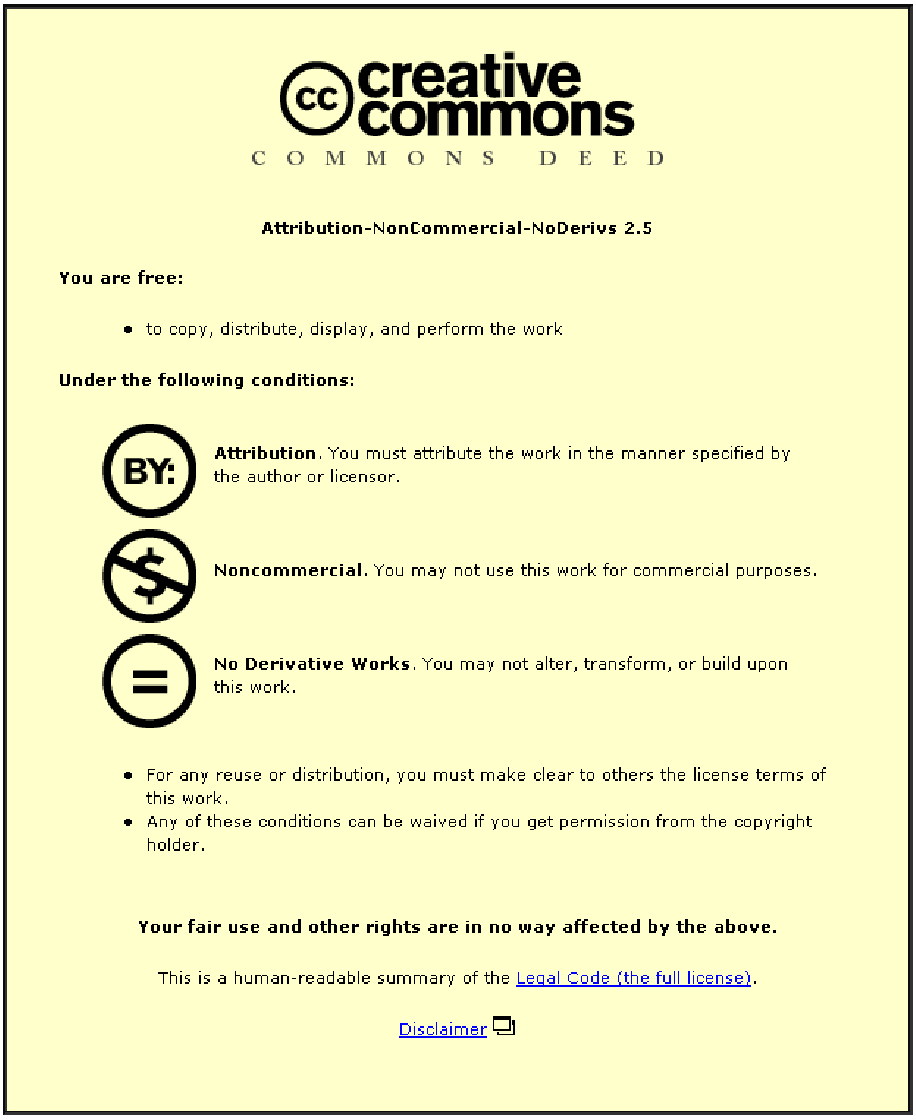

Existence and Nonexistence of Descriptive Patterns

Dominik D. Freydenberger[1] and Daniel Reidenbach[2*]

[1] Institut für Informatik, Goethe-Universität, Postfach 111932,
D-60054 Frankfurt am Main, Germany
freydenberger@em.uni-frankfurt.de

[2] Department of Computer Science, Loughborough University,
Loughborough, Leicestershire, LE11 3TU, United Kingdom
D.Reidenbach@lboro.ac.uk

Abstract. In the present paper, we study the existence of descriptive patterns, i. e. patterns that cover all words in a given set through morphisms and that are optimal in terms of revealing commonalities of these words. Our main result shows that if patterns may be mapped onto words by arbitrary morphisms, then there exist infinite sets of words that do not have a descriptive pattern. This answers a question posed by Jiang, Kinber, Salomaa, Salomaa and Yu (*International Journal of Computer Mathematics* 50, 1994). Since the problem of whether a pattern is descriptive depends on the inclusion relation of so-called pattern languages, our technical considerations lead to a number of deep insights into the inclusion problem for and the topology of the class of terminal-free E-pattern languages.

