Regular and Context-Free Pattern Languages Over Small Alphabets

Daniel Reidenbach and Markus L. Schmid *

Department of Computer Science, Loughborough University,
Loughborough, Leicestershire, LE11 3TU, United Kingdom
{D.Reidenbach,M.Schmid}@lboro.ac.uk

Abstract. Pattern languages are generalisations of the copy language, which is a standard textbook example of a context-sensitive and non-context-free language. In this work, we investigate a counter-intuitive phenomenon: with respect to alphabets of size 2 and 3, pattern languages can be regular or context-free in an unexpected way. For this regularity and context-freeness of pattern languages, we give several sufficient and necessary conditions and improve known results.

Keywords: Pattern Languages, Regular Languages, Context-Free Languages

1 Introduction

Within the scope of this paper, a *pattern* is a finite sequence of terminal symbols and variables, taken from two disjoint alphabets Σ and X. We say that such a pattern α generates a word w if w can be obtained from α by substituting arbitrary words of terminal symbols for all variables in α, where, for any variable, the substitution word must be identical for all of its occurrences in α. More formally, a substitution is therefore a *terminal-preserving* morphism, i. e., a morphism $\sigma : (\Sigma \cup X)^* \to \Sigma^*$ that satisfies $\sigma(a) = a$ for every $a \in \Sigma$. The *pattern language* $L(\alpha)$ is then simply the set of all words that can be obtained from α by arbitrary substitutions. For example, the language generated by $\alpha_1 := x_1 x_1 \mathtt{aba} x_2$ (where $\Sigma := \{\mathtt{a}, \mathtt{b}\}$ and $X \supset \{x_1, x_2\}$) is the set of all words over $\{\mathtt{a}, \mathtt{b}\}$ that have any square as a prefix, an arbitrary suffix and the factor $\mathtt{aba}$ in between. Hence, e. g., $w_1 := \mathtt{abbabbabaaa}$ and $w_2 := \mathtt{bbaba}$ are included in $L(\alpha_1)$, whereas $w_3 := \mathtt{abbababb}$ and $w_4 := \mathtt{bbbabaaa}$ are not.

Pattern languages were introduced by Angluin [1] in 1980 in order to formalise the process of computing commonalities of words in some given set. Her original definition disallows the substitution of the empty word for the variables, and therefore these languages are also referred to as *nonerasing* pattern languages (or *NE*-pattern languages for short). This notion of pattern languages was soon afterwards complemented by Shinohara [16], who included the empty word as an admissible substitution word, leading to the definition of *extended* or

* Corresponding author.

erasing pattern languages (or *E*-pattern languages for short). Thus, in the above example, w_2 is contained in the E-pattern language, but not in the NE-pattern language of α_1. As revealed by numerous studies, the small difference between the definitions of NE- and E-pattern languages entails substantial differences between some of the properties of the resulting (classes of) formal languages (see, e. g., Mateescu and Salomaa [11] for a survey).

Pattern languages have not only been intensively studied within the scope of inductive inference (see, e. g., Lange and Wiehagen [9], Rossmanith and Zeugmann [15], Reidenbach [14] and, for a survey, Ng and Shinohara [12]), but their properties are closely connected to a variety of fundamental problems in computer science and discrete mathematics, such as for (un-)avoidable patterns (cf. Jiang et al. [8]), word equations (cf. Mateescu and Salomaa [10]), the ambiguity of morphisms (cf. Freydenberger et al. [5]), equality sets (cf. Harju and Karhumäki [6]) and extended regular expressions (cf. Câmpeanu et al. [3]). Therefore, quite a number of basic questions for pattern languages are still open or have been resolved just recently (see, e. g., Freydenberger and Reidenbach [4]).

If a pattern contains each of its variables once, then this pattern can be interpreted as a regular expression, and therefore its language is regular. In contrast to this, if a pattern has at least one variable with multiple occurrences, then its languages is a variant of the well known *copy language* $\{xx \mid x \in \Sigma^*\}$, which for $|\Sigma| \geq 2$ is a standard textbook example of a context-sensitive and non-context-free language. Nevertheless, there are some well-known example patterns of the latter type that generate regular languages. For instance, the NE-pattern language of $\alpha_2 := x_1x_2x_2x_3$ is regular for $|\Sigma| = 2$, since squares are unavoidable for binary alphabets, which means that the language is co-finite. Surprisingly, for terminal alphabets of size 2 and 3, there are even certain E- and NE-pattern languages that are context-free but not regular. This recent insight is due to Jain et al. [7] and solves a longstanding open problem.

It is the purpose of our paper to further investigate this counter-intuitive existence of languages that appear to be variants of the copy language, but are nevertheless regular or context-free. Thus, we wish to establish criteria where the seemingly high complexity of a pattern does not translate into a high complexity of its language. Since, as demonstrated by Jain et al., this phenomenon does not occur for E-pattern languages if the pattern does not contain any terminal symbols or if the size of the terminal alphabet is at least 4, our investigations focus on patterns with terminal symbols and on small alphabets of sizes 2 or 3.

Note that, due to space constraints, all proofs are omitted from this paper.

2 Definitions and Known Results

Let $\mathbb{N} := \{1, 2, 3, \ldots\}$ and let $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. For an arbitrary alphabet A, a *string* (*over* A) is a finite sequence of symbols from A, and ε stands for the *empty string*. The notation A^+ denotes the set of all nonempty strings over A, and $A^* := A^+ \cup \{\varepsilon\}$. For the *concatenation* of two strings w_1, w_2 we write $w_1 \cdot w_2$ or simply w_1w_2. We say that a string $v \in A^*$ is a *factor* of a string $w \in A^*$ if there are $u_1, u_2 \in A^*$ such that $w = u_1 \cdot v \cdot u_2$. If u_1 or u_2 is the empty string, then v is a *prefix* (or a *suffix*, respectively) of w. The notation $|K|$ stands for the size of a set K or the length of a string K.

If we wish to refer to the symbol at a certain position j, $1 \leq j \leq n$, in a string $w = \mathtt{a}_1 \cdot \mathtt{a}_2 \cdot \cdots \cdot \mathtt{a}_n$, $\mathtt{a}_i \in A$, $1 \leq i \leq n$, then we use $w[j] := \mathtt{a}_j$ and if the length of a string is unknown, then we denote its last symbol by $w[-] := w[|w|]$. Furthermore, for each j, j', $1 \leq j < j' \leq |w|$, let $w[j, j'] := \mathtt{a}_j \cdot \mathtt{a}_{j+1} \cdot \cdots \cdot \mathtt{a}_{j'}$ and $w[j, -] := w[j, |w|]$.

