On the learnability of E-pattern languages over small alphabets

On the Learnability of E-pattern Languages over Small Alphabets

Daniel Reidenbach*

Fachbereich Informatik, Technische Universität Kaiserslautern,
Postfach 3049, 67653 Kaiserslautern, Germany
reidenba@informatik.uni-kl.de

Abstract. This paper deals with two well discussed, but largely open problems on E-pattern languages, also known as extended or erasing pattern languages: primarily, the learnability in Gold's learning model and, secondarily, the decidability of the equivalence. As the main result, we show that the full class of E-pattern languages is not inferrable from positive data if the corresponding terminal alphabet consists of exactly three or of exactly four letters – an insight that remarkably contrasts with the recent positive finding on the learnability of the subclass of terminal-free E-pattern languages for these alphabets. As a side-effect of our reasoning thereon, we reveal some particular example patterns that disprove a conjecture of Ohlebusch and Ukkonen (*Theoretical Computer Science* 186, 1997) on the decidability of the equivalence of E-pattern languages.

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. A word of its language is generated by a uniform substitution of all variables with arbitrary strings of terminal symbols. 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 $w_1 = \mathtt{a\,a\,a\,b\,a}$, $w_2 = \mathtt{a\,b\,a\,b\,a\,b\,a\,b}$, $w_3 = \mathtt{a\,b\,b\,b}$, whereas the following examples are not covered by α: $v_1 = \mathtt{b\,a}$, $v_2 = \mathtt{b\,b\,b\,b\,b}$, $v_3 = \mathtt{b\,a\,a\,b\,a}$. Consequently, numerous regular and nonregular languages can be described by patterns in a compact and "natural" way.

The investigation of patterns in strings – initiated by Thue in [22] – may be seen as a classical topic in the research on word monoids and combinatorics of words (cf. [19]); 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 [7], [8]) – for a survey see [19] again. These examinations reveal

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

that a definition disallowing the substitution of variables with the empty word – as given by Angluin – leads to a language with particular features being quite different from the one allowing the empty substitution (that has been applied when generating w_3 in our example). Languages of the latter type have been introduced by Shinohara in [20]; contrary to those following Angluin's definition (called *NE*-pattern languages), they are referred to as *extended*, *erasing*, or simply *E*-pattern languages.

Particularly for E-pattern languages, a number of fundamental properties is still unresolved; one of the best-known open problems among these is the decidability of the *equivalence*, i.e. the question on the existence of a total computable function that, given any pair of patterns, decides whether or not they generate the same language. This problem, that for NE-pattern languages has a trivial answer in the affirmative, has been discussed several times (cf. [7], [8], [5], and [12]), contributing a number of conjectures, conditions and positive results on subclasses, but no comprehensive answer.

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. [6]) – 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 texts (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 complete description of a (potentially infinite) language from finite data. According to [6], 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, moreover, that it has been transformed into a widely analysed learning model for classes of recursive functions (cf., e.g., [4], for a survey see [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. in [9], [23], [17] and many more (for a survey see [21]) – concerning the complexity of learning algorithms, consequences of different input data, efficient strategies for subclasses, and so on. The question, however, whether or not the class of E-pattern languages is learnable – considered to be "one of the outstanding open problems in inductive infer-

ence" (cf. [11]) – remained unresolved for two decades, until it was answered in [14] in a negative way for terminal alphabets with exactly two letters. Positive results on subclasses have been presented in [20], [11], [13], and [15]. Moreover, [11] proves the full class of E-pattern languages to be learneable for infinite and unary alphabets as these alphabets significantly facilitate inferrability.

In the present paper we show that the class of E-pattern languages is not inferrable from positive data if the corresponding terminal alphabet consists of exactly three or of exactly four letters (cf. Section 3). We consider this outcome for the full class of E-pattern languages as particularly interesting as it contrasts with the results presented in [14] and [15]. The first proves the class of E-pattern languages not to be learnable for binary alphabets since even its subclass of terminal-free E-pattern languages (generated by patterns that consist of variables only) is not learnable for these alphabets. Contrary to this, the latter shows that the class of terminal-free E-pattern languages is inferrable if the corresponding terminal alphabet contains more than two letters. Consequently, with the result of the present paper in mind, for E-pattern languages there obviously is no general way to extend positive findings for the terminal-free subclass on the full class. The method we use is similar to the argumentation in [14], i.e. we give for both types of alphabets a respective example pattern with a certain property which can mislead any potential learning strategy. The foundations of this way of reasoning – that, as in [14], is solely made possible by an appropriate alphabet size and the nondeterminism of E-pattern languages – are explained in Section 2. Finally, in Section 4 one of our example patterns is shown to be applicable to the examinations on the equivalence problem by Ohlebusch and Ukkonen in [12], disproving the central conjecture given therein.

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 [18]; for unexplained aspects of formal language theory, [19] may be consulted.

$\mathbb{N}$ 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.

For any word w that contains at least one occurrence of a letter $\mathtt{a}$ we define the following subwords: $[w/\,\mathtt{a}]$ is the prefix of w up to (but not including) the leftmost occurrence of the letter $\mathtt{a}$ and $[\mathtt{a}\backslash w]$ is the suffix of w beginning with the first letter that is to the right of the leftmost occurrence of $\mathtt{a}$ in w. Thus, the specified subwords satisfy $w = [w/\,\mathtt{a}]\ \mathtt{a}\ [\mathtt{a}\backslash w]$; e.g., for $w = \mathtt{b\,c\,a\,a\,b}$, the subwords read $[w/\,\mathtt{a}] = \mathtt{b\,c}$ and $[\mathtt{a}\backslash w] = \mathtt{a\,b}$.

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 *variables*, $\Sigma \cap X = \emptyset$. Henceforth, we use lower case letters in typewriter font, e.g. $\mathtt{a}, \mathtt{b}, \mathtt{c}$, as terminal symbols exclusively; words of terminal symbols are named as u, v, or w. For every $j \geq 1$, the variable y_j is *unspecified*, i.e. there may exist indices k, k' such that $k \neq k'$, but $y_k = y_{k'}$. For unspecified terminal symbols we use upper case letters in typewriter font, such as $\mathtt{A}$.

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_Σ for the set $(\Sigma \cup X)^+$ and we use Pat instead of Pat_Σ if Σ is understood. The pattern $\chi(\alpha)$ derives from any $\alpha \in \mathrm{Pat}$ removing all terminal symbols; e.g., $\chi(x_1 x_1 \,\mathtt{a}\, x_2 \,\mathtt{b}) = x_1 x_1 x_2$.

Following [5], we designate two patterns α, β as *similar* if and only if $\alpha = \alpha_0\, u_1\, \alpha_1\, u_2\, \dots\, \alpha_{m-1}\, u_m\, \alpha_m$ and $\beta = \beta_0\, u_1\, \beta_1\, u_2\, \dots\, \beta_{m-1}\, u_m\, \beta_m$ with $m \in \mathbb{N}$, $\alpha_i, \beta_i \in X^+$ for $1 \leq i < m$, $\alpha_0, \beta_0, \alpha_m, \beta_m \in X^*$ and $u_i \in \Sigma^+$ for $i \leq m$; in other words, we call patterns similar if and only if their terminal substrings coincide.