1 On Patterns Descriptive of a Set of Strings

A *pattern* is a finite string that consists of variables taken from an alphabet X and terminal symbols taken from an alphabet Σ. For any pattern α and any word w over Σ, α is said to cover w if w can be obtained from α by substituting the variables with appropriate strings of terminal symbols. Whenever α contains several occurrences of the same variable, the substitution of variables needs to be "uniform", i. e. each of the occurrences must be replaced with the same word over Σ. Therefore, and more formally, such a substitution is simply a terminal-preserving morphism $\sigma : (\Sigma \cup X)^* \to \Sigma^*$, i. e. a morphism that satisfies $\sigma(a) = a$ for every terminal symbol a in the pattern. For instance, the pattern $\alpha := xy\mathtt{b}x\mathtt{a}$ (where x, y are variables and $\mathtt{a}, \mathtt{b}$ are terminal symbols) covers the word $w_1 := \mathtt{abababa}$ since there is a substitution σ, given by $\sigma(x) := \mathtt{ab}$ and $\sigma(y) := \mathtt{a}$, satisfying $\sigma(\alpha) = w$. In contrast to this, α does not cover, e. g., $w_2 := \mathtt{bbbbaa}$.

Due to the simplicity of the concepts involved, the above described notion of a pattern is studied in a variety of fields of research. The present paper mainly deals with two quite closely related approaches: Firstly, a pattern α over $\Sigma \cup X$ can be

[*] Corresponding author.

regarded as a generator of a formal language $L(\alpha)$, the so-called *pattern language*, which simply comprises all words in Σ^* that can be obtained from the pattern by arbitrary substitutions. Secondly, for any given finite or infinite language S, patterns can be used to approximate S; i. e., a pattern α is sought that is *consistent* with S (which means that α covers all words in S or alternatively, in terms of pattern languages, $L(\alpha) \supseteq S$). The latter concept is motivated by the fact that if a pattern is consistent with a language S, then this pattern reveals a common structure of the strings in S. Hence, and since they are compact devices that can be easily read and interpreted by humans, patterns can be very helpful when commonalities of data represented by strings are analysed.

The characteristics of pattern languages have been intensively studied in the past decades. Therefore, quite a number of basic properties of pattern languages, e. g. regarding the usual decision problems for classes of formal languages, are known (cf. the surveys by Mateescu and Salomaa [7] and Salomaa [11] and our recent paper [4]). Furthermore, pattern languages have been a focus of interest of inductive inference from the very beginning, investigating whether it is possible to infer a pattern from the words in its pattern language (see Ng and Shinohara [8]). It is quite remarkable that many of the corresponding results in language theory and inductive inference differ for the two main types of pattern languages that are normally considered, namely the *NE*-pattern language of a pattern (introduced by Angluin [1]), which merely consists of those words in Σ^* that can be obtained from the pattern by *nonerasing* substitutions (i. e. substitutions that do not replace any variables with the empty word), and the E-pattern language (established by Shinohara [12]), which additionally comprises those words that can be derived from the pattern by substituting the empty word for arbitrary variables.

