{b})}}^{\gamma_{2,3}}\ \overbrace{\underbrace{x_4}_{\bar\sigma_2(\mathtt{b})}}^{\gamma_{2,3}},$$

$$\gamma_3 = \overbrace{\underbrace{x_6x_6x_6}_{\bar\sigma_3(\mathtt{b\,bb})}}^{\gamma_{3,1}}\ \overbrace{\underbrace{x_6x_6x_6}_{\bar\sigma_3(\mathtt{b\,bb})}}^{\gamma_{3,1}}\ \overbrace{\underbrace{x_6x_5x_6}_{\bar\sigma_3(\mathtt{bab})}}^{\gamma_{3,2}}\ \overbrace{\underbrace{x_6x_5x_6}_{\bar\sigma_3(\mathtt{bab})}}^{\gamma_{3,2}}\ \overbrace{e}^{\gamma_{3,3}}\ \overbrace{e}^{\gamma_{3,3}},$$

$$\gamma_4 = \overbrace{\underbrace{x_7x_8}_{\bar\sigma_4(\mathtt{ab})}}^{\gamma_{4,1}}\ \overbrace{\underbrace{x_7x_8}_{\bar\sigma_4(\mathtt{ab})}}^{\gamma_{4,1}}\ \overbrace{\underbrace{x_8x_8}_{\bar\sigma_4(\mathtt{bb})}}^{\gamma_{4,2}}\ \overbrace{\underbrace{x_8x_8}_{\bar\sigma_4(\mathtt{bb})}}^{\gamma_{4,2}}\ \overbrace{\underbrace{x_8x_7x_8}_{\bar\sigma_4(\mathtt{bab})}}^{\gamma_{4,3}}\ \overbrace{\underbrace{x_8x_7x_8}_{\bar\sigma_4(\mathtt{bab})}}^{\gamma_{4,3}}.$$

Hence, the passe-partout for $\alpha_{\mathtt{ab}}$ *and* W *reads*

$$\beta = \underbrace{\overbrace{x_1}^{\gamma_{1,1}}\overbrace{x_4x_4}^{\gamma_{2,1}}\overbrace{x_6x_6x_6}^{\gamma_{3,1}}\overbrace{x_7x_8}^{\gamma_{4,1}}}_{\phi(x_1)}\underbrace{\overbrace{x_1}^{\gamma_{1,1}}\overbrace{x_4x_4}^{\gamma_{2,1}}\overbrace{x_6x_6x_6}^{\gamma_{3,1}}\overbrace{x_7x_8}^{\gamma_{4,1}}}_{\phi(x_1)}$$

$$\underbrace{\overbrace{x_2}^{\gamma_{1,2}}\overbrace{x_3x_4}^{\gamma_{2,2}}\overbrace{x_6x_5x_6}^{\gamma_{3,2}}\overbrace{x_8x_8}^{\gamma_{4,2}}}_{\phi(x_2)}\underbrace{\overbrace{x_2}^{\gamma_{1,2}}\overbrace{x_3x_4}^{\gamma_{2,2}}\overbrace{x_6x_5x_6}^{\gamma_{3,2}}\overbrace{x_8x_8}^{\gamma_{4,2}}}_{\phi(x_2)}$$

$$\underbrace{\overbrace{x_2}^{\gamma_{1,3}}\overbrace{x_4}^{\gamma_{2,3}}\overbrace{x_8x_7x_8}^{\gamma_{4,3}}}_{\phi(x_3)}\underbrace{\overbrace{x_2}^{\gamma_{1,3}}\overbrace{x_4}^{\gamma_{2,3}}\overbrace{x_8x_7x_8}^{\gamma_{4,3}}}_{\phi(x_3)}.$$