A *substitution* is a morphism $\sigma : (\Sigma \cup X)^* \longrightarrow \Sigma^*$ such that $\sigma(\mathtt{a}) = \mathtt{a}$ for every $\mathtt{a} \in \Sigma$. 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. 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_Σ (or ePAT for short) as an abbreviation for the full class of E-pattern languages over an alphabet Σ.

Following [11], 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 $\beta = x_1 x_2 x_1 x_2$, for instance, generates the same language as the pattern $\alpha = x_1 x_1$, 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. [10]) 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 now proceed with some decidability problems on E-pattern languages: Let $\mathrm{ePAT}^\star$ be any set of E-pattern languages. We say that the *inclusion problem* for $\mathrm{ePAT}^\star$ is *decidable* if and only if there exists a computable function which, given two arbitrary patterns α, β with $L(\alpha), L(\beta) \in \mathrm{ePAT}^\star$, decides whether or not $L(\alpha) \subseteq L(\beta)$. Correspondingly, the *equivalence problem* is decidable if and only if there exists another computable function which for every pair of patterns α, β with $L(\alpha), L(\beta) \in \mathrm{ePAT}^\star$ decides whether or not $L(\alpha) = L(\beta)$. Obviously, the decidability of the inclusion implies the decidability of the equiva-

lence. The decidability of the equivalence problem for ePAT has not been resolved yet (cf. Section 4), whereas the inclusion problem is known to be undecidable (cf. [8]). Under certain circumstances, however, the inclusion problem is decidable; this is a consequence of the following fact:

Fact 1 (Ohlebusch, Ukkonen [12]). *Let Σ be an alphabet and α, β two arbitrary similar patterns such that Σ contains two distinct letters not occurring in α and β. Then $L_\Sigma(\beta) \subseteq L_\Sigma(\alpha)$ iff there exists a morphism $\phi : \mathrm{var}(\alpha)^* \longrightarrow \mathrm{var}(\beta)^*$ with $\phi(\alpha) = \beta$.*

In particular, Fact 1 implies the decidability of the inclusion problem for the class of terminal-free E-pattern languages if the alphabet contains at least two distinct letters (shown in [8]).

This paper exclusively deals with language theoretical properties of E-pattern languages. Both motivation and interpretation of our examination, however, are based on learning theory, and therefore we consider it useful to provide an adequate background. To this end, we now introduce our notions on Gold's learning model (cf. [6]) and begin with a specification of the objects to be learned. In this regard, we restrict ourselves to any *indexable class of non-empty languages*; a class $\mathcal{L}$ of languages is indexable if and only if there exists an *indexed family (of non-empty recursive languages)* $(L_i)_{i\in\mathbb{N}}$ such that $\mathcal{L} = \{L_i \mid i \in \mathbb{N}\}$ – this means that the *membership* is uniformly decidable for $(L_i)_{i\in\mathbb{N}}$, i.e. there is a total and computable function which, given any pair of an index $i \in \mathbb{N}$ and a word $w \in \Sigma^*$, decides whether or not $w \in L_i$. Concerning the learner's input, we exclusively consider inference from positive data 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. $t^n := t(0) \diamond t(1) \diamond t(2) \ldots \diamond t(n)$. Last, the learner and the learning goal need to be explained: Let the *learner* (or: the *learning strategy*) S be any total computable function 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 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.

In fact, the specific learning model given above – that largely is based on [2] – is just a special case of Gold's learning model, which frequently is considered for more general applications as well. For numerous different analyses the elements of our 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 a general *hypothesis space* instead of the indexed family).

Concerning the latter point we state that for the case when the LIM-TEXT model is applied to an indexed family, the choice of a general hypothesis spaces instead of the indexed family itself does not yield any additional learning power. For information on suchlike aspects, see [24].

Angluin has introduced some criteria on indexed families that reduce learnability to a particular language theoretical aspect (cf. [2]) and thereby facilitate our approach to learnability questions. For our purposes, the following is sufficient (combining Condition 2 and Corollary 1 of the referenced paper):

Fact 2 (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 2 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.

With regard to E-pattern languages, Fact 2 is applicable because ePAT is an indexable class of non-empty languages. This is evident as, first, a recursive enumeration of all patterns can be constructed with little effort and, second, the decidability of the membership problem for any pattern $\alpha \in$ Pat and word $w \in \Sigma^*$ is guaranteed since the search space for a successful substitution of α is bounded by the length of w.

Thus, we can conclude this section with a naming for a particular type of patterns that has been introduced in [14] and that directly aims at the content of Fact 2: A pattern β is a *passe-partout* (for a pattern α and a finite set W of words) if and only if $W \subseteq L(\beta)$ and $L(\beta) \subset L(\alpha)$. Consequently, if there exists such a passe-partout β then W is not a telltale for $L(\alpha)$.

3 The Main Result

When asking for the learnability of the class of E-pattern languages then, because of the different results on unary, binary and infinite terminal alphabets (cf. [11] and [14]), it evidently is necessary to specify the size of the alphabet. Keeping this in mind, there are some results on the learnability of subclasses that are worth to be taken into consideration, namely [20] and [15]. The first shows that the class

of *regular* E-pattern languages is learnable; these are languages generated by patterns α with $|\alpha|_{x_j} = 1$ for all $x_j \in \mathrm{var}(\alpha)$. Thus, roughly speaking, there is a way to algorithmically detect the position and the shape of the terminal symbols in the pattern from positive data. On the other hand, the latter publication shows that the class of terminal-free E-pattern languages is learnable if and only if the terminal alphabet does not consist of exactly two letters, or, in other words, that it is possible to extract the dependencies of variables for appropriate alphabets. However, our main result states that these theorems are only valid in their own context (i.e. the respective subclasses) and, consequently, that the combination of both approaches is impossible:

Theorem 1. *Let Σ be an alphabet, $|\Sigma| \in \{3, 4\}$. Then* $\mathrm{ePAT}_\Sigma \notin$ LIM-TEXT.

The proof of this theorem is given in the subsequent section.

Thus, with Theorem 1 and the results in [11] and [14], the learnability of the class of E-pattern languages is resolved for infinite alphabets and for finite alphabets with up to four letters. Concerning finite alphabets with five or more distinct letters we conjecture – as an indirect consequence of Section 3.1 – that the question of learnability for all of them can be answered in the same way:

Conjecture 1. Let Σ_1, Σ_2 be arbitrary finite alphabets with at least five letters each. Then $\mathrm{ePAT}_{\Sigma_1} \in$ LIM-TEXT iff $\mathrm{ePAT}_{\Sigma_2} \in$ LIM-TEXT.

3.1 Proof of the Main Result

First, we give an elementary lemma on morphisms, that can be formulated in several equivalent ways; however, with regard to the needs of the subsequent reasoning on Lemma 2 and Lemma 3 (that provide the actual proof of Theorem 1), we restrict ourselves to a rather special statement on mappings between terminal-free patterns. Although the fact specified therein may be considered evident we additionally give an appropriate proof sketch in order to keep this paper self-contained.

Lemma 1. *Let α, β be terminal-free patterns and ϕ, ψ morphisms with $\phi(\alpha) = \beta$ and $\psi(\beta) = \alpha$. Then either $\psi(\phi(x_j)) = x_j$ for every $x_j \in \mathrm{var}(\alpha)$ or there exists an $x_{j'} \in \mathrm{var}(\alpha)$ such that $|\psi(\phi(x_{j'}))| \geq 2$ and $x_{j'} \in \mathrm{var}(\psi(\phi(x_{j'})))$.*

We call any $x_{j'}$ satisfying these two conditions an *anchor variable* (in respect of ϕ and ψ).