The problem of finding a consistent pattern for an arbitrary set S of strings is often referred to as *(string) pattern discovery*, and many of its applications are derived from tasks in bioinformatics (cf. Brazma et al. [2]). In contrast to the inductive inference approach to pattern languages, where a pattern shall be inferred that exactly describes the given language, string pattern discovery faces the problem that S can typically have many consistent patterns showing very different characteristics. For instance, both

$$\alpha_1 := xyxyx \text{ and } \alpha_2 := x\mathtt{ab}y$$

are consistent with the language

$$S_0 := \{\mathtt{ababa}, \mathtt{ababbababbab}, \mathtt{babab}\},$$

and the pattern $\alpha_0 := x$ is consistent with every set of strings, anyway. Hence, the algorithms of string pattern discovery require an underlying notion of the quality of a pattern in order to determine what patterns to strive for. With regard to the above example set and patterns, it seems quite likely that one might not be interested in a procedure outputting α_0 when reading S_0. Concerning α_1 and α_2, however, it is, a priori, by no means evident which of them to prefer. Thus, the definition of the quality of a pattern might often depend on the field of application where string pattern discovery is conducted. In addition to this, it is a worthwhile goal to develop *generic* notions of quality of consistent patterns that can inform the design of pattern discovery algorithms.

In this regard, the *descriptiveness* of patterns is a well-known and plausible concept, that is also used within the scope of inductive inference (cf. Ng and Shinohara [8]). A pattern δ is said to be descriptive of a given set S of strings if there is no pattern α satisfying $S \subseteq L(\alpha) \subset L(\delta)$. Intuitively, this means that if δ is descriptive of S, then no consistent pattern for S provides a strictly closer match than δ. Thus, although δ does not need to be unique (as to be further discussed below), it is guaranteed that it is one of the most accurate approximations of S that can be provided by patterns. While descriptiveness is unquestionably an appropriate notion of quality of consistent patterns, it leads to major technical challenges, as its application requires insights into the inclusion problem for pattern languages, which is known to be undecidable in the general case and still combinatorially involved for some major natural subclasses where it is decidable. This aspect is crucial to the subsequent formal parts of our paper.

Since the definition of a descriptive pattern is based on the concept of pattern languages, the question of whether NE- or E-pattern languages are chosen can have a significant impact on the descriptiveness of a pattern. This is reflected by the terminology we use: we call a pattern δ an NE-descriptive pattern if it is descriptive in terms of its NE-pattern language and the NE-pattern languages of all patterns in $(\Sigma \cup X)^+$; accordingly, we call δ E-descriptive if its descriptiveness is based on interpreting all patterns as generators of E-pattern languages. In order to illustrate these terms, we now briefly discuss the descriptiveness of the example patterns introduced above (though the full verification of our corresponding claims is not always straightforward and might require certain tools to be introduced later). If we deal with S_0 and the patterns in the context of NE-pattern languages, then it can be stated that both α_1 and α_2 are NE-descriptive of S_0, since no NE-pattern languages can comprise S_0 and, at the same time, be a proper sublanguage of the NE-pattern languages of α_1 or α_2. If we study S_0 in terms of E-pattern languages, it turns out that α_1 is also E-descriptive of S_0, i. e. there is no pattern generating an E-pattern language that is consistent with S_0 and strictly included in the E-pattern language of α_1. However, the second NE-descriptive example pattern α_2 is not E-descriptive of S_0, since the E-pattern language generated by

$$\alpha_3 := x\mathtt{abab}y$$

is a proper sublanguage of the E-pattern language of α_2 and comprises S_0. The pattern α_3, in turn, is even E-descriptive of S_0, but not NE-descriptive, since it is not consistent with S_0 if we disallow empty substitutions. Exactly the same holds for $\alpha_4 := x\mathtt{baba}y$, which also is consistent with S_0 if we allow the empty substitution of variables, generates an E-pattern language that is strictly included in the E-pattern language of α_2 and is E-descriptive, but not NE-descriptive of S_0.

The present paper examines the basic underlying problem of descriptive pattern discovery, namely the *existence* of such patterns; this means that we study the question of whether or not, for a given language S, there is a pattern that is descriptive of S. To this end, four different cases can be considered: NE-descriptive patterns of finite languages, NE-descriptive patterns of infinite languages, E-descriptive patterns of finite languages and E-descriptive patterns of infinite languages. The problem of the existence of the former three types of descriptive patterns is either trivial or has already been solved in previous publications. We therefore largely study the latter case, and our corresponding main result answers a question posed by Jiang, Kinber, Salomaa, Salomaa and Yu [5]. Our technical considerations do not only provide insights into the actual topic of our paper, but – due to the definition of descriptive patterns – also reveal vital phenomena related to the inclusion of E-pattern languages and, hence, the topology of class of terminal-free E-pattern languages. Due to the way the inclusion of terminal-free E-pattern languages is characterised, this implies that we have to deal with combinatorial properties of morphisms in free monoids.