For any alphabets A, B, a *morphism* is a function $h : A^* \to B^*$ that satisfies $h(vw) = h(v)h(w)$ for all $v, w \in A^*$; h is said to be *nonerasing* if and only if, for every $a \in A$, $h(a) \neq \varepsilon$. Let Σ be a finite alphabet of so-called *terminal symbols* and X a countably infinite set of *variables* with $\Sigma \cap X = \emptyset$. We normally assume $X := \{x_1, x_2, x_3, \ldots\}$. A *pattern* is a nonempty string over $\Sigma \cup X$, a *terminal-free pattern* is a nonempty string over X and a *word* is a string over Σ. For any pattern α, we refer to the set of variables in α as $\mathrm{var}(\alpha)$ and for any $x \in \mathrm{var}(\alpha)$, $|\alpha|_x$ denotes the number of occurrences of x in α. A morphism $h : (\Sigma \cup X)^* \to \Sigma^*$ is called a *substitution* if $h(a) = a$ for every $a \in \Sigma$.

Definition 1. *Let $\alpha \in (\Sigma \cup X)^*$ be a pattern. The* E-pattern language *of α is defined by $L_{\mathrm{E},\Sigma}(\alpha) := \{h(\alpha) \mid h : (\Sigma \cup X)^* \to \Sigma^*$ is a substitution$\}$. The* NE-pattern language *of α is defined by $L_{\mathrm{NE},\Sigma}(\alpha) := \{h(\alpha) \mid h : (\Sigma \cup X)^* \to \Sigma^*$ is a nonerasing substitution$\}$.*

We denote the class of *regular* languages, *context-free* languages, E-*pattern* languages over Σ and NE-*pattern* languages over Σ by REG, CF, E-PAT$_\Sigma$ and NE-PAT$_\Sigma$, respectively. We use regular expressions as they are commonly defined (see, e. g., Yu [18]) and for any regular expression r, $L(r)$ denotes the language described by r.

We recapitulate regular and block-regular patterns as defined by Shinohara [17] and Jain et al. [7]. A pattern α is a *regular* pattern if, for every $x \in \mathrm{var}(\alpha)$, $|\alpha|_x = 1$. Every factor of variables of α that is delimited by terminal symbols is called a variable block. More precisely, for every i, j, $1 \leq i \leq j \leq |\alpha|$, $\alpha[i, j]$ is a *variable block* if and only if $\alpha[k] \in X$, $i \leq k \leq j$, $\alpha[i-1] \in \Sigma$ or $i = 1$ and $\alpha[j+1] \in \Sigma$ or $j = |\alpha|$. A pattern α is *block-regular* if in every variable block of α there occurs at least one variable x with $|\alpha|_x = 1$. Let $\mathrm{Z} \in \{\mathrm{E}, \mathrm{NE}\}$. The class of Z-pattern languages defined by regular patterns and block-regular patterns are denoted by Z-PAT$_{\Sigma,\mathrm{reg}}$ and Z-PAT$_{\Sigma,\text{b-reg}}$, respectively. To avoid any confusion, we explicitly mention that the term regular pattern always refers to a pattern with the syntactical property of being a regular pattern and a regular E- or NE-pattern language is a pattern language that is regular, but that is not necessarily given by a regular pattern.

In order to prove some of the technical claims in this paper, the following two versions of the pumping lemma for regular languages as stated in Yu [18] can be used.

Lemma 1. *Let $L \subseteq \Sigma^*$ be a regular language. Then there is a constant n, depending on L, such that for every $w \in L$ with $|w| \geq n$ there exist $x, y, z \in \Sigma^*$ such that $w = xyz$ and*

1. $|xy| \leq n$,
2. $|y| \geq 1$,
3. $xy^kz \in L$ *for every* $k \in \mathbb{N}_0$.

Lemma 2. *Let $L \subseteq \Sigma^*$ be a regular language. Then there is a constant n, depending on L, such that for all $u, v, w \in \Sigma^*$, if $|w| \geq n$, then there exist $x, y, z \in \Sigma^*$, $y \neq \varepsilon$, such that $w = xyz$ and, for every $k \in \mathbb{N}_0$, $uxy^kzv \in L$ if and only if $uwv \in L$.*

For the sake of convenience, we shall refer to Lemmas 1 and 2 by *Pumping Lemma* 1 and *Pumping Lemma* 2, respectively. We also need the following generalisation of Ogden's Lemma:

Lemma 3 (Bader and Moura [2]). *Let $L \subseteq \Sigma^*$ be a context-free language. Then there is a constant n, such that for every $z \in L$, if d positions in z are "distinguished" and e positions are "excluded", with $d > n^{(e+1)}$, then there exist $u, v, w, x, y \in \Sigma^*$ such that $z = uvwxy$ and*

1. *vx contains at least one distinguished position and no excluded positions,*
2. *if r is the number of distinguished positions in vwx and s is the number of excluded positions in vwx, then $r \leq n^{(s+1)}$,*
3. *$uv^iwx^iy \in L$ for every $i \in \mathbb{N}_0$.*

Known Characterisations

It can be easily shown that every E- or NE-pattern language over a unary alphabet is a regular language (cf. Reidenbach [13] for further details). Hence, the classes of regular and context-free pattern languages over a unary alphabet are trivially characterised. In Jain et al. [7] it has been shown that for any alphabet of cardinality at least 4, the regular and context-free E-pattern languages are characterised by the class of regular patterns.

Theorem 1 (Jain et al. [7]). *Let Σ be an alphabet with $|\Sigma| \geq 4$. Then* $(\text{E-PAT}_\Sigma \cap \text{REG}) = (\text{E-PAT}_\Sigma \cap \text{CF}) = \text{E-PAT}_{\Sigma,\text{reg}}$.

Unfortunately, the above mentioned cases are the only complete characterisations of regular or context-free pattern languages that are known to date. In particular, characterisations of the regular and context-free E-pattern languages with respect to alphabets with cardinality 2 and 3, and characterisations of the regular and context-free NE-pattern languages with respect to alphabets with cardinality at least 2 are still missing. In the following, we shall briefly summarise the known results in this regard and the reader is referred to Jain et al. [7] and Reidenbach [13] for further details.