Proof. Let $\alpha := y_1 y_2 y_3 \dots y_m$; then $\beta = \phi(y_1)\phi(y_2)\phi(y_3)\dots\phi(y_m)$. Let y_{k_0} be the leftmost variable such that $\psi(\phi(y_{k_0})) \neq y_{k_0}$. Now assume to the contrary there is no anchor variable in α. Then $\psi(\phi(y_{k_0}))$ necessarily equals e as otherwise $\psi(\beta) \neq \alpha$. Hence, $|\psi(\phi(y_1))\,\psi(\phi(y_2))\,\psi(\phi(y_3))\,\dots\,\psi(\phi(y_{k_0}))| = k_0 - 1$, and obviously, as there is no anchor variable in α, $|\psi(\phi(y_1))\psi(\phi(y_2))\psi(\phi(y_3))\dots\psi(\phi(y_k))| \leq k-1$ for every $k > k_0$. Consequently, $|\psi(\beta)| < |\alpha|$ and therefore $\psi(\beta) \neq \alpha$. This contradiction proves the lemma. □

We now proceed with the patterns that are crucial for our proof of Theorem 1. Contrary to the simply structured pattern used in [14] as an instrument for the negative result on binary alphabets, the examples given here unfortunately have to be rather sophisticated:

Definition 1. *The patterns* $\alpha_{\mathtt{abc}}$ *and* $\alpha_{\mathtt{abcd}}$ *are given by*

$$\alpha_{\mathtt{abc}} := x_1\ \mathtt{a}\ x_2\, x_3^2\, x_4^2\, x_5^2\, x_6^2\ \mathtt{a}\ x_7\ \mathtt{a}\ x_2\, x_8^2\, x_4^2\, x_5^2\, x_6^2\,,$$
$$\alpha_{\mathtt{abcd}} := x_1\ \mathtt{a}\ x_2\, x_3^2\, x_4^2\, x_5^2\, x_6^2\, x_7^2\, x_8\, \mathtt{b}\ x_9\ \mathtt{a}\ x_2\, x_{10}^2\, x_4^2\, x_5^2\, x_6^2\, x_{11}^2\, x_8\ \mathtt{b}\ x_{12}\,.$$

$\alpha_{\mathtt{abc}}$ is used in Lemma 2 for the proof of Theorem 1 in case of alphabets with exactly three letters and $\alpha_{\mathtt{abcd}}$ in Lemma 3 for those with four. In these lemmata we show that $L(\alpha_{\mathtt{abc}})$ and $L(\alpha_{\mathtt{abcd}})$ for their particular alphabets do not have any telltale in respect of ePAT.

First, due to the intricacy of these patterns, we consider it helpful for the understanding of the proofs of the lemmata to briefly discuss the meaning of some of their variables and terminal symbols in our reasoning; we focus on $\alpha_{\mathtt{abc}}$ since $\alpha_{\mathtt{abcd}}$ is a natural extension thereof. Our argumentation on the lemmata utilises the insight that, with regard to E-pattern languages, the ambiguity of a word decides on the question of whether this word can be a useful part of a telltale. For instance, concerning the pattern $\alpha_0 := x_4^2 x_5^2 x_6^2$, that makes up the core of our example patterns, it is shown in [14] and [15] that any telltale of $L(\alpha_0)$ necessarily has to contain particular words which consist of three distinct letters in order to avoid a specific and unwanted kind of ambiguity. However, if for any substitution σ that is applied to $\alpha_1 := x_1\,\mathtt{a}\,x_2 x_3^2\alpha_0$ – which is a prefix of $\alpha_{\mathtt{abc}}$ – $\sigma(\alpha_0)$ contains all three letters of the alphabet and, thus, includes the letter $\mathtt{a}$ then $\sigma(\alpha_1)$ again is ambiguous and always may be generated by a second substitution σ' with $\sigma'(\alpha_0) = e$, $\sigma'(x_1) = \sigma(x_1\,\mathtt{a}\,x_2 x_3^2)[\sigma(\alpha_0)/\,\mathtt{a}]$, $\sigma'(x_2) = [\mathtt{a}\setminus\sigma(\alpha_0)]$. With σ', in turn, we can give an inverse substitution leading to a tailor-made pattern that assuredly can be part of a passe-partout. Thus, for α_1 we can state the desired gap between, on the one hand, the need of substituting α_0 by three different letters and, on the other hand, the ambiguity of all words that conform to this requirement. However, due to the unique variable x_2 in α_1, the language generated by α_1 evidently equals that of $\alpha_2 := x_1\,\mathtt{a}\,x_2$, turning the core substring α_0 to be redundant. Therefore, α_1 has to occur at least twice in the pattern (with an optional separating occurrence of the letter $\mathtt{a}$). Since in the pattern $\alpha_1\,\mathtt{a}\,\alpha_1$ still both occurrences of the substring α_0 are redundant, the second occurrence of α_1 is transformed into $\alpha_1' := x_7\,\mathtt{a}\,x_2 x_8^2\alpha_0$. Hence, $\alpha_{\mathtt{abc}} = \alpha_1\,\mathtt{a}\,\alpha_1'$.

With regard to $\alpha_{\mathtt{abcd}}$, the underlying principle is similar. As stated above, three distinct letters are needed for an appropriate telltale substitution σ of α_0. However, if $\mathtt{b}, \mathtt{c}, \mathtt{d}$ are chosen as these letters, the desired ambiguity of $\sigma(\alpha_1)$ cannot be guaranteed. Hence, α_1 in $\alpha_{\mathtt{abcd}}$ is extended to $\hat{\alpha}_1 := \alpha_1 x_7^2 x_8\,\mathtt{b}\,x_9$, such that every $\sigma(\hat{\alpha}_1)$ is ambiguous as soon as $\sigma(\alpha_0)$ contains the letters $\mathtt{a}$ or $\mathtt{b}$. Furthermore, due to the reasons described above, a modification of $\hat{\alpha}_1$ serves as suffix of $\alpha_{\mathtt{abcd}}$, namely $\hat{\alpha}_1' := x_9\,\mathtt{a}\,x_2 x_{10}^2\alpha_0 x_{11}^2 x_8\,\mathtt{b}\,x_{12}$. Contrary to the structure of $\alpha_{\mathtt{abc}}$, the prefix $\hat{\alpha}_1$ and the suffix $\hat{\alpha}_1'$ in this case are not separated by a terminal symbol, but they are overlapping.

Now we specify and formalise the approach discussed above:

Lemma 2. *Let* $\Sigma := \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$. *Then for* $\alpha_{\mathtt{abc}}$ *and for every finite* $W \subset L_\Sigma(\alpha_{\mathtt{abc}})$ *there exists a passe-partout* $\beta \in \mathrm{Pat}$.