Obviously, $W \subseteq L(\beta)$ *and* $L(\beta) \subseteq L(\alpha_{\mathtt{ab}})$*. In addition note that, for all* $x_j \in \mathrm{var}(\phi(x_3))$, $|\beta|_{x_j} \geq 4$*. In order to show that* $L(\beta)$ *is a proper subset of* $L(\alpha_{\mathtt{ab}})$*, we state without proof that, e.g.,* $\mathtt{a\,a\,b\,b\,aa\,aa} \in L(\alpha_{\mathtt{ab}}) \setminus L(\beta)$*.*

Evidently, every variable of $\alpha_{\mathtt{ab}}$ occurs exactly twice. Therefore its language belongs to the class of quasi-regular E-pattern languages (cf. Section 1) that – according to Mitchell (cf. [15]) – is learnable in the limit. Nevertheless, the findings of Mitchell and Lemma 7 are consistent as β not necessarily is quasi-regular. Consequently, our result promotes the interpretation that the quasi-regular E-pattern languages are learnable because they do not include all possible passe-partouts and not on account of their shape as such.

Referring to the necessary condition for the learnability of indexed families given in Fact 3 the consequence of Lemma 7 can be stated with little effort:

Theorem 9 *Let* Σ *be an alphabet,* $|\Sigma| = 2$*. Then* $\mathrm{ePAT}_{\mathrm{tf},\Sigma} \notin$ LIM-TEXT*.*

PROOF. Lemma 7 provides a terminal-free pattern $\alpha_{\mathtt{ab}}$, such that for any finite set $W \subset L_\Sigma(\alpha_{\mathtt{ab}})$ there exists a terminal-free passe-partout β. Obviously, every set of patterns generating $\mathrm{ePAT}_{\mathrm{tf}}$ needs to contain two patterns α' and β' such that $L(\alpha_{\mathtt{ab}}) = L(\alpha')$ and $L(\beta) = L(\beta')$. Therefore, no indexed family for the class of terminal-free E-pattern languages satisfies Angluin's Condition 2 (cf. [2]), and according to Fact 3 it is not learnable in the limit. □

With this negative result for the subclass of terminal-free E-pattern languages, Theorem 5 is proven immediately.

We conclude this section with some additional remarks on the role of ambiguity of words in the proof of Lemma 7: With the capability of a suitable inverse substitution in mind, we assume that, when the full classes ePAT and $\mathrm{ePAT}_{\mathrm{tf}}$ are considered, any telltale for an E-pattern language has to include words generated by a substitution containing a unique letter (see case (i) in the proof of Lemma 7). If the alphabet consists of just two letters – as taken

into consideration in the present section – these specific words may turn out to be ambiguous, leading to a decisive loss of significance. We claim that if the words of Example 8 were unambiguous, then these words would work as a telltale for $\alpha_{\mathtt{ab}}$ – and, in fact, even the set $\{w_1, w_2, w_3\}$ would be sufficient. Thus, we consider it beneficial for learnability analyses to ask for the existence of appropriate unambiguous words in E-pattern languages, a question that is closely connected to the research on so-called equality sets (and, therefore, on the Post Correspondence Problem, cf. [21]). In the following section, as a demonstration of this approach, we utilise unambiguous words for a minor positive learnability result. Since we examine an appropriate subclass of terminal-free E-pattern languages, we even may allow binary alphabets in this case. For the full class of terminal-free E-pattern languages and for alphabets with at least three distinct letters, however, meanwhile a similar method – that is not based on unambiguous words, but on those with some "bounded" ambiguity – has led to a positive result (cf. [17]).

4 Unambiguous Words and the Learnability of Terminal-Free Non-Cross E-pattern Languages

The outcome of Section 3 entails the finding that all positive results on inductive inference of E-pattern languages cited in Section 1 follow the only practicable course: any learnable (sub-)class of these languages has to be provided with appropriate restrictions on the shape of the variables or of the terminal alphabet.