Jain et al. [7] show that there exist regular E-pattern languages with respect to alphabet sizes 2 and 3 that cannot be described by regular patterns. Moreover, there exist non-regular context-free E-pattern languages with respect to alphabet sizes 2 and 3. Regarding NE-pattern languages, it is shown that, for every alphabet Σ with cardinality at least 2, the class $(\text{NE-PAT}_\Sigma \cap \text{REG})$ is not characterised by regular patterns and with respect to alphabet sizes 2 and 3 it is not characterised by block-regular patterns either. Furthermore, for alphabet sizes 2 and 3, there exist non-regular context-free NE-pattern languages and for alphabets with cardinality of at least 4 this question is still open.

3 Regularity and Context-Freeness of Pattern Languages: Sufficient Conditions and Necessary Conditions

Since their introduction by Shinohara [17], it has been known that, for both the E and NE case and for any terminal alphabet, regular patterns can only describe regular languages. This is an immediate consequence of the fact that regular patterns do not use the essential mechanism of patterns, i. e., repeating variables in order to define sets of words that contain repeated occurrences of variable factors. In Jain et al. [7], the concept of regular patterns is extended to block-regular patterns, defined in Section 2. By definition, every regular pattern is a block-regular pattern. Furthermore, in the E case, every block-regular pattern α is equivalent to the regular pattern obtained from α by substituting every variable block by a single occurrence of a variable.

Proposition 1. *Let Σ be some terminal alphabet and let $\alpha \in (\Sigma \cup X)^*$ be a pattern. If α is regular, then $L_{\text{NE},\Sigma}(\alpha) \in \text{REG}$. If α is block-regular, then $L_{\text{E},\Sigma}(\alpha) \in \text{REG}$.*

As mentioned in Section 2, for alphabets of size at least 4, both the class of regular patterns and the class of block-regular patterns characterise the set of regular and context-free E-pattern languages. However, in the NE case as well as in the E case with respect to alphabets of size 2 or 3, Jain et al. [7] demonstrate that block-regular patterns do not characterise the set of regular or context-free pattern languages.

Obviously, the regularity of languages given by regular patterns or block-regular patterns follows from the fact that there are variables that occur only once in the pattern. Hence, it is the next logical step to ask whether or not the existence of variables with only one occurrence is also necessary for the regularity or the context-freeness of a pattern language. Jain et al. [7] answer that question with respect to terminal-free patterns.

Theorem 2 (Jain et al. [7]). *Let Σ be a terminal alphabet with $|\Sigma| \geq 2$ and let α be a terminal-free pattern with $|\alpha|_x \geq 2$, for every $x \in \text{var}(\alpha)$. Then $L_{\text{E},\Sigma}(\alpha) \notin \text{CF}$ and $L_{\text{NE},\Sigma}(\alpha) \notin \text{REG}$.*

We can note that Proposition 1 and Theorem 2 characterise the regular and context-free E-pattern languages given by terminal-free patterns with respect to alphabets of size at least 2. More precisely, for every alphabet Σ with $|\Sigma| \geq 2$ and for every terminal-free pattern α, if α is block-regular, then $L_{\mathrm{E},\Sigma}(\alpha)$ is regular (and, thus, also context-free) and if α is not block-regular, then every variable of α occurs at least twice, which implies that $L_{\mathrm{E},\Sigma}(\alpha)$ is neither regular nor context-free.

However, for the NE case, we cannot hope for such a simple characterisation. This is due to the close relationship between the regularity of NE-pattern languages and the combinatorial phenomenon of unavoidable patterns, as already mentioned in Section 1.

In the following, we concentrate on E-pattern languages over alphabets of size 2 and 3 (since for all other alphabet sizes complete characterisations are known) that are given by patterns that are *not* terminal-free (since, as described above, the characterisation of regular and context-free E-pattern languages given by terminal-free patterns has been settled). Nevertheless, some of our results also hold for the NE case and we shall always explicitly mention if this is the case.

The next two results present a sufficient condition for the non-regularity and a sufficient condition for the non-context-freeness of pattern languages over small alphabets. More precisely, we generalise Theorem 2 to patterns that are not necessarily terminal-free. The first result states that for a pattern α (that may contain terminal symbols), if every variable in α occurs at least twice, then both the E- and NE-pattern language of α, with respect to alphabets of size at least two, is not regular.

Theorem 3. *Let Σ be a terminal alphabet with $|\Sigma| \geq 2$, let $\alpha \in (\Sigma \cup X)^*$, and let $\mathrm{Z} \in \{\mathrm{E}, \mathrm{NE}\}$. If, for every $x \in \mathrm{var}(\alpha)$, $|\alpha|_x \geq 2$, then $L_{\mathrm{Z},\Sigma}(\alpha) \notin \mathrm{REG}$.*

For alphabets of size at least 3 we can strengthen Theorem 3, i. e., if every variable in a pattern α occurs at least twice, then the E- and NE-pattern language of α is not context-free.

Theorem 4. *Let Σ be a terminal alphabet with $|\Sigma| \geq 3$, let $\alpha \in (\Sigma \cup X)^+$, and let $Z \in \{\mathrm{E}, \mathrm{NE}\}$. If, for every $x \in \mathrm{var}(\alpha)$, $|\alpha|_x \geq 2$, then $L_{Z,\Sigma}(\alpha) \notin \mathrm{CF}$.*

At this point, we recall that patterns, provided that they contain repeated variables, describe languages that are generalisations of the copy language, which strongly suggests that these languages are context-sensitive, but not context-free or regular. However, as stated in Section 1, for small alphabets this is not necessarily the case and the above results provide a strong indication of where to find this phenomenon of regular and context-free copy languages. More precisely, by Theorems 3 and 4, the existence of variables with only one occurrence is crucial. Furthermore, since, in the terminal-free case, regular and context-free E-pattern languages are characterised in a compact and simple manner, we should also focus on patterns containing terminal symbols.