Proof. If W is empty then the claim of Lemma 2 holds trivially. Hence, let $W = \{w_1, w_2, w_3, \ldots, w_n\}$ be non-empty. Then, as $W \subset L_\Sigma(\alpha_{\mathtt{abc}})$, for every $w_i \in W$ there exists a substitution σ_i satisfying $\sigma_i(\alpha_{\mathtt{abc}}) = w_i$. Using these σ_i the following procedure constructs a passe-partout $\beta \in \mathrm{Pat}$:

Initially, we define

$$\beta_0 := \gamma_{1,0}\ \mathtt{a}\ \gamma_{2,0}\,\gamma_{3,0}^2\,\gamma_{4,0}^2\,\gamma_{5,0}^2\,\gamma_{6,0}^2\ \mathtt{a}\ \gamma_{7,0}\ \mathtt{a}\ \gamma_{2,0}\,\gamma_{8,0}^2\,\gamma_{4,0}^2\,\gamma_{5,0}^2\,\gamma_{6,0}^2$$

with $\gamma_{j,0} := e$ for every j, $1 \le j \le 8$.

For every $w_i \in W$ we define an inverse substitution $\bar\sigma_i : \Sigma^* \longrightarrow X^*$ by

$$\bar\sigma_i(\mathtt{A}) := \begin{cases} x_{3i-2} & , \quad \mathtt{A} = \mathtt{a}\,, \\ x_{3i-1} & , \quad \mathtt{A} = \mathtt{b}\,, \\ x_{3i} & , \quad \mathtt{A} = \mathtt{c}\ . \end{cases}$$

For every $i = 1, 2, 3, \ldots, n$ we now consider the following cases:

Case 1: There is no $\mathtt{A} \in \Sigma$ with $|\sigma_i(x_6)|_{\mathtt{A}} = 1$ and $|\sigma_i(\chi(\alpha_{\mathtt{abc}}))|_{\mathtt{A}} = 4$
Define $\gamma_{j,i} := \gamma_{j,i-1}\ \bar\sigma_i(\sigma_i(x_j))$ for every j, $1 \le j \le 8$.

Case 2: There is an $\mathtt{A} \in \Sigma$ with $|\sigma_i(x_6)|_{\mathtt{A}} = 1$ and $|\sigma_i(\chi(\alpha_{\mathtt{abc}}))|_{\mathtt{A}} = 4$

Case 2.1: $\mathtt{A} = \mathtt{a}$
Define $\gamma_{1,i} := \gamma_{1,i-1}\ \bar\sigma_i(\sigma_i(x_1\ \mathtt{a}\ x_2\,x_3^2\,x_4^2\,x_5^2))\ \bar\sigma_i([\sigma_i(x_6^2)/\,\mathtt{a}])\,,$
$\gamma_{2,i} := \gamma_{2,i-1}\ \bar\sigma_i([\mathtt{a}\setminus\sigma_i(x_6^2)])\,,$
$\gamma_{7,i} := \gamma_{7,i-1}\ \bar\sigma_i(\sigma_i(x_7\ \mathtt{a}\ x_2\,x_8^2\,x_4^2\,x_5^2))\ \bar\sigma_i([\sigma_i(x_6^2)/\,\mathtt{a}])\,,$
$\gamma_{j,i} := \gamma_{j,i-1},\ j \in \{3,4,5,6,8\}\,.$

Case 2.2: $\mathtt{A} = \mathtt{b}$

Case 2.2.1: $\sigma_i(x_4^2\,x_5^2) \in \{\mathtt{a}\}^* \cup \{\mathtt{c}\}^*$
Define $\gamma_{4,i} := \gamma_{4,i-1}\ \bar\sigma_i(\sigma_i(x_4\,x_5))\,,$
$\gamma_{5,i} := \gamma_{5,i-1}\ \bar\sigma_i(\sigma_i(x_6))\,,$
$\gamma_{6,i} := \gamma_{6,i-1}\,,$
$\gamma_{j,i} := \gamma_{j,i-1}\ \bar\sigma_i(\sigma_i(x_j)),\ j \in \{1,2,3,7,8\}\,.$

Case 2.2.2: $\sigma_i(x_4^2\,x_5^2) \in \{\mathtt{a},\mathtt{c}\}^+ \setminus (\{\mathtt{a}\}^+ \cup \{\mathtt{c}\}^+)$
Define $\gamma_{1,i} := \gamma_{1,i-1}\ \bar\sigma_i(\sigma_i(x_1\ \mathtt{a}\ x_2\,x_3^2))\ \bar\sigma_i([\sigma_i(x_4^2\,x_5^2)/\,\mathtt{a}])\,,$
$\gamma_{2,i} := \gamma_{2,i-1}\ \bar\sigma_i([\mathtt{a}\setminus\sigma_i(x_4^2\,x_5^2\,x_6^2)])\,,$
$\gamma_{7,i} := \gamma_{7,i-1}\ \bar\sigma_i(\sigma_i(x_7\ \mathtt{a}\ x_2\,x_8^2))\ \bar\sigma_i([\sigma_i(x_4^2\,x_5^2)/\,\mathtt{a}])\,,$
$\gamma_{j,i} := \gamma_{j,i-1},\ j \in \{3,4,5,6,8\}\,.$

Case 2.3: $\mathtt{A} = \mathtt{c}$
Adapt case 2.2 replacing $\mathtt{c}$ by $\mathtt{b}$ in the predicates of cases 2.2.1 and 2.2.2.

Finally, define

$$\beta_i := \gamma_{1,i}\ \mathtt{a}\ \gamma_{2,i}\,\gamma_{3,i}^2\,\gamma_{4,i}^2\,\gamma_{5,i}^2\,\gamma_{6,i}^2\ \mathtt{a}\ \gamma_{7,i}\ \mathtt{a}\ \gamma_{2,i}\,\gamma_{8,i}^2\,\gamma_{4,i}^2\,\gamma_{5,i}^2\,\gamma_{6,i}^2\,.$$

When this has been accomplished for every i, $1 \leq i \leq n$, then define $\beta := \beta_n$.

Now, in order to conclude the proof, the following has to be shown: β is a passe-partout for $\alpha_{\mathtt{abc}}$ and W, i.e.

1. $W \subseteq L(\beta)$ and
2. $L(\beta) \subset L(\alpha_{\mathtt{abc}})$.

ad 1. For every i, $1 \leq i \leq n$, we define a substitution σ_i' by

$$\sigma_i'(x_j) := \begin{cases} \mathtt{a} & , \quad j = 3i-2\,, \\ \mathtt{b} & , \quad j = 3i-1\,, \\ \mathtt{c} & , \quad j = 3i\,, \\ e & , \quad \text{else}\,. \end{cases}$$

If w_i satisfies case 1 then obviously $\sigma_i'(\beta) = w_i$; if w_i satisfies case 2 then w_i necessarily is ambiguous and therefore in that case $\sigma_i'(\beta) = w_i$ as well. Thus, $W \subseteq L(\beta)$.

ad 2. Obviously, $\alpha_{\mathtt{abc}}$ and β are similar and there are two letters in Σ, namely $\mathtt{b}$ and $\mathtt{c}$, that do not occur in these patterns. Consequently, the inclusion criterion given in Fact 1 is applicable. According to this, $L(\beta) \subseteq L(\alpha_{\mathtt{abc}})$ since there exists a morphism $\phi : \mathrm{var}(\alpha_{\mathtt{abc}}) \longrightarrow \mathrm{var}(\beta)^*$ with $\phi(\alpha_{\mathtt{abc}}) = \beta$, given by $\phi(x_j) = \gamma_{j,n}$ for every $x_j \in \mathrm{var}(\alpha_{\mathtt{abc}})$.