2 Basic Definitions and Preparatory Technical Considerations

This paper is largely self-contained. For notations not explicitly defined, Rozenberg and Salomaa [10] can be consulted.

Let $\mathbb{N} := \{0, 1, 2, 3, \ldots\}$ and, for every $k \geq 0$, $\mathbb{N}_k := \{n \in \mathbb{N} \mid n \geq k\}$. The symbols $\subseteq$, $\subset$, $\supseteq$ and $\supset$ refer to subset, proper subset, superset and proper superset relation, respectively. The symbol ∞ stands for infinity. For an arbitrary alphabet A, a *string* (over A) is a finite sequence of symbols from A, and λ stands for the *empty string*. The symbol A^+ denotes the set of all nonempty strings over A, and $A^* := A^+ \cup \{\lambda\}$. For the *concatenation* of two strings w_1, w_2 we write $w_1 \cdot w_2$ or simply $w_1 w_2$. 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 $w = u_1 v u_2$. If $u_1 = \lambda$ (or $u_2 = \lambda$), then v is a *prefix* of w (or a *suffix*, respectively). The notation $|K|$ stands for the size of a set K or the length of a string K; the term $|w|_a$ refers to the number of occurrences of the symbol a in the string w. For any $w \in \Sigma^*$ and any $n \in \mathbb{N}$, w^n denotes the *n-fold concatenation of* w, with $w^0 := \lambda$.

For any alphabets A, B, a *morphism* is a function $h : A^* \to B^*$ that satisfies $h(vw) = h(v)h(w)$ for all $v, w \in A^*$. Given morphisms $g : A^* \to B^*$ and $h : B^* \to C^*$ (for alphabets A, B, C), their *composition* $(h \circ g)$ is defined by $(h \circ g)(w) := h(g(w))$ for all $w \in A^*$. For every morphism $h : A^* \to A^*$ and every $n \geq 0$, h^n denotes the *n-fold iteration of* h, i. e., $h^{n+1} := (h \circ h^n)$, where h^0 is the identity on A^*.

A morphism $h : A^* \to B^*$ is said to be *nonerasing* if $h(a) \neq \lambda$ for all $a \in A$. For any string $w \in C^*$, where $C \subseteq A$ and $|w|_a \geq 1$ for every $a \in C$, the morphism $h : A^* \to B^*$ is called a *renaming (of w)* if $h : C^* \to B^*$ is injective and $|h(a)| = 1$ for every $a \in C$.

Let Σ be a (finite or infinite) alphabet of so-called *terminal symbols* (or: *letters*) and X an infinite set of *variables* with $\Sigma \cap X = \emptyset$. We normally assume $\{\mathtt{a}, \mathtt{b}, \ldots\} \subseteq \Sigma$ and $\{y, z, x_0, x_1, x_2 \ldots\} \subseteq X$. A *pattern* is a string over $\Sigma \cup X$, a *terminal-free pattern* is a string over X and a *word* is a string over Σ. The set of all patterns over $\Sigma \cup X$ is denoted by Pat_Σ. For any pattern α, we refer to the set of variables in α as $\mathrm{var}(\alpha)$.

A morphism $\sigma : (\Sigma \cup X)^* \to (\Sigma \cup X)^*$ is called *terminal-preserving* if $\sigma(a) = a$ for every $a \in \Sigma$. A terminal-preserving morphism $\sigma : (\Sigma \cup X)^* \to \Sigma^*$ is called a *substitution*. Let $S \subseteq \Sigma^*$; then we say that a pattern α is *consistent with* S if, for every $w \in S$, there exists a substitution σ satisfying $\sigma(\alpha) = w$.