Consequently, we concentrate on the question of how the occurrences of terminal symbols in conjunction with non-repeated variables can cause E-pattern languages to become regular. To this end, we shall now consider some simply structured examples of such patterns for which we can formally prove whether or not they describe a regular language with respect to terminal alphabets $\Sigma_2 := \{\mathtt{a}, \mathtt{b}\}$ and $\Sigma_{\geq 3}$, where $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\} \subseteq \Sigma_{\geq 3}$. Most parts of the following propositions require individual proofs, some of which, in contrast to the simplicity of the example patterns, are surprisingly involved. If, for some pattern α and $\mathrm{Z} \in \{\mathrm{E}, \mathrm{NE}\}$, $L_{\mathrm{Z},\Sigma_2}(\alpha) \notin \mathrm{REG}$, then $L_{\mathrm{Z},\Sigma_{\geq 3}}(\alpha) \notin \mathrm{REG}$. This follows directly from the fact that regular languages are closed under intersection. Hence, in the following examples, we consider $L_{\mathrm{Z},\Sigma_{\geq 3}}(\alpha)$ only if $L_{\mathrm{Z},\Sigma_2}(\alpha)$ is regular.

Firstly, we consider the pattern $x_1 \cdot d \cdot x_2 x_2 \cdot d' \cdot x_3$, which, for all choices of $d, d' \in \{\mathtt{a}, \mathtt{b}\}$, describes a regular E-pattern language with respect to Σ_2, but a non-regular E-pattern language with respect to $\Sigma_{\geq 3}$.

Proposition 2.

$$\begin{aligned}
L_{\mathrm{E},\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{a} \ x_3) &\in \mathrm{REG}\,,\\
L_{\mathrm{E},\Sigma_{\geq 3}}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{a} \ x_3) &\notin \mathrm{REG}\,,\\
L_{\mathrm{E},\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{b} \ x_3) &\in \mathrm{REG}\,,\\
L_{\mathrm{E},\Sigma_{\geq 3}}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{b} \ x_3) &\notin \mathrm{REG}\,.
\end{aligned}$$

Next, we insert another occurrence of a terminal symbol in between the two occurrences of x_2, i. e., we consider $\beta := x_1 \cdot d \cdot x_2 \cdot d' \cdot x_2 \cdot d'' \cdot x_3$, where $d, d', d'' \in \{\mathtt{a}, \mathtt{b}\}$. Here, we find that $L_{\mathrm{Z},\Sigma}(\beta) \in \mathrm{REG}$ if and only if $\mathrm{Z} = \mathrm{E}$, $\Sigma = \Sigma_2$ and $d = d''$, $d \neq d' \neq d''$.

Proposition 3. *For every $Z \in \{\mathrm{E}, \mathrm{NE}\}$,*

$$\begin{aligned}
L_{Z,\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ \mathtt{a} \ x_2 \ \mathtt{a} \ x_3) &\notin \mathrm{REG}\,,\\
L_{Z,\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ \mathtt{a} \ x_2 \ \mathtt{b} \ x_3) &\notin \mathrm{REG}\,,\\
L_{\mathrm{E},\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ \mathtt{b} \ x_2 \ \mathtt{a} \ x_3) &\in \mathrm{REG}\,,\\
L_{\mathrm{NE},\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ \mathtt{b} \ x_2 \ \mathtt{a} \ x_3) &\notin \mathrm{REG}\,,\\
L_{Z,\Sigma_{\geq 3}}(x_1 \ \mathtt{a} \ x_2 \ \mathtt{b} \ x_2 \ \mathtt{a} \ x_3) &\notin \mathrm{REG}\,.
\end{aligned}$$

The next type of pattern that we investigate is similar to the first one, but it contains two factors of the form xx instead of only one, i. e., $\beta' := x_1 \cdot d \cdot x_2 x_2 \cdot d' \cdot x_3 x_3 \cdot d'' \cdot x_4$, where $d, d', d'' \in \{\mathtt{a}, \mathtt{b}\}$. Surprisingly, $L_{\mathrm{E},\Sigma_2}(\beta')$ is not regular if $d = d' = d''$, but regular in all other cases. However, if we consider the NE case or alphabet $\Sigma_{\geq 3}$, then β' describes a non-regular language with respect to all choices of $d, d', d'' \in \{\mathtt{a}, \mathtt{b}\}$.

Proposition 4. *For every $Z \in \{\mathrm{E}, \mathrm{NE}\}$,*

$$\begin{aligned}
L_{Z,\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{a} \ x_3 \ x_3 \ \mathtt{a} \ x_4) &\notin \mathrm{REG}\,,\\
L_{\mathrm{E},\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{b} \ x_3 \ x_3 \ \mathtt{a} \ x_4) &\in \mathrm{REG}\,,\\
L_{\mathrm{NE},\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{b} \ x_3 \ x_3 \ \mathtt{a} \ x_4) &\notin \mathrm{REG}\,,\\
L_{\mathrm{E},\Sigma_{\geq 3}}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{b} \ x_3 \ x_3 \ \mathtt{a} \ x_4) &\notin \mathrm{REG}\,,\\
L_{\mathrm{E},\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{a} \ x_3 \ x_3 \ \mathtt{b} \ x_4) &\in \mathrm{REG}\,,\\
L_{\mathrm{NE},\Sigma_2}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{a} \ x_3 \ x_3 \ \mathtt{b} \ x_4) &\notin \mathrm{REG}\,,\\
L_{\mathrm{E},\Sigma_{\geq 3}}(x_1 \ \mathtt{a} \ x_2 \ x_2 \ \mathtt{a} \ x_3 \ x_3 \ \mathtt{b} \ x_4) &\notin \mathrm{REG}\,.
\end{aligned}$$

We call two patterns $\alpha, \beta \in (\Sigma_2 \cup X)^*$ *almost identical* if and only if $|\alpha| = |\beta|$ and, for every i, $1 \leq i \leq |\alpha|$, $\alpha[i] \neq \beta[i]$ implies $\alpha[i], \beta[i] \in \Sigma_2$. The above examples show that even for almost identical patterns α and β, we can have the situation that α describes a regular and β a non-regular language. Even if α and β are almost identical and further satisfy $|\alpha|_{\mathtt{a}} = |\beta|_{\mathtt{a}}$ and $|\alpha|_{\mathtt{b}} = |\beta|_{\mathtt{b}}$, then it is still possible that α describes a regular and β a non-regular language (cf. Proposition 3 above). This implies that the regular E-pattern languages over an alphabet with size 2 require a characterisation that caters for the exact order of terminal symbols in the patterns.