We now prove that $L(\beta)$ is a proper subset of $L(\alpha_{\mathtt{abc}})$. More precisely, we show that there is no morphism $\psi : \mathrm{var}(\beta) \longrightarrow \mathrm{var}(\alpha_{\mathtt{abc}})^*$ with $\psi(\beta) = \alpha_{\mathtt{abc}}$. For that purpose, assume to the contrary there is such a morphism ψ. Then, as there is no variable in $\mathrm{var}(\alpha_{\mathtt{abc}})$ with more than four occurrences in $\alpha_{\mathtt{abc}}$, $\psi(x_k) = e$ for all $x_k \in \mathrm{var}(\beta)$ with $|\beta|_{x_k} \geq 5$. With regard to the variables in $\mathrm{var}(\gamma_{6,n})$, this means the following: If every letter in $\sigma_i(x_6)$ occurs more than four times in $\sigma_i(\chi(\alpha_{\mathtt{abc}}))$ then case 1 is satisfied and, consequently, every variable that is added to $\gamma_{6,i}$ occurs at least five times in β. If any letter $\mathtt{A}$ in $\sigma_i(x_6)$ occurs exactly four times in $\sigma_i(\chi(\alpha_{\mathtt{abc}}))$ – and, obviously, it must be at least four times as $|\,\alpha_{\mathtt{abc}}\,|_{x_6} = 4$ – then case 2 is applied, which, enabled by the ambiguity of w_i in that case, arranges the newly added components of $\gamma_{6,i}$ such that $\bar{\sigma}_i(\sigma_i(\mathtt{A}))$ is shifted to a different $\gamma_{j,i}$. Consequently, $|\beta|_{x_k} \geq 5$ for all $x_k \in \mathrm{var}(\gamma_{6,n})$ and, therefore, $\psi(\gamma_{6,n}) = e \neq x_6$. Hence, we analyse whether or not $\mathrm{var}(\alpha_{\mathtt{abc}})$ contains an anchor variable $x_{j'}$ in respect of ϕ and ψ (cf. Lemma 1). Evidently, $j' \notin \{1, 7\}$; for $j' \in \{3, 4, 5, 8\}$, $x_{j'}$ being an anchor variable implies that $\psi(\gamma_{j',n}^2) = x_k x_{k'} \delta x_k x_{k'} \delta$ with variables $x_k, x_{k'}$ and $\delta \in X^*$, but there is no substring in $\alpha_{\mathtt{abc}}$ that equals the given shape of $\psi(\gamma_{j',n}^2)$. Finally, x_2 cannot be an anchor variable since $\psi(\gamma_{2,n})$ had to equal both $x_2 x_3 \delta$ and $x_2 x_8 \delta$ for a $\delta \in X^*$. Consequently, there is no anchor variable in $\mathrm{var}(\alpha_{\mathtt{abc}})$. This contradicts $\psi(\gamma_{6,n}) = e \neq x_6$ and therefore the assumption is incorrect. Thus, $L(\beta) \not\supseteq L(\alpha_{\mathtt{abc}})$ and, finally, $L(\beta) \subset L(\alpha_{\mathtt{abc}})$. □

Lemma 3. *Let* $\Sigma := \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$. *Then for* $\alpha_{\mathtt{abcd}}$ *and for every finite* $W \subset L_\Sigma(\alpha_{\mathtt{abcd}})$ *there exists a passe-partout* $\beta \in \mathrm{Pat}$.

Proof. We can argue similar to the proof of Lemma 2: For an empty W the claim of Lemma 2 holds obviously. For any non-empty $W = \{w_1, w_2, w_3, \ldots, w_n\} \subset L_\Sigma(\alpha_{\mathtt{abcd}})$ there exist substitutions σ_i, $1 \leq i \leq n$, satisfying $\sigma_i(\alpha_{\mathtt{abcd}}) = w_i$. With these σ_i we give the following procedure that constructs a passe-partout $\beta \in \mathrm{Pat}$:

Initially, we define

$$\beta_0 := \gamma_{1,0}\,\mathtt{a}\,\gamma_{2,0}\gamma_{3,0}^2\gamma_{4,0}^2\gamma_{5,0}^2\gamma_{6,0}^2\gamma_{7,0}^2\gamma_{8,0}\,\mathtt{b}\,\gamma_{9,0}\,\mathtt{a}\,\gamma_{2,0}\gamma_{10,0}^2\gamma_{4,0}^2\gamma_{5,0}^2\gamma_{6,0}^2\gamma_{11,0}^2\gamma_{8,0}\,\mathtt{b}\,\gamma_{12,0}$$

with $\gamma_{j,0} := e$ for every j, $1 \leq j \leq 12$.

For every $w_i \in W$ we define an inverse substitution $\bar\sigma_i : \Sigma^* \longrightarrow X^*$ by

$$\bar\sigma_i(\mathtt{A}) := \begin{cases} x_{4i-3} & , \quad \mathtt{A} = \mathtt{a}\,, \\ x_{4i-2} & , \quad \mathtt{A} = \mathtt{b}\,, \\ x_{4i-1} & , \quad \mathtt{A} = \mathtt{c}\,, \\ x_{4i} & , \quad \mathtt{A} = \mathtt{d}\,. \end{cases}$$

For every $i = 1, 2, 3, \ldots, n$ we now consider the following cases:

Case 1: There is no $\mathtt{A} \in \Sigma$ with $|\sigma_i(x_6)|_{\mathtt{A}} = 1$ and $|\sigma_i(\chi(\alpha_{\mathtt{abcd}}))|_{\mathtt{A}} = 4$
Define $\gamma_{j,i} := \gamma_{j,i-1}\ \bar\sigma_i(\sigma_i(x_j))$ for every j, $1 \leq j \leq 12$.

Case 2: There is an $\mathtt{A} \in \Sigma$ with $|\sigma_i(x_6)|_{\mathtt{A}} = 1$ and $|\sigma_i(\chi(\alpha_{\mathtt{abcd}}))|_{\mathtt{A}} = 4$

Case 2.1: $\mathtt{A} = \mathtt{a}$
Define
$$\begin{aligned}
\gamma_{1,i} &:= \gamma_{1,i-1}\ \bar\sigma_i(\sigma_i(x_1\ \mathtt{a}\ x_2\, x_3^2\, x_4^2\, x_5^2))\ \bar\sigma_i([\sigma_i(x_6^2)/\,\mathtt{a}])\,, \\
\gamma_{2,i} &:= \gamma_{2,i-1}\ \bar\sigma_i([\mathtt{a}\setminus\sigma_i(x_6^2)])\,, \\
\gamma_{9,i} &:= \gamma_{9,i-1}\ \bar\sigma_i(\sigma_i(x_9\ \mathtt{a}\ x_2\, x_{10}^2\, x_4^2\, x_5^2))\ \bar\sigma_i([\sigma_i(x_6^2)/\,\mathtt{a}])\,, \\
\gamma_{j,i} &:= \gamma_{j,i-1},\ j \in \{3,4,5,6,10\}\,, \\
\gamma_{j,i} &:= \gamma_{j,i-1}\ \bar\sigma_i(\sigma_i(x_j)),\ j \in \{7,8,11,12\}\,.
\end{aligned}$$