According to these demands, the present section proves the learnability of a natural subclass of terminal-free E-pattern languages for arbitrary terminal alphabets. We refer to the following set of patterns, that analogously has been considered by Shinohara with regard to NE-pattern languages (cf. [23]):

Definition 10 (Terminal-free non-cross patterns) *A pattern α is a* terminal-free non-cross pattern *iff it satisfies*

$$\alpha = x_1^{r_1}\, x_2^{r_2}\, x_3^{r_3}\, \cdots x_n^{r_n}$$

for some n and numbers $r_1, r_2, \cdots, r_n$ with $n \geq 1$ and $r_i \geq 1$, $1 \leq i \leq n$. We denote a language L as terminal-free non-cross E-pattern language *if $L = L(\alpha)$ for some terminal-free non-cross pattern α.*

We designate the set of all terminal-free non-cross patterns as $\mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc}}$ *and the class of all terminal-free non-cross E-pattern languages as* $\mathrm{ePAT}_{\mathrm{tf}}^{\mathrm{nc}}$. $\mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>} \subset \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc}}$ *is the set of those patterns with $r_i \geq 2$ for every i, $1 \leq i \leq n$.*

The separate naming in Definition 10 of those terminal-free non-cross patterns

that contain every of their variables at least twice is motivated by the following fact: Obviously, $\mathrm{ePAT}^{\mathrm{nc}}_{\mathrm{tf}} = \{L(\alpha) \mid \alpha \in \mathrm{Pat}^{\mathrm{nc},>}_{\mathrm{tf}} \cup \{x_1\}\}$ since, for all terminal-free non-cross patterns β that are not contained in $\mathrm{Pat}^{\mathrm{nc},>}_{\mathrm{tf}}$, $L(\beta)$ equals $L(x_1)$. For the latter language we can easily give a telltale, e.g., by the set $\{\mathtt{a}\}$ for any letter $\mathtt{a}$ in the corresponding terminal alphabet. Consequently, when examining the learnability of the class of terminal-free non-cross E-pattern languages, we may focus on patterns in $\mathrm{Pat}^{\mathrm{nc},>}_{\mathrm{tf}}$ – and, in fact, for the specific argumentation in the present section, this restriction even is mandatory as we implicitly require succinctness of patterns. We state without proof that this holds for every pattern in $\mathrm{Pat}^{\mathrm{nc},>}_{\mathrm{tf}}$, whereas the patterns in $\mathrm{Pat}^{\mathrm{nc}}_{\mathrm{tf}} \setminus (\mathrm{Pat}^{\mathrm{nc},>}_{\mathrm{tf}} \cup \{x_1\})$ evidently are not succinct.

Before we present our result on the learnability of $\mathrm{ePAT}^{\mathrm{nc}}_{\mathrm{tf}}$, it seems worth mentioning – with the remark on $\alpha_{\mathtt{ab}}$ being quasi-regular in mind (cf. Section 3.1) – that $\alpha_{\mathtt{ab}}$ obviously is non-cross, as well. So this section features a second example of a class of E-pattern languages that in fact only is learnable because possible passe-partouts are not contained in the class. This aspect can directly be detected in the upcoming proof of Theorem 14.

We begin with a notion that is motivated by technical reasons:

Definition 11 (Uniform Substring) *Let Σ be an alphabet with $|\Sigma| \geq 2$ and let w be a non-empty word, $w = v_1\, u\, v_2$ with $u \in \Sigma^+$, $v_1, v_2 \in \Sigma^*$. Then we call u a* uniform substring (over $\mathtt{a}$) *iff $u \in \{\mathtt{a}\}^+$ for an arbitrary $\mathtt{a} \in \Sigma$ and v_1 does not end with $\mathtt{a}$ and v_2 does not start with $\mathtt{a}$.*

Example 12 *In this example word all uniform substrings are marked:*

$$\underbrace{\mathtt{aaa}}\ \underbrace{\mathtt{bb}}\ \underbrace{\mathtt{a}}\ \underbrace{\mathtt{ccc}}.$$

As mentioned in Section 3.1, the proof of the Theorem 14 is based on specific unambiguous words, that are due to Sandra Zilles:

Lemma 13 *Let Σ be an alphabet, $|\Sigma| = 2$. Then for every $\alpha \in \mathrm{Pat}^{\mathrm{nc},>}_{\mathrm{tf}}$ there exists an unambiguous word over Σ.*