The examples considered in Propositions 2 and 4 mainly consist of factors of the form $d \cdot xx \cdot d'$, $d, d' \in \Sigma_2$, where x does not have any other occurrence in the pattern. Hence, it might be worthwhile to investigate the question of whether or not patterns can also describe regular languages if we allow them to contain factors of the form $d \cdot x^k \cdot d'$, where $k \geq 3$ and there is no other occurrence of x in the pattern. In the next result, we state that if a pattern α contains a factor $d \cdot x^k \cdot d'$ with $d = d'$, $k \geq 3$ and $|\alpha|_x = k$, then, for every $\mathrm{Z} \in \{\mathrm{E}, \mathrm{NE}\}$, its Z-pattern language with respect to any alphabet of size at least 2 is not regular and, furthermore, for alphabets of size at least 3, we can show that this also holds for $d \neq d'$.

Theorem 5. *Let Σ and Σ' be terminal alphabets with $\{\mathtt{a}, \mathtt{b}\} \subseteq \Sigma$ and $\{\mathtt{a}, \mathtt{b}, \mathtt{c}\} \subseteq \Sigma'$. Let $\alpha := \alpha_1 \cdot \mathtt{a} \cdot z^l \cdot \mathtt{a} \cdot \alpha_2$, let $\beta := \beta_1 \cdot \mathtt{a} \cdot z^l \cdot \mathtt{c} \cdot \beta_2$, where $z \in X$, $\alpha_1, \alpha_2 \in ((\Sigma \cup X) \setminus \{z\})^*$, $\beta_1, \beta_2 \in ((\Sigma' \cup X) \setminus \{z\})^*$ and $l \geq 3$. Then, for every $\mathrm{Z} \in \{\mathrm{E}, \mathrm{NE}\}$, $L_{\mathrm{Z},\Sigma}(\alpha) \notin \mathrm{REG}$ and $L_{\mathrm{Z},\Sigma'}(\beta) \notin \mathrm{REG}$.*

In the examples of Propositions 2, 3 and 4 as well as in the above theorem, we did not consider the situation that two occurrences of the same variable are separated by a terminal symbol. In the next result, we state that, in certain cases, this situation implies non-regularity of pattern languages.

Proposition 5. *Let Σ and Σ' be terminal alphabets with $|\Sigma| \geq 2$ and $|\Sigma'| \geq 3$ and let $\mathrm{Z} \in \{\mathrm{E}, \mathrm{NE}\}$. Furthermore, let $\alpha_1 \in (\Sigma \cup X)^*$ and $\alpha_2 \in (\Sigma' \cup X)^*$ be patterns.*

1. *If there exists a $\gamma \in (\Sigma \cup X)^*$ with $|\operatorname{var}(\gamma)| \geq 1$ such that, for some $d \in \Sigma$,*
   - *$\alpha_1 = \gamma \cdot d \cdot \delta$ and $\operatorname{var}(\gamma) \subseteq \operatorname{var}(\delta)$,*
   - *$\alpha_1 = \gamma \cdot d \cdot \delta$ and $\operatorname{var}(\delta) \subseteq \operatorname{var}(\gamma)$ or*
   - *$\alpha_1 = \beta \cdot d \cdot \gamma \cdot d \cdot \delta$ and $\operatorname{var}(\gamma) \subseteq (\operatorname{var}(\beta) \cup \operatorname{var}(\delta))$,*

   then $L_{\mathrm{Z},\Sigma}(\alpha_1) \notin \mathrm{REG}$.
2. *If in α_2 there exists a non-empty variable block, all the variables of which also occur outside this block, then $L_{\mathrm{Z},\Sigma'}(\alpha_2) \notin \mathrm{REG}$.*

We conclude this section by referring to the examples presented in Propositions 2, 3 and 4, which, as described above, suggest that complete characterisations of the regular E-pattern languages over small alphabets might be extremely complex. In the next section, we wish to find out about the fundamental mechanisms of the above example patterns that are responsible for the regularity of their pattern languages. Intuitively speaking, some of the above example patterns describe regular languages, because they contain a factor that is less complex than it seems to be, e. g., for the pattern $\beta := x_1 \cdot \mathtt{a} \cdot x_2x_2 \cdot \mathtt{a} \cdot x_3x_3 \cdot \mathtt{b} \cdot x_4$ it can be shown that the factor $\mathtt{a} \cdot x_2x_2 \cdot \mathtt{a} \cdot x_3x_3 \cdot \mathtt{b}$ could be replaced by $\mathtt{a} \cdot x_{(\mathtt{bb})^*} \cdot \mathtt{a} \cdot \mathtt{b}$ (where $x_{(\mathtt{bb})^*}$ is a special variable that can only be substituted by a unary string over $\mathtt{b}$ of even length) without changing its E-pattern language with respect to Σ_2. This directly implies that $L_{\mathrm{E},\Sigma_2}(\beta) = L(\Sigma_2^* \cdot \mathtt{a}(\mathtt{bb})^*\mathtt{ab} \cdot \Sigma_2^*)$, which shows that $L_{\mathrm{E},\Sigma_2}(\beta) \in \mathrm{REG}$. In the next section, we generalise this observation.

4 Regularity of E-Pattern Languages: A Sufficient Condition Taking Terminal Symbols into Account

In this section we investigate the phenomenon that a whole factor in a pattern can be substituted by a less complex one, without changing the corresponding pattern language. This technique can be used in order to show that a complicated pattern is equivalent to one that can be easily seen to describe a regular language.

For the sake of a better presentation of our results, we slightly redefine the concept of patterns. A *pattern with regular expressions* is a pattern that may contain regular expressions. Such a regular expressions is then interpreted as a variable with only one occurrence that can only be substituted by words described by the corresponding regular expression. For example $L_{\mathrm{E},\Sigma_2}(x_1\mathtt{b}^*x_1\mathtt{a}^*) = \{h(x_1x_2x_1x_3) \mid h \text{ is a substitution with } h(x_2) \in L(\mathtt{b}^*), h(x_3) \in L(\mathtt{a}^*)\}$. Obviously, patterns with regular expressions exceed the expressive power of classical patterns. However, we shall use this concept exclusively in the case where a classical pattern is equivalent to a pattern with regular expressions. For example, the pattern $x_1 \cdot \mathtt{a} \cdot x_2x_3x_3x_2 \cdot \mathtt{a} \cdot x_4$ is equivalent to the pattern $x_1 \cdot \mathtt{a}(\mathtt{bb})^*\mathtt{a} \cdot x_2$ (see Lemma 6).