Case 2.2: $\mathtt{A} = \mathtt{b}$
Define
$$\begin{aligned}
\gamma_{8,i} &:= \gamma_{8,i-1}\ \bar\sigma(\sigma_i(x_4^2\, x_5^2))\ \bar\sigma_i([\sigma_i(x_6^2)/\,\mathtt{b}])\,, \\
\gamma_{9,i} &:= \gamma_{9,i-1}\ \bar\sigma_i([\mathtt{b}\setminus\sigma_i(x_6^2\, x_7^2\, x_8\ \mathtt{b}\ x_9)])\,, \\
\gamma_{12,i} &:= \gamma_{12,i-1}\ \bar\sigma_i([\mathtt{b}\setminus\sigma_i(x_6^2\, x_{11}^2\, x_8\ \mathtt{b}\ x_{12})])\,, \\
\gamma_{j,i} &:= \gamma_{j,i-1},\ j \in \{4,5,6,7,11\}\,, \\
\gamma_{j,i} &:= \gamma_{j,i-1}\ \bar\sigma_i(\sigma_i(x_j)),\ j \in \{1,2,3,10\}\,.
\end{aligned}$$

Case 2.3: $\mathtt{A} = \mathtt{c}$

Case 2.3.1: $\sigma_i(x_4^2\, x_5^2) \in \{\mathtt{a}\}^* \cup \{\mathtt{b}\}^* \cup \{\mathtt{d}\}^*$
Define
$$\begin{aligned}
\gamma_{4,i} &:= \gamma_{4,i-1}\ \bar\sigma_i(\sigma_i(x_4\, x_5))\,, \\
\gamma_{5,i} &:= \gamma_{5,i-1}\ \bar\sigma_i(\sigma_i(x_6))\,, \\
\gamma_{6,i} &:= \gamma_{6,i-1}\,, \\
\gamma_{j,i} &:= \gamma_{j,i-1}\ \bar\sigma_i(\sigma_i(x_j)),\ j \in \{1,2,3,7,8,9,10,11,12\}\,.
\end{aligned}$$

Case 2.3.2: $\sigma_i(x_4^2\, x_5^2) \in \{\mathtt{a},\mathtt{d}\}^+ \setminus (\{\mathtt{a}\}^+ \cup \{\mathtt{d}\}^+)$

Define $\gamma_{1,i} := \gamma_{1,i-1}\ \bar{\sigma}_i(\sigma_i(x_1\ \mathtt{a}\ x_2\, x_3^2))\ \bar{\sigma}_i([\sigma_i(x_4^2\, x_5^2)/\,\mathtt{a}])\,,$

$\gamma_{2,i} := \gamma_{2,i-1}\ \bar{\sigma}_i([\mathtt{a}\setminus\sigma_i(x_4^2\, x_5^2\, x_6^2)])\,,$

$\gamma_{9,i} := \gamma_{9,i-1}\ \bar{\sigma}_i(\sigma_i(x_9\ \mathtt{a}\ x_2\, x_{10}^2))\ \bar{\sigma}_i([\sigma_i(x_4^2\, x_5^2)/\,\mathtt{a}])\,,$

$\gamma_{j,i} := \gamma_{j,i-1},\ j \in \{3,4,5,6,10\}\,,$

$\gamma_{j,i} := \gamma_{j,i-1}\ \bar{\sigma}_i(\sigma_i(x_j)),\ j \in \{7,8,11,12\}\,.$

Case 2.3.3: $\sigma_i(x_4^2\, x_5^2) \in \{\mathtt{a},\mathtt{b},\mathtt{d}\}^+ \setminus (\{\mathtt{a}\}^+ \cup \{\mathtt{b}\}^+ \cup \{\mathtt{d}\}^+ \cup \{\mathtt{a},\mathtt{d}\}^+)$

Define $\gamma_{8,i} := \gamma_{8,i-1}\ \bar{\sigma}([\sigma_i(x_4^2\, x_5^2)/\,\mathtt{b}])\,,$

$\gamma_{9,i} := \gamma_{9,i-1}\ \bar{\sigma}_i([\mathtt{b}\setminus\sigma_i(x_4^2\, x_5^2\, x_6^2\, x_7^2\, x_8\ \mathtt{b}\ x_9)])\,,$

$\gamma_{12,i} := \gamma_{12,i-1}\ \bar{\sigma}_i([\mathtt{b}\setminus\sigma_i(x_4^2\, x_5^2\, x_6^2\, x_{11}^2\, x_8\ \mathtt{b}\ x_{12})])\,,$

$\gamma_{j,i} := \gamma_{j,i-1},\ j \in \{4,5,6,7,11\}\,,$

$\gamma_{j,i} := \gamma_{j,i-1}\ \bar{\sigma}_i(\sigma_i(x_j)),\ j \in \{1,2,3,10\}\,.$

Case 2.4: $\mathtt{A} = \mathtt{d}$

Adapt case 2.3 replacing $\mathtt{d}$ by $\mathtt{c}$ in the predicates of cases 2.3.1, 2.3.2 and 2.3.3.

Finally, define

$$\beta_i := \gamma_{1,i}\,\mathtt{a}\,\gamma_{2,i}\gamma_{3,i}^2\gamma_{4,i}^2\gamma_{5,i}^2\gamma_{6,i}^2\gamma_{7,i}^2\gamma_{8,i}\,\mathtt{b}\,\gamma_{9,i}\,\mathtt{a}\,\gamma_{2,i}\gamma_{10,i}^2\gamma_{4,i}^2\gamma_{5,i}^2\gamma_{6,i}^2\gamma_{11,i}^2\gamma_{8,i}\,\mathtt{b}\,\gamma_{12,i}\,.$$

When this has been accomplished for every i, $1 \le i \le n$, then define $\beta := \beta_n$.

For the proof that β indeed is a passe-partout for $\alpha_{\mathtt{abcd}}$ and W, see the proof of Lemma 2, mutatis mutandis. □

Concluding the proof of Theorem 1, we state that it directly follows from Lemma 2, Lemma 3, and Fact 2: Obviously, any indexed family $(L_i)_{i\in\mathbb{N}}$ with $\{L_i \mid i \in \mathbb{N}\} = \text{ePAT}$ necessarily contains all languages generated by potential passe-partouts for $\alpha_{\mathtt{abc}}$ and $\alpha_{\mathtt{abcd}}$, respectively. Thus, $L_\Sigma(\alpha_{\mathtt{abc}})$ has no telltale in respect of ePAT_Σ if $|\Sigma| = 3$ and $L_\Sigma(\alpha_{\mathtt{abcd}})$ has no telltale in respect of ePAT_Σ if $|\Sigma| = 4$. Consequently, ePAT_Σ is not learnable for these two types of alphabets.