PROOF. According to Definition 10, $\alpha = x_1^{r_1}\, x_2^{r_2}\, x_3^{r_3} \cdots x_n^{r_n}$ for an $n \geq 1$ and $r_1, r_2, \ldots, r_n \in \mathbb{N}$, $r_i \geq 2$ for all i with $1 \leq i \leq n$. Let the substitution σ^{nc} be given by

$$\sigma^{\mathrm{nc}}(x_j) := \mathtt{a}\,\mathtt{b}^j$$

for all $x_j \in \mathrm{var}(\alpha)$. Then obviously $\sigma^{\mathrm{nc}}(\alpha) = (\mathtt{ab})^{r_1}\,(\mathtt{abb})^{r_2} \cdots (\mathtt{ab}^n)^{r_n}$ and, thus, the sequence of the lengths of the uniform substrings over $\mathtt{b}$ in $\sigma^{\mathrm{nc}}(\alpha)$ (from the left to the right) is monotonic increasing. We show that $\sigma'(x_j) =$

$\sigma^{\mathrm{nc}}(x_j)$ necessarily holds true for every substitution σ' with $\sigma'(\alpha) = \sigma^{\mathrm{nc}}(\alpha)$ and for all $x_j \in \mathrm{var}(\alpha)$.

To begin with, we give the following claim:

Claim 1. For every σ' with $\sigma'(\alpha) = \sigma^{\mathrm{nc}}(\alpha)$ there does not exist any $x_j \in \mathrm{var}(\alpha)$ such that $\sigma'(x_j)$ satisfies one of the following equations:

$$\sigma'(x_j) = \mathtt{ab}^p\ u_1\ \mathtt{ab}^q\ u_2 \tag{1}$$

with $p, q \geq 0$, $p \neq q$, $u_1 = e$ or $u_1 = \mathtt{a}\, v_1$, $u_2 = e$ or $u_2 = \mathtt{a}\, v_2$, $v_1, v_2 \in \Sigma^*$, or

$$\sigma'(x_j) = u_1\ \mathtt{b}^p\mathtt{a}\ u_2\ \mathtt{b}^q\mathtt{a} \tag{2}$$

with $p, q \geq 0$, $p \neq q$, $u_1 = e$ or $u_1 = v_1\, \mathtt{a}$, $u_2 = e$ or $u_2 = v_2\, \mathtt{a}$, $v_1, v_2 \in \Sigma^*$, or

$$\sigma'(x_j) = \mathtt{b}^{\mathtt{p}}\ u_1\ \mathtt{ab}^j\mathtt{a}\ u_2\ \mathtt{b}^q \tag{3}$$

with $p, q \geq 0$, $p \neq q$, $u_1 = e$ or $u_1 = \mathtt{a}\, v_1$, $u_2 = e$ or $u_2 = v_2\, \mathtt{a}$, $v_1, v_2 \in \Sigma^*$.

Proof of Claim 1: We regard the equations (1) and (2) first: Assume to the contrary there exists a substitution σ' with $\sigma'(\alpha) = \sigma^{\mathrm{nc}}(\alpha)$ and σ' satisfying (1) or (2) for an $x_{j'} \in \mathrm{var}(\alpha)$. Since $\alpha \in \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>}$, we may conclude that $r_{j'} \geq 2$. Consequently, $\sigma'(\alpha)$ has the following shape:

$$\sigma'(\alpha) = w_1\ \mathtt{ab}^p\ w_2\ \mathtt{ab}^q\ w_3\ \mathtt{ab}^p\ w_4\ \mathtt{ab}^q\ w_5$$

or

$$\sigma'(\alpha) = w_1\ \mathtt{b}^p\mathtt{a}\ w_2\ \mathtt{b}^q\mathtt{a}\ w_3\ \mathtt{b}^p\mathtt{a}\ w_4\ \mathtt{b}^q\mathtt{a}\ w_5,$$