Intuitively, the pattern language of a pattern α is the maximum set of words α is consistent with. Formally, we consider two types of pattern languages, depending on whether we restrict ourselves to nonerasing substitutions: the *NE-pattern language* $L_{\mathrm{NE},\Sigma}(\alpha)$ of a pattern $\alpha \in \mathrm{Pat}_\Sigma$ is given by

$$L_{\mathrm{NE},\Sigma}(\alpha) := \{\sigma(\alpha) \mid \sigma : (\Sigma \cup X)^* \to \Sigma^* \text{ is a nonerasing substitution}\},$$

and the *E-pattern language* $L_{\mathrm{E},\Sigma}(\alpha)$ of α is given by

$$L_{\mathrm{E},\Sigma}(\alpha) := \{\sigma(\alpha) \mid \sigma : (\Sigma \cup X)^* \to \Sigma^* \text{ is a substitution}\}.$$

The term *pattern language* refers to any of the definitions introduced above. We call a pattern language *terminal-free* if it is generated by a terminal-free pattern.

We now can introduce our terminology on the main topic of this paper, namely the descriptiveness of a pattern. For any alphabet Σ and any language $S \subseteq \Sigma^*$, a pattern $\delta \in \mathrm{Pat}_\Sigma$ is said to be *NE-descriptive (of S)* provided that $L_{\mathrm{NE},\Sigma}(\delta) \supseteq S$ and, for every $\alpha \in \mathrm{Pat}_\Sigma$ with $L_{\mathrm{NE},\Sigma}(\alpha) \supseteq S$, $L_{\mathrm{NE},\Sigma}(\alpha) \not\subset L_{\mathrm{NE},\Sigma}(\delta)$. Analogously, δ is called *E-descriptive (of S)* if $L_{\mathrm{E},\Sigma}(\delta) \supseteq S$ and, for every $\alpha \in \mathrm{Pat}_\Sigma$ with $L_{\mathrm{E},\Sigma}(\alpha) \supseteq S$, $L_{\mathrm{E},\Sigma}(\alpha) \not\subset L_{\mathrm{E},\Sigma}(\delta)$.

Obviously, the definition of a descriptive pattern is based on the inclusion of pattern languages, which is an undecidable problem for both the full class of NE-pattern languages and the full class of E-pattern languages (cf. Jiang et al. [6], Freydenberger and Reidenbach [4]). A significant part of our subsequent technical considerations, however, can be restricted to terminal-free E-pattern languages, and here the inclusion problem is known to be decidable. This directly results from the following characterisation:

Theorem 1 (Jiang et al. [6]). *Let Σ be an alphabet, $|\Sigma| \geq 2$, and let $\alpha, \beta \in X^+$ be terminal-free patterns. Then $L_{\mathrm{E},\Sigma}(\alpha) \subseteq L_{\mathrm{E},\Sigma}(\beta)$ if and only if there exists a morphism $h : X^* \to X^*$ satisfying $h(\beta) = \alpha$.*

While Theorem 1 is a powerful tool when dealing with the inclusion of terminal-free E-pattern languages, the examination of the descriptiveness of a pattern requires insights into *proper* inclusion relations, and therefore we use some further combinatorial results on morphisms in free monoids to give a more convenient criterion that can replace the use of Theorem 1.

In accordance with Reidenbach and Schneider [9], we designate a terminal-free pattern $\alpha \in X^+$ as *morphically imprimitive* if there is a pattern $\beta \in X^*$ satisfying the following conditions: $|\beta| < |\alpha|$ and there are morphisms $g, h : X^* \to X^*$ such that $g(\alpha) = \beta$ and $h(\beta) = \alpha$. Otherwise, α is *morphically primitive.* Let $\alpha \in X^+$ be morphically primitive. A morphism $h : X^* \to X^*$ is said to be an *imprimitivity morphism (for α)* provided that $|h(\alpha)| > |\alpha|$ and there is a morphism $g : X^* \rightarrow X^*$ satisfying $g(h(\alpha)) = \alpha$. Referring to these concepts, we now can give a characterisation of certain proper inclusion relations between terminal-free E-pattern languages:

Lemma 1. *Let Σ be an alphabet, $|\Sigma| \geq 2$, $\alpha \in X^+$ a morphically primitive pattern and $h : X^* \rightarrow X^*$ a morphism. Then $L_{\mathrm{E},\Sigma}(h(\alpha)) \subset L_{\mathrm{E},\Sigma}(\alpha)$ if and only if h is neither an imprimitivity morphism for, nor a renaming of α.*

The proof for Lemma 1 is omitted due to space constraints.