Next, we present a lemma that states that in special cases whole factors of a pattern can be removed without changing the corresponding pattern language.

Lemma 4. *Let* $\alpha := \beta \cdot y \cdot \beta' \cdot \mathtt{a} \cdot \gamma \cdot \mathtt{b} \cdot \delta' \cdot z \cdot \delta$*, where* $\beta, \delta \in (\Sigma_2 \cup X)^*$*,* $\beta', \gamma, \delta' \in X^*$*,* $y, z \in X$ *and* $|\alpha|_y = |\alpha|_z = 1$*. Then* $L_{\mathrm{E},\Sigma_2}(\alpha) \subseteq L_{\mathrm{E},\Sigma_2}(\beta \cdot y \cdot \mathtt{ab} \cdot z \cdot \delta)$*. If, furthermore,* $\mathrm{var}(\beta' \cdot \gamma \cdot \delta') \cap \mathrm{var}(\beta \cdot \delta) = \emptyset$*, then also* $L_{\mathrm{E},\Sigma_2}(\beta \cdot y \cdot \mathtt{ab} \cdot z \cdot \delta) \subseteq L_{\mathrm{E},\Sigma_2}(\alpha)$*.*

The fact that $L_{\mathrm{E},\Sigma_2}(x_1 \cdot \mathtt{a} \cdot x_2x_2 \cdot \mathtt{b} \cdot x_3) \in \mathrm{REG}$, which has already been stated in Proposition 2, is a simple application of Lemma 4, which implies $L_{\mathrm{E},\Sigma_2}(x_1 \cdot \mathtt{a} \cdot x_2x_2 \cdot \mathtt{b} \cdot x_3) = L_{\mathrm{E},\Sigma_2}(x_1 \cdot \mathtt{ab} \cdot x_3)$. It is straightforward to construct more complex applications of Lemma 4 and it is also possible to apply it in an iterative way. For example, by applying Lemma 4 twice, we can show that

$$
\begin{aligned}
&L_{\mathrm{E},\Sigma_2}(x_1x_2x_3 \cdot \mathtt{a} \cdot x_2x_4 \cdot \mathtt{b} \cdot x_3x_4x_5x_6 \cdot \mathtt{b} \cdot x_6x_7 \cdot \mathtt{a} \cdot x_7x_8 \cdot \mathtt{b} \cdot x_9 \cdot \mathtt{a} \cdot x_{10}) = \\
&L_{\mathrm{E},\Sigma_2}(x_1 \cdot \mathtt{ab} \cdot x_5x_6 \cdot \mathtt{b} \cdot x_6x_7 \cdot \mathtt{a} \cdot x_7x_8 \cdot \mathtt{b} \cdot x_9 \cdot \mathtt{a} \cdot x_{10}) = \\
&L_{\mathrm{E},\Sigma_2}(x_1 \cdot \mathtt{ab} \cdot x_5 \cdot \mathtt{ba} \cdot x_8 \cdot \mathtt{b} \cdot x_9 \cdot \mathtt{a} \cdot x_{10}) \in \mathrm{REG}\ .
\end{aligned}
$$

In the previous lemma, it is required that the factor γ is delimited by different terminal symbols and, in the following, we shall see that an extension of the statement of Lemma 4 for the case that γ is delimited by the same terminal symbols, is much more difficult to prove.

Roughly speaking, Lemma 4 holds due to the following reasons. Let $\alpha := y \cdot \beta' \cdot \mathtt{a} \cdot \gamma \cdot \mathtt{b} \cdot \delta' \cdot z$ be a pattern that satisfies the conditions of Lemma 4, then, for any substitution h (with respect to Σ_2), $h(\alpha)$ necessarily contains the factor $\mathtt{ab}$. Conversely, since y and z are variables with only one occurrence and there are no terminals in $\beta' \cdot \gamma \cdot \delta'$, α can be mapped to every word that contains the factor $\mathtt{ab}$. On the other hand, for $\alpha' := y \cdot \beta' \cdot \mathtt{a} \cdot \gamma \cdot \mathtt{a} \cdot \delta' \cdot z$, $h(\alpha')$ does not necessarily contain the factor $\mathtt{aa}$ and it is not obvious if the factor $\beta' \cdot \mathtt{a} \cdot \gamma \cdot \mathtt{a} \cdot \delta'$ collapses to some simpler structure, as it is the case for α. In fact, Theorem 5 states that if $\beta' = \delta' = \varepsilon$ and $\gamma = x^3$, then $L_{\mathrm{E},\Sigma_2}(\alpha') \notin \mathrm{REG}$.

However, by imposing a further restriction with respect to the factor γ, we can extend Lemma 4 to the case where γ is delimited by the same terminal symbol. In order to prove this result, the next lemma is crucial, which states that for any terminal-free pattern that is delimited by two occurrences of symbols $\mathtt{a}$ and that has an even number of occurrences for every variable, if we apply any substitution to this pattern, we will necessarily obtain a word that contains a unary factor over $\mathtt{b}$ of even length that is delimited by two occurrences of $\mathtt{a}$.

Lemma 5. *Let $\alpha \in X^*$ such that, for every $x \in \mathrm{var}(\alpha)$, $|\alpha|_x$ is even. Then every $w \in L_{\mathrm{E},\Sigma_2}(\mathtt{a} \cdot \alpha \cdot \mathtt{a})$ contains a factor $\mathtt{ab}^{2n}\mathtt{a}$, $n \in \mathbb{N}_0$.*

By applying Lemma 5, we can show that if a pattern $\alpha := \beta \cdot y \cdot \beta' \cdot \mathtt{a} \cdot \gamma \cdot \mathtt{a} \cdot \delta' \cdot z \cdot \delta$ satisfies the conditions of Lemma 4, all variables in γ have an even number of occurrences and there is at least one variable in γ that occurs only twice, then the factor $y \cdot \beta' \cdot \mathtt{a} \cdot \gamma \cdot \mathtt{a} \cdot \delta' \cdot z$ can be substituted by a regular expression.