respectively, for some $w_1, w_2, w_3, w_4, w_5 \in \Sigma^*$ such that the substrings $\mathtt{b}^p$ and $\mathtt{b}^q$ are empty or uniform (the latter holds due to the demands on u_1, u_2 in equations (1) and (2)). Obviously, $p \neq 0$ and $q \neq 0$ since $\sigma^{\mathrm{nc}}(\alpha)$ does not contain any substring $\mathtt{aa}$. As $p \neq q$, $\sigma'(\alpha)$ contains some uniform substrings over $\mathtt{b}$ such that the sequence of their lengths is not monotonic increasing. This contradicts the shape of $\sigma^{\mathrm{nc}}(\alpha)$ described above. Thus, for every σ' with $\sigma'(\alpha) = \sigma^{\mathrm{nc}}(\alpha)$ there does not exist any $x_j \in \mathrm{var}(\alpha)$ such that $\sigma'(x_j)$ satisfies (1) or (2).

Concerning equation (3) we argue as follows: Assume to the contrary there exists a substitution σ' with $\sigma'(\alpha) = \sigma^{\mathrm{nc}}(\alpha)$ and σ' satisfying (3) for an $x_{j'} \in \mathrm{var}(\alpha)$. Then $\sigma'(\alpha)$ has the following shape:

$$\sigma'(\alpha) = w_1\ \mathtt{b}^p\ w_2\ \mathtt{ab}^{j'}\mathtt{a}\ w_3\ \mathtt{b}^q\mathtt{b}^p\ w_4\ \mathtt{ab}^{j'}\mathtt{a}\ w_5\ \mathtt{b}^q\ w_6$$

for some $w_1, w_2, w_3, w_4, w_5, w_6 \in \Sigma^*$ such that the substring $\mathtt{b}^q\mathtt{b}^p$ is empty or uniform (the latter holds due to the demands on u_1, u_2 in equation (3)). Obviously, $p+q \neq 0$ since $\sigma^{\mathrm{nc}}(\alpha)$ does not contain any substring $\mathtt{aa}$. Furthermore, if

$p+q \neq j'$ then there are some uniform substrings over $\mathtt{b}$ in $\sigma'(\alpha)$ such that the sequence of their lengths is not monotonic increasing. Thus, $p+q$ must equal j'. However, with (3), $\sigma'(\alpha)$ contains at least $2r_{j'}-1$ uniform substrings $\mathtt{b}^{j'}$, whereas there are exactly $r_{j'}$ occurrences of this uniform substring in $\sigma^{\mathrm{nc}}(\alpha)$. This contradicts the assumption. □ *Claim 1.*

We now proceed with the main part of our proof; for the respective argumentation, recall that $\alpha \in \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>}$ and therefore $\sigma(\alpha) = u_1\ \sigma(x_j)\ \sigma(x_j)\ u_2$, $u_1, u_2 \in \Sigma^*$, for every substitution σ and for every $x_j \in \mathrm{var}(\alpha)$. Assume to the contrary there exist a leftmost index j' with $\sigma'(x_{j'}) \neq \mathtt{ab}^{j'}$ for a substitution σ' with $\sigma'(\alpha) = \sigma^{\mathrm{nc}}(\alpha)$. Clearly, $\sigma'(x_{j'})$ must not begin with the letter $\mathtt{b}$ and therefore – as $\sigma^{\mathrm{nc}}(\alpha)$ does not contain any substring $\mathtt{aa}$ – it must not end with the letter $\mathtt{a}$. Furthermore,