3.2 Some Remarks

Clearly, both procedures constructing the passe-partouts implement only one out of many possibilities. The definition of the $\gamma_{j,i}$ in case 2.3.1 in the proof of Lemma 3, for instance, could be separated in cases 2.3.1.1 and 2.3.1.2 depending on the question whether or not $\sigma_i(x_4^2x_5^2) \in \{\mathtt{a}\}^+$. If so then case 2.3.1.1 could equal case 2.3.2, possibly leading to a different passe-partout. It can be seen easily that there are numerous other options like this. On the other hand, there are infinitely many different succinct patterns that can act as a substitute for $\alpha_{\mathtt{abc}}$ and $\alpha_{\mathtt{abcd}}$ in the respective lemmata. Some of these patterns, for instance, can be constructed replacing in $\alpha_{\mathtt{abc}}$ and $\alpha_{\mathtt{abcd}}$ the substring $\alpha_0 = x_4^2x_5^2x_6^2$ by any $\alpha_0' = x_p^2x_{p+1}^2\ldots x_{p+q}^2$, $p > \max\{j \mid x_j \in \mathrm{var}(\alpha_{\mathtt{abcd}})\}$, $q \ge 4$. Hence, the phenomenon described in Lemma 2 and Lemma 3 is ubiquitous in ePAT. Therefore

we give some brief considerations concerning the question on the shortest patterns generating a language without telltale in respect of ePAT. Obviously, even for the proof concept of Lemma 2 and Lemma 3, shorter patterns are suitable. In $\alpha_{\mathtt{abc}}$, e.g., the substring x_3^2 and the separating terminal symbol $\mathtt{a}$ in the middle of the pattern can be removed without loss of applicability; for $\alpha_{\mathtt{abcd}}$, e.g., the substrings x_3^2 and x_7^2 can be mentioned. Nevertheless, we consider both patterns in the given shape easier to grasp, and, moreover, we assume that the indicated steps for shortening $\alpha_{\mathtt{abc}}$ and $\alpha_{\mathtt{abcd}}$ lead to patterns with minimum length:

Conjecture 2. Let the alphabets Σ_1 and Σ_2 be given by $\Sigma_1 := \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and $\Sigma_2 := \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$. Let the patterns $\alpha_{\mathtt{abc}}{}'$ and $\alpha_{\mathtt{abcd}}{}'$ be given by

$$\alpha_{\mathtt{abc}}{}' := x_1\ \mathtt{a}\ x_2\, x_4^2\, x_5^2\, x_6^2\, x_7\ \mathtt{a}\ x_2\, x_8^2\, x_4^2\, x_5^2\, x_6^2,$$
$$\alpha_{\mathtt{abcd}}{}' := x_1\ \mathtt{a}\ x_2\, x_4^2\, x_5^2\, x_6^2\, x_8\, \mathtt{b}\ x_9\ \mathtt{a}\ x_2\, x_{10}^2\, x_4^2\, x_5^2\, x_6^2\, x_{11}^2\, x_8\ \mathtt{b}\ x_{12}.$$

Then $L_{\Sigma_1}(\alpha_{\mathtt{abc}}{}')$ has no telltale in respect of ePAT_{Σ_1}, $L_{\Sigma_2}(\alpha_{\mathtt{abcd}}{}')$ has no telltale in respect of ePAT_{Σ_2} and there do not exist any shorter patterns in Pat with this respective property.

Finally, we emphasise that we consider it necessary to prove our result for both alphabet types separately. Obviously, for our way of reasoning, this is caused by the fact that the proof of Lemma 2 cannot be conducted with $\alpha_{\mathtt{abcd}}$ since this pattern – in combination with any passe-partout an adapted procedure could generate – does not satisfy the conditions of Fact 1 for alphabets with three letters. In effect, the problem is even more fundamental: Assume there are two alphabets Σ_1 and Σ_2 with $\Sigma_1 \subset \Sigma_2$. If for some $\alpha \in \mathrm{Pat}_{\Sigma_1}$ there is no telltale $T_\alpha \subseteq L_{\Sigma_2}(\alpha)$ – as shown to be true for $\alpha_{\mathtt{abcd}}$ – then, at first glance, it seems natural to expect the same for $L_{\Sigma_1}(\alpha)$ since $L_{\Sigma_1}(\alpha) \subset L_{\Sigma_2}(\alpha)$. These considerations, however, immediately are disproven, for instance, by the fact that ePAT is learnable for unary, but not for binary alphabets (cf. [11] and [14]). This can be illustrated easily, e.g., by $\alpha_{\mathtt{abc}}$ and the pattern $\alpha = \mathtt{a}\,\mathtt{a}\,x_1\,\mathtt{a}$. With $\Sigma_1 = \{\mathtt{a}\}$ and $\Sigma_2 = \{\mathtt{a}, \mathtt{b}\}$ we may state $L_{\Sigma_1}(\alpha) = L_{\Sigma_1}(\alpha_{\mathtt{abc}})$, but $L_{\Sigma_2}(\alpha) \subset L_{\Sigma_2}(\alpha_{\mathtt{abc}})$. Thus, for Σ_1 both patterns generate the same language and, consequently, they have the same telltale, whereas any telltale for $L_{\Sigma_2}(\alpha_{\mathtt{abc}})$ has to contain a word that is not in $L_{\Sigma_2}(\alpha)$. The changing equivalence of E-pattern languages is a well-known fact for pairs of alphabets if the smaller one contains at most two distinct letters, but, concerning those pairs with three or more letters each, [12] conjectures that the situation stabilises. This is examined in the following section.

4 $\alpha_{\mathtt{abcd}}$ and the Equivalence Problem

The equivalence problem for E-pattern languages – one of the most prominent and well discussed open problems on this subject – has first been examined in [7] and later in [8], [5], and [12]. The latter authors conjecture that, for patterns $\alpha, \beta \in \mathrm{Pat}$ and any alphabet Σ, $|\Sigma| \geq 3$, $L_\Sigma(\alpha) = L_\Sigma(\beta)$ if and only if there are morphisms $\phi : \mathrm{var}(\alpha) \longrightarrow \mathrm{var}(\beta)$ and $\psi : \mathrm{var}(\beta) \longrightarrow \mathrm{var}(\alpha)$ such that $\phi(\alpha) = \beta$

and $\psi(\beta) = \alpha$ (cf. [12], paraphrase of Conjecture 1). Furthermore, derived from Fact 1 and Theorem 5.3 of [7], the authors state that the equivalence problem is decidable if the following question (cf. [12], Open Question 2) has a positive answer: For arbitrary alphabets Σ_1, Σ_2 with $|\Sigma_1| \geq 3$ and $\Sigma_2 = \Sigma_1 \cup \{\mathtt{d}\}$, $\mathtt{d} \notin \Sigma_1$, and patterns $\alpha, \beta \in \mathrm{Pat}_{\Sigma_1}$, does the following statement hold: $L_{\Sigma_1}(\alpha) = L_{\Sigma_1}(\beta)$ iff $L_{\Sigma_2}(\alpha) = L_{\Sigma_2}(\beta)$? In other words: Is the equivalence of E-pattern languages *preserved under alphabet extension*?

We now show that for $|\Sigma_1| = 3$ this question has an answer in the negative, using $\alpha_{\mathtt{abcd}}$ – which for the learnability result in Section 3 is applied to $|\Sigma| = 4$ – and the following pattern: $\alpha_\sim := x_1\ \mathtt{a}\ x_2\, x_3^2\, x_4^2\, x_7^2\ x_8\ \mathtt{b}\ x_9\ \mathtt{a}\ x_2\, x_{10}^2\, x_4^2\, x_{11}^2\, x_8\ \mathtt{b}\ x_{12}$.