Lemma 6. *Let $\alpha := \beta \cdot y \cdot \beta' \cdot \mathtt{a} \cdot \gamma \cdot \mathtt{a} \cdot \delta' \cdot z \cdot \delta$, where $\beta, \delta \in (\Sigma_2 \cup X)^*$, $\beta', \gamma, \delta' \in X^*$, $y, z \in X$, $|\alpha|_y = |\alpha|_z = 1$ and, for every $x \in \mathrm{var}(\gamma)$, $|\gamma|_x$ is even. Then $L_{\mathrm{E},\Sigma_2}(\alpha) \subseteq L_{\mathrm{E},\Sigma_2}(\beta \cdot y \cdot \mathtt{a}(\mathtt{bb})^*\mathtt{a} \cdot z \cdot \delta)$. If, furthermore, $\mathrm{var}(\beta' \cdot \gamma \cdot \delta') \cap \mathrm{var}(\beta \cdot \delta) = \emptyset$ and there exists a $z' \in \mathrm{var}(\gamma)$ with $|\alpha|_{z'} = 2$, then also $L_{\mathrm{E},\Sigma_2}(\beta \cdot y \cdot \mathtt{a}(\mathtt{bb})^*\mathtt{a} \cdot z \cdot \delta) \subseteq L_{\mathrm{E},\Sigma_2}(\alpha)$.*

Obviously, Lemmas 4 and 6 can also be applied in any order in the iterative way pointed out above with respect to Lemma 4. We shall illustrate this now in a more general way. Let α be an arbitrary pattern such that

$$\alpha := \beta \cdot y_1 \cdot \beta'_1 \cdot \mathtt{a} \cdot \gamma_1 \cdot \mathtt{a} \cdot \delta'_1 \cdot z_1 \cdot \pi \cdot y_2 \cdot \beta'_2 \cdot \mathtt{b} \cdot \gamma_2 \cdot \mathtt{a} \cdot \delta'_2 \cdot z_2 \cdot \delta\,,$$

with $\beta, \pi, \delta \in (\Sigma_2 \cup X)^*$, $\beta'_1, \beta'_2, \gamma_1, \gamma_2, \delta'_1, \delta'_2 \in X^*$ and $y_1, y_2, z_1, z_2 \in X$. If the factors $y_1 \cdot \beta'_1 \cdot \mathtt{a} \cdot \gamma_1 \cdot \mathtt{a} \cdot \delta'_1 \cdot z_1$ and $y_2 \cdot \beta'_2 \cdot \mathtt{b} \cdot \gamma_2 \cdot \mathtt{a} \cdot \delta'_2 \cdot z_2$ satisfy the conditions of Lemma 6 and Lemma 4, respectively, then we can conclude that α is equivalent to $\alpha' := \beta \cdot y_1 \cdot \mathtt{a}(\mathtt{bb})^*\mathtt{a} \cdot z_1 \cdot \pi \cdot y_2 \cdot \mathtt{ba} \cdot z_2 \cdot \delta$. This particularly means that the rather strong conditions

1. $\mathrm{var}(\beta'_1 \cdot \gamma_1 \cdot \delta'_1) \cap \mathrm{var}(\beta \cdot \pi \cdot \beta'_2 \cdot \gamma_2 \cdot \delta'_2 \cdot \delta) = \emptyset$,
2. $\mathrm{var}(\beta'_2 \cdot \gamma_2 \cdot \delta'_2) \cap \mathrm{var}(\beta \cdot \beta'_1 \cdot \gamma_1 \cdot \delta'_1 \cdot \pi \cdot \delta) = \emptyset$

must be satisfied. However, we can state that $L_{\mathrm{E},\Sigma_2}(\alpha) = L_{\mathrm{E},\Sigma_2}(\alpha')$ still holds if instead of conditions 1 and 2 from above the weaker condition $\mathrm{var}(\beta'_1 \cdot \gamma_1 \cdot \delta'_1 \cdot \beta'_2 \cdot \gamma_2 \cdot \delta'_2) \cap \mathrm{var}(\beta \cdot \pi \cdot \delta) = \emptyset$ is satisfied. This claim can be easily proved by applying the same argumentations as in the proofs of Lemmas 4 and 6, and we can extend this result to arbitrarily many factors of the form $y_i \cdot \beta'_i \cdot c_1 \cdot \gamma_i \cdot c_2 \cdot \delta'_i \cdot z_i$, $c_1, c_2 \in \Sigma_2$. Next, by the following definition, we formalise this observation in terms of a relation on patterns with regular expressions.

Definition 2. *For any two patterns with regular expressions α and α', we write $\alpha \rhd \alpha'$ if and only if the following conditions are satisfied.*

- *α contains factors $\alpha_i \in (\Sigma_2 \cup X)^*$, $1 \leq i \leq k$, where, for every i, $1 \leq i \leq k$, $\alpha_i := y_i \cdot \beta'_i \cdot d_i \cdot \gamma_i \cdot d'_i \cdot \delta'_i \cdot z_i$, with $\beta'_i, \gamma_i, \delta'_i \in X^+$, $y_i, z_i \in X$, $|\alpha|_{y_i} = |\alpha|_{z_i} = 1$, $d_i, d'_i \in \Sigma_2$ and, if $d_i = d'_i$, then, for every $x \in \mathrm{var}(\gamma_i)$, $|\gamma_i|_x$ is even and there exists an $x' \in \mathrm{var}(\gamma_i)$ with $|\alpha|_{x'} = 2$. Furthermore, the factors $\alpha_1, \alpha_2, \dots, \alpha_k$ can overlap by at most one symbol and the variables in the factors $\alpha_1, \alpha_2, \dots, \alpha_k$ occur exclusively in these factors.*
- *α' is obtained from α by substituting every α_i, $1 \leq i \leq k$, by $y_i \cdot d_i d'_i \cdot z_i$, if $d_i \neq d'_i$ and by $y_i \cdot d_i(d''_i d''_i)^* d'_i \cdot z_i$, $d''_i \in \Sigma_2$, $d''_i \neq d_i$, if $d_i = d'_i$.*

By generalising Lemmas 4 and 6, we can prove that $\alpha \rhd \alpha'$ implies that α and α' describe the same E-pattern language with respect to alphabet Σ_2.