- $\sigma'(x_{j'}) \neq (\mathtt{ab}^{j'})^{r'}\ u$ with $r' \geq 2$, $u = e$ or $u = \mathtt{a}\,v,\ v \in \Sigma^*$, and
- $\sigma'(x_{j'}) \neq \mathtt{ab}^{j''}$ with $j'' \neq j'$

as these substitutions would cause uniform substrings over $\mathtt{b}$ with the wrong number or with the wrong length in respect of $\sigma^{\mathrm{nc}}(\alpha)$. Hence, and due to the assumption and *Claim 1*, $\sigma'(x_{j'})$ must equal the empty word. Consequently, since the number of uniform substrings over $\mathtt{b}$ in $\sigma^{\mathrm{nc}}(\alpha)$ equals the length of α, there must be some variable $x_{j''} \in \mathrm{var}(\alpha)$ such that $\sigma'(x_{j''})$ contains at least two uniform substrings over $\mathtt{b}$ with different length. Moreover, we even may assume without loss of generality that one of these uniform substrings must be of length j'' as a simple combinatorial consideration reveals that the existence of $x_{j''}$ implies the existence of a variable with such a feature. Since, obviously, $\sigma'(x_{j''}) \neq \mathtt{a}\ u\ \mathtt{a}$ for any $u \in \Sigma^*$, it must satisfy equation (1), (2), or (3). This contradicts *Claim 1*. Thus, the assumption is incorrect. □

With the unambiguous words identified in Lemma 13 we can prove the main result of this section:

Theorem 14 *Let Σ be a finite alphabet, $|\Sigma| \geq 2$. Then* $\mathrm{ePAT}_{\mathrm{tf},\Sigma}^{\mathrm{nc}} \in$ LIM-TEXT.

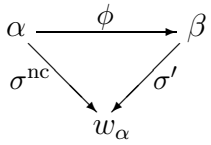
PROOF. Let α be an arbitrary pattern, $\alpha \in \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc}}$. Let the word w_α over Σ be given by $w_\alpha := \sigma^{\mathrm{nc}}(\alpha)$ with σ^{nc} derived from the proof of Lemma 13. The set T_α is defined as

$$T_\alpha := \begin{cases} \{w_\alpha\} & , \quad \alpha \in \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>}, \\ \{\mathtt{a}\} & , \quad \text{else.} \end{cases}$$

We now show that T_α is a telltale for $L(\alpha)$ in respect of $\mathrm{ePAT}_{\mathrm{tf},\Sigma}^{\mathrm{nc}}$. For $\alpha \notin \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>}$ this holds trivially. So we restrict ourselves in the following lines to

$\alpha \in \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>}$.

Assume to the contrary there exists a pattern $\beta \in \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>}$ such that $T_\alpha \subseteq L(\beta) \subset L(\alpha)$. Then – according to Fact 1 – there exists a morphism $\phi : X^* \longrightarrow X^*$ with $\phi(\alpha) = \beta$. Furthermore, there is another substitution σ' such that $\sigma'(\beta) = w_\alpha$. The following diagram illustrates the relation of α, β and w_α:



Thus, $\sigma^{\mathrm{nc}}(x_j) = \sigma'(\phi(x_j))$ for all $x_j \in \mathrm{var}(\alpha)$ since w_α is unambiguous in respect of α (cf. Lemma 13). This leads to $\sigma'(\phi(x_j)) = \mathtt{a}\,\mathtt{b}^j$ for all x_j. Consequently, $\phi(x_j) = \gamma_1\ x_{j'}\ \gamma_2$ with $\gamma_1, \gamma_2 \in X^*$ and $x_{j'} \notin \mathrm{var}(\gamma_1) \cup \mathrm{var}(\gamma_2)$, since the letter $\mathtt{a}$ is unique in $\sigma'(\phi(x_j))$ and therefore it must be generated by a variable that is unique in $\phi(x_j)$. Now we can identify two cases:

Case (i) $|\alpha|_{x_j} = |\beta|_{x_{j'}}$ for all $x_j \in \mathrm{var}(\alpha)$.
Then the morphism $\psi : X^* \longrightarrow X^*$, for all $x_k \in \mathrm{var}(\beta)$ given by

$$\psi(x_k) := \begin{cases} x_j & , \quad k = j', \\ e & , \quad \text{else}, \end{cases}$$

implies $L(\beta) \supseteq L(\alpha)$. This contradicts the assumption $L(\beta) \subset L(\alpha)$.