The question of whether a given morphism is an imprimitivity morphism for a pattern can be easily answered using the following insight:

Theorem 2 (Reidenbach, Schneider [9]). *Let $\alpha \in X^+$ be a morphically primitive pattern. Then a morphism $h : X^* \rightarrow X^*$ is an imprimitivity morphism for α if and only if*

1. *for every $x \in \mathrm{var}(\alpha)$, there exists an $x_h \in \mathrm{var}(h(x))$ such that $|h(x)|_{x_h} = 1$ and $|h(y)|_{x_h} = 0$ for every $y \in \mathrm{var}(\alpha) \setminus \{x\}$, and*
2. *there exists an $x \in \mathrm{var}(\alpha)$ with $|h(x)| \geq 2$.*

Evidently, Lemma 1 can only be applied if there is a tool for checking whether a terminal-free pattern is morphically primitive. This is provided by the following characterisation:

Theorem 3 (Reidenbach, Schneider [9]). *A pattern $\alpha \in X^+$ is morphically primitive if and only if there is no factorisation $\alpha = \beta_0 \gamma_1 \beta_1 \gamma_2 \beta_2 \ldots \beta_{n-1} \gamma_n \beta_n$ with $n \geq 1$, $\beta_k \in X^*$ and $\gamma_k \in X^+$, $k \leq n$, such that*

1. *$|\gamma_k| \geq 2$ for every k, $1 \leq k \leq n$,*
2. *$\mathrm{var}(\beta_0 \ldots \beta_n) \cap \mathrm{var}(\gamma_1 \ldots \gamma_n) = \emptyset$ and*
3. *for every k, $1 \leq k \leq n$, there exists an $x_k \in \mathrm{var}(\gamma_k)$ such that $|\gamma_k|_{x_k} = 1$ and, for every k', $1 \leq k' \leq n$, if $x_k \in \mathrm{var}(\gamma_{k'})$ then $\gamma_k = \gamma_{k'}$.*

Thus, with Lemma 1, Theorem 2 and Theorem 3 we now have an appropriate tool for deciding on particular proper inclusion relations between terminal-free E-pattern languages.

3 Descriptive Patterns and Infinite Strictly Decreasing Chains of Pattern Languages

Before we state and prove the main results of our paper, we discuss some simple yet enlightening observations that establish a connection between descriptiveness of patterns and infinite strictly decreasing chains of pattern languages over some fixed alphabet, i. e. sequences $(L_i)_{i \in \mathbb{N}}$ of pattern languages satisfying, for every $j \in \mathbb{N}$, $L_j \supset L_{j+1}$. This aspect is already briefly mentioned by Jiang et al. [5].

Since, by definition, a descriptive pattern generates a smallest pattern language comprising a language S, S does not have a descriptive pattern if and only if no pattern language L satisfying $L \supseteq S$ is smallest. Hence, the existence of a descriptive pattern essentially depends on the existence of a pattern language that is not contained in an infinite strictly decreasing chain:

Observation 1. *Let Σ be an alphabet and $S \subseteq \Sigma^*$ a language. Then there is no pattern that is NE-descriptive (or E-descriptive) of S if and only if, for every pattern α with $L_{\mathrm{NE},\Sigma}(\alpha) \supseteq S$ (or $L_{\mathrm{E},\Sigma}(\alpha) \supseteq S$, respectively) there is*

- *a sequence of patterns $\alpha_i \in \mathrm{Pat}_\Sigma$, $i \in \mathbb{N}$, satisfying, for every $j \in \mathbb{N}$, $L_{\mathrm{NE},\Sigma}(\alpha_