Theorem 6. *Let α and α' be patterns with regular expressions. If $\alpha \rhd \alpha'$, then $L_{\mathrm{E},\Sigma_2}(\alpha) = L_{\mathrm{E},\Sigma_2}(\alpha')$.*

We conclude this section by discussing a more complex example that illustrates how Definition 2 and Theorem 6 constitute a sufficient condition for the regularity of the E-pattern language of a pattern with respect to Σ_2. Let α be the following pattern.

$$\underbrace{x_1 \mathtt{a} x_2 x_3^2 \mathtt{b} x_4 x_3 x_5 x_6}_{\alpha_1 := y_1 \cdot \beta'_1 \cdot \mathtt{a} \cdot \gamma_1 \cdot \mathtt{b} \cdot \delta'_1 \cdot z_1} x_7^2 \underbrace{x_8 x_9 x_5 x_3 \mathtt{a} x_4 x_5 x_4 x_9 x_{10} \mathtt{b} x_{11}}_{\alpha_2 := y_2 \cdot \beta'_2 \cdot \mathtt{a} \cdot \gamma_2 \cdot \mathtt{b} \cdot \delta'_2 \cdot z_2} \mathtt{a} x_{12} \mathtt{b} x_{13} \mathtt{a} \underbrace{x_{14} x_{15} \mathtt{b} x_{15}^2 x_{16}^2 \mathtt{b} x_{17}}_{\alpha_3 := y_3 \cdot \beta'_3 \cdot \mathtt{a} \cdot \gamma_3 \cdot \mathtt{b} \cdot \delta'_3 \cdot z_3} .$$

By Definition 2, $\alpha \rhd \beta$ holds, where β is obtained from α by substituting the above defined factors α_1, α_2 and α_3 by factors $x_1 \cdot \mathtt{ab} \cdot x_6$, $x_8 \cdot \mathtt{ab} \cdot x_{11}$ and $x_{14} \cdot \mathtt{b}(\mathtt{aa})^* \mathtt{b} \cdot x_{17}$, respectively, i. e.,

$$\beta := x_1 \mathtt{ab} x_6 x_7 x_7 x_8 \mathtt{ab} x_{11} \mathtt{a} x_{12} \mathtt{b} x_{13} \mathtt{a} x_{14} \mathtt{b}(\mathtt{aa})^* \mathtt{b} x_{17} .$$

Furthermore, by Theorem 6, we can conclude that $L_{\mathrm{E},\Sigma_2}(\alpha) = L_{\mathrm{E},\Sigma_2}(\beta)$. However, we can also apply the same argumentation to different factors of α, as pointed out below:

$$x_1 \mathtt{a} \underbrace{x_2 x_3^2 \mathtt{b} x_4 x_3 x_5 x_6 x_7^2 x_8 x_9 x_5 x_3 \mathtt{a} x_4 x_5 x_4 x_9 x_{10}}_{\alpha_1 := y_1 \cdot \beta'_1 \cdot \mathtt{a} \cdot \gamma_1 \cdot \mathtt{b} \cdot \delta'_1 \cdot z_1} \mathtt{b} x_{11} \mathtt{a} x_{12} \mathtt{b} x_{13} \mathtt{a} \underbrace{x_{14} x_{15} \mathtt{b} x_{15}^2 x_{16}^2 \mathtt{b} x_{17}}_{\alpha_2 := y_2 \cdot \beta'_2 \cdot \mathtt{a} \cdot \gamma_2 \cdot \mathtt{b} \cdot \delta'_2 \cdot z_2} .$$

Now, again by Definition 2, $\alpha \rhd \beta'$ is satisfied, where

$$\beta' := x_1 \mathtt{a} x_2 \mathtt{ba} x_{10} \mathtt{b} x_{11} \mathtt{a} x_{12} \mathtt{b} x_{13} \mathtt{a} x_{14} \mathtt{b}(\mathtt{aa})^* \mathtt{b} x_{17} .$$

Since every variable of β' has only one occurrence, it can be easily seen that $L_{\mathrm{E},\Sigma_2}(\beta') \in \mathrm{REG}$ and, by Theorem 6, $L_{\mathrm{E},\Sigma_2}(\alpha) \in \mathrm{REG}$ follows.

References

1. D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.
2. C. Bader and A. Moura. A generalization of Ogden's Lemma. *Journal of the Association for Computing Machinery*, 29:404–407, 1982.
3. C. Câmpeanu, K. Salomaa, and S. Yu. A formal study of practical regular expressions. *International Journal of Foundations of Computer Science*, 14:1007–1018, 2003.
4. D.D. Freydenberger and D. Reidenbach. Bad news on decision problems for patterns. *Information and Computation*, 208:83–96, 2010.
5. D.D. Freydenberger, D. Reidenbach, and J.C. Schneider. Unambiguous morphic images of strings. *International Journal of Foundations of Computer Science*, 17:601–628, 2006.
6. T. Harju and J. Karhumäki. Morphisms. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, chapter 7, pages 439–510. Springer, 1997.
7. S. Jain, Y. S. Ong, and F. Stephan. Regular patterns, regular languages and context-free languages. *Information Processing Letters*, 110:1114–1119, 2010.
8. 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.
9. S. Lange and R. Wiehagen. Polynomial-time inference of arbitrary pattern languages. *New Generation Computing*, 8:361–370, 1991.
10. A. Mateescu and A. Salomaa. Finite degrees of ambiguity in pattern languages. *RAIRO Informatique théoretique et Applications*, 28:233–253, 1994.
11. A. Mateescu and A. Salomaa. Patterns. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, pages 230–242. Springer, 1997.
12. Y.K. Ng and T. Shinohara. Developments from enquiries into the learnability of the pattern languages from positive data. *Theoretical Computer Science*, 397:150–165, 2008.
13. D. Reidenbach. *The Ambiguity of Morphisms in Free Monoids and its Impact on Algorithmic Properties of Pattern Languages*. PhD thesis, Fachbereich Informatik, Technische Universität Kaiserslautern, 2006. Logos Verlag, Berlin.
14. D. Reidenbach. Discontinuities in pattern inference. *Theoretical Computer Science*, 397:166–193, 2008.
15. P. Rossmanith and T. Zeugmann. Stochastic finite learning of the pattern languages. *Machine Learning*, 44:67–91, 2001.
16. T. Shinohara. Polynomial time inference of extended regular pattern languages. In *Proc. RIMS Symposia, Kyoto*, volume 147 of *LNCS*, pages 115–127, 1982.
17. T. Shinohara. Polynomial time inference of pattern languages and its application. In *Proc. 7th IBM MFCS*, pages 191–209, 1982.
18. S. Yu. Regular languages. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, chapter 2, pages 41–110. Springer, 1997.