Case (ii) Not (i).
Thus, because of $\phi(\alpha) = \beta$, there exists an $x_j \in \mathrm{var}(\alpha)$ with $|\beta|_{x_{j'}} > |\alpha|_{x_j}$. Hence, we can assume without loss of generality that $j > 1$ as there must exist at least two variables in α that are transformed by ϕ into a string containing $x_{j'}$. Consequently, $\phi(x_j) = \gamma_1\ x_{j'}\ x_{j''}\ \gamma_2'$ with $j' \neq j''$ and $\gamma_1, \gamma_2' \in X^*$, since $\sigma'(x_{j'}) = \mathtt{a}\ v$ with $v \in \{\mathtt{b}\}^*$ (caused by $\sigma'(\phi(x_j)) = \mathtt{a}\,\mathtt{b}^j$), whereas $\sigma'(x_{j''})$ necessarily must not contain the letter $\mathtt{a}$. However, this leads to $\beta \notin \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc}}$ since $|\alpha|_{x_j} \geq 2$ for all $x_j \in \mathrm{var}(\alpha)$ (because of $\alpha \in \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>}$) and, thus, $\beta = \gamma_1\ x_{j'}\ x_{j''}\ \gamma_2\ x_{j'}\ x_{j''}\ \gamma_3$ for $\gamma_1, \gamma_2, \gamma_3 \in X^*$. This contradicts the assumption $\beta \in \mathrm{Pat}_{\mathrm{tf}}^{\mathrm{nc},>}$.

Consequently, the assumption is incorrect. Therefore T_α is a telltale for $L(\alpha)$ in respect of $\mathrm{ePAT}_{\mathrm{tf},\Sigma}^{\mathrm{nc}}$ and thus, with Fact 4, the theorem is proven. □

We consider it noteworthy that every language in $\mathrm{ePAT}_{\mathrm{tf}}^{\mathrm{nc}}$ even has a singleton telltale, as it is revealed by the proof of Theorem 14.

Finally, our result can be extended easily:

Corollary 15 $\mathrm{ePAT}^{\mathrm{nc}}_{\mathrm{tf}} \in$ LIM-TEXT.

PROOF. For an infinite or unary alphabet, not covered by Theorem 14, refer to [15]. □

5 Conclusion

Describing the results of [16], we have provided a partial answer to the long-term unresolved question on the Gold style learnability of the class of E-pattern languages: ePAT_{Σ} and even its subclass $\mathrm{ePAT}_{\mathrm{tf},\Sigma}$ are not inferrable from positive data if $|\Sigma| = 2$. Furthermore, as a positive result, we have proven the learnability of the class of terminal-free non-cross E-pattern languages for any alphabet.

We have omitted the learnability criteria on classes of terminal-free E-Pattern languages given in [16] as these criteria mostly were meant to be a substantiation of the conjecture that $\mathrm{ePAT}_{\mathrm{tf}}$ might be learnable for other than binary alphabets. Meanwhile, this assumption has been confirmed for all finite terminal alphabets with three or more letters (cf. [17]), using a different, but similar criterion.

Acknowledgements

The author is indebted to the anonymous referees, whose careful remarks helped to improve this paper significantly. Moreover, the author wishes to thank Sandra Zilles, Rolf Wiehagen and Thomas Zeugmann for their support.

References

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

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

[3] D. Angluin and C. Smith. Inductive inference: Theory and methods. *Computing Surveys*, 15:237–269, 1983.

[4] G. R. Baliga, J. Case and S. Jain. The synthesis of language learners. *Information and Computation*, 152:16–43, 1999.

[5] Ja. M. Barzdin and R. V. Freivald. On the prediction of general recursive functions. *Soviet Mathematics Doklady*, 13:1224–1228, 1972.

[6] D. R. Bean, A. Ehrenfeucht, and G. F. McNulty. Avoidable patterns in strings of symbols. *Pacific Journal of Mathematics*, 85:261–294, 1979.