Theorem 2. *Let the alphabets Σ_1 and Σ_2 be given by $\Sigma_1 := \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ and $\Sigma_2 \supseteq \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$. Then $L_{\Sigma_1}(\alpha_{\mathtt{abcd}}) = L_{\Sigma_1}(\alpha_\sim)$, but $L_{\Sigma_2}(\alpha_{\mathtt{abcd}}) \neq L_{\Sigma_2}(\alpha_\sim)$.*

Proof. We first show that $L_{\Sigma_1}(\alpha_{\mathtt{abcd}}) = L_{\Sigma_1}(\alpha_\sim)$. Let $\sigma : (\Sigma_1 \cup X)^* \longrightarrow \Sigma_1$ be any substitution that is applied to $\alpha_\sim$. Then, obviously, the substitution σ' with $\sigma'(x_j) = \sigma(x_j)$ for all $x_j \in \mathrm{var}(\alpha_\sim)$ and $\sigma'(x_j) = e$ for all $x_j \notin \mathrm{var}(\alpha_\sim)$ leads to $\sigma'(\alpha_{\mathtt{abcd}}) = \sigma(\alpha_\sim)$ and, thus, $L_{\Sigma_1}(\alpha_\sim) \subseteq L_{\Sigma_1}(\alpha_{\mathtt{abcd}})$.

Now, let σ be any substitution that is applied to $\alpha_{\mathtt{abcd}}$. We give a second substitution σ' that leads to $\sigma'(\alpha_\sim) = \sigma(\alpha_{\mathtt{abcd}})$ and, thus, $L_{\Sigma_1}(\alpha_\sim) = L_{\Sigma_1}(\alpha_{\mathtt{abcd}})$:

Case 1: $\sigma(x_4^2\, x_5^2\, x_6^2) \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^+ \setminus \{\mathtt{b}, \mathtt{c}\}^+$

Define $\sigma'(x_1) := \sigma(x_1\ \mathtt{a}\ x_2\, x_3^2)\ [\sigma(x_4^2\, x_5^2\, x_6^2)/\,\mathtt{a}]$,
$\sigma'(x_2) := [\mathtt{a}\setminus\sigma(x_4^2\, x_5^2\, x_6^2)]$,
$\sigma'(x_9) := \sigma(x_9\ \mathtt{a}\ x_2\, x_{10}^2)\ [\sigma(x_4^2\, x_5^2\, x_6^2)/\,\mathtt{a}]$,
$\sigma'(x_j) := \sigma(x_j),\ j \in \{7, 8, 11, 12\}$,
$\sigma'(x_j) := e,\ j \in \{3, 4, 10\}$.

Case 2: $\sigma(x_4^2\, x_5^2\, x_6^2) \in \{\mathtt{b}, \mathtt{c}\}^+ \setminus \{\mathtt{c}\}^+$

Define σ' symmetrically to case 1 using x_9 for x_1, x_8 for x_2, and x_{12} for x_9 (cf., e.g., case 2.2 in the proof of Lemma 3).

Case 3: $\sigma(x_4^2\, x_5^2\, x_6^2) \in \{\mathtt{c}\}^*$

Define $\sigma'(x_4) = \sigma(x_4\, x_5\, x_6)$ and $\sigma'(x_j) = \sigma(x_j)$ for $x_j \in \mathrm{var}(\alpha_\sim)$, $j \neq 4$.

The proof for $L_{\Sigma_2}(\alpha_\sim) \neq L_{\Sigma_2}(\alpha_{\mathtt{abcd}})$ uses Fact 1 and Lemma 1 and is similar to the argumentation on $L(\beta) \subset L(\alpha_{\mathtt{abc}})$ in the proof of Lemma 2. □

Moreover, the reasoning on Theorem 2 reveals that Conjecture 1 in [12] – as cited above – is incorrect:

Corollary 1. *Let Σ be an alphabet, $|\Sigma| = 3$. Then $L_\Sigma(\alpha_{\mathtt{abcd}}) = L_\Sigma(\alpha_\sim)$ and there exists a morphism $\phi : \mathrm{var}(\alpha_{\mathtt{abcd}}) \longrightarrow \mathrm{var}(\alpha_\sim)$ with $\phi(\alpha_{\mathtt{abcd}}) = \alpha_\sim$, but there does not exist any morphism $\psi : \mathrm{var}(\alpha_\sim) \longrightarrow \mathrm{var}(\alpha_{\mathtt{abcd}})$ with $\psi(\alpha_\sim) = \alpha_{\mathtt{abcd}}$.*

Note that the argumentation on Theorem 2 and Corollary 1 can be conducted with a pattern that is shorter than $\alpha_{\mathtt{abcd}}$ (e.g., by removing x_6).

In [16], that solely examines the above questions for the transition between alphabets with four and alphabets with five letters, some methods of the present section are adopted and, thus, they are explained in more detail.

References

[1] D. Angluin. Finding patterns common to a set of strings. *J. Comput. Syst. Sci.*, 21:46–62, 1980.

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

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

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

[5] G. Dány and Z. Fülöp. A note on the equivalence problem of E-patterns. *Inf. Process. Lett.*, 57:125–128, 1996.

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

[7] T. Jiang, E. Kinber, A. Salomaa, K. Salomaa, and S. Yu. Pattern languages with and without erasing. *Int. J. Comput. Math.*, 50:147–163, 1994.

[8] T. Jiang, A. Salomaa, K. Salomaa, and S. Yu. Decision problems for patterns. *J. Comput. Syst. Sci.*, 50:53–63, 1995.

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

[10] A. Mateescu and A. Salomaa. Finite degrees of ambiguity in pattern languages. *RAIRO Inform. théor.*, 28(3–4):233–253, 1994.

[11] A.R. Mitchell. Learnability of a subclass of extended pattern languages. In *Proc. COLT 1998*, pages 64–71, 1998.

[12] E. Ohlebusch and E. Ukkonen. On the equivalence problem for E-pattern languages. *Theor. Comp. Sci.*, 186:231–248, 1997.

[13] D. Reidenbach. A non-learnable class of E-pattern languages. *Theor. Comp. Sci.*, to appear.

[14] D. Reidenbach. A negative result on inductive inference of extended pattern languages. In *Proc. ALT 2002*, volume 2533 of *LNAI*, pages 308–320, 2002.

[15] D. Reidenbach. A discontinuity in pattern inference. In *Proc. STACS 2004*, volume 2996 of *LNCS*, pages 129–140, 2004.

[16] D. Reidenbach. On the equivalence problem for E-pattern languages over four letters. In *Proc. MFCS 2004*, *LNCS*, 2004. Submitted.

[17] R. Reischuk and T. Zeugmann. Learning one-variable pattern languages in linear average time. In *Proc. COLT 1998*, pages 198–208, 1998.

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

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

[20] T. Shinohara. Polynomial time inference of extended regular pattern languages. In *Proc. RIMS Symp.*, volume 147 of *LNCS*, pages 115–127, 1982.

[21] T. Shinohara and S. Arikawa. Pattern inference. In *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *LNAI*, pages 259–291. Springer, 1995.

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

[23] R. Wiehagen and T. Zeugmann. Ignoring data may be the only way to learn efficiently. *J. Exp. Theor. Artif. Intell.*, 6:131–144, 1994.

[24] T. Zeugmann and S. Lange. A guided tour across the boundaries of learning recursive languages. In *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *LNAI*, pages 190–258. Springer, 1995.