[7] G. Filè. The relation of two patterns with comparable language. In *Proceedings of the 5th Annual Symposium on Theoretical Aspects of Computer Science, STACS 1988*, volume 294 of *Lecture Notes in Computer Science*, pages 184–192, 1988.

[8] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[9] T. Jiang, E. Kinber, A. Salomaa, K. Salomaa, and S. Yu. Pattern languages with and without erasing. *International Journal of Computer Mathematics*, 50:147–163, 1994.

[10] T. Jiang, A. Salomaa, K. Salomaa, and S. Yu. Decision problems for patterns. *Journal of Computer and System Sciences*, 50:53–63, 1995.

[11] V. Keränen. Abelian squares are avoidable on 4 letters. In *Proceedings of the 19th International Colloquium on Automata, Languages and Programming, ICALP 1992*, volume 623 of *Lecture Notes in Computer Science*, pages 41–52, 1992.

[12] R. Klette and R. Wiehagen. Research in the theory of inductive inference by GDR mathematicians – a survey. *Information Sciences*, 22:149–169, 1980.

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

[14] A. Mateescu and A. Salomaa. Finite degrees of ambiguity in pattern languages. *RAIRO Informatique théoretique et Applications*, 28:233–253, 1994.

[15] A. R. Mitchell. Learnability of a subclass of extended pattern languages. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998*, pages 64–71, 1998.

[16] D. Reidenbach. A negative result on inductive inference of extended pattern languages. In *Proceedings of the 13th International Conference on Algorithmic Learning Theory, ALT 2002*, volume 2533 of *Lecture Notes in Artificial Intelligence*, pages 308–320, 2002.

[17] D. Reidenbach. A discontinuity in pattern inference. In *Proceedings of the 21st Symposium on Theoretical Aspects of Computer Science, STACS 2004*, volume 2996 of *Lecture Notes in Computer Science*, pages 129–140, 2004.

[18] D. Reidenbach. On the learnability of E-pattern languages over small alphabets. In *Proceedings of the 17th Annual Conference on Learning Theory, COLT 2004*, to appear in *Lecture Notes in Artificial Intelligence*, 2004.

[19] R. Reischuk and T. Zeugmann. Learning one-variable pattern languages in linear average time. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998*, pages 198–208, 1998.

[20] H. Rogers. *Theory of Recursive Functions and Effective Computability*. MIT Press, Cambridge, Mass., 1992. 3rd print.

[21] G. Rozenberg and A. Salomaa, editors. *Handbook of Formal Languages*, volume 1. Springer, Berlin, 1997.

[22] T. Shinohara. Polynomial time inference of extended regular pattern languages. In *Proceedings of RIMS Symposia on Software Science and Engineering, Kyoto*, volume 147 of *Lecture Notes in Computer Science*, pages 115–144, 1982.

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

[24] T. Shinohara and S. Arikawa. Pattern inference. In *Algorithmic Learning for Knowledge-Based Systems, GOSLER Final Report*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 259–291, 1995.

[25] A. Thue. Über unendliche Zeichenreihen. *Kra. Vidensk. Selsk. Skrifter. I. Mat. Nat. Kl.*, 7, 1906.

[26] A. Thue. Über die gegenseitige Lage gleicher Teile gewisser Zeichenreihen. *Kra. Vidensk. Selsk. Skrifter. I. Mat. Nat. Kl.*, 1, 1912.

[27] R. Wiehagen and T. Zeugmann. Ignoring data may be the only way to learn efficiently. *Journal of Experimental and Theoretical Artificial Intelligence*, 6:131–144, 1994.

[28] K. Wright. Identification of unions of languages drawn from an identifiable class. In *Proceedings of the Second Annual Workshop on Computational Learning Theory, COLT 1989*, pages 328–333, 1989.

[29] T. Zeugmann and S. Lange. A guided tour across the boundaries of learning recursive languages. In *Algorithmic Learning for Knowledge-Based Systems, GOSLER Final Report*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 190–258, 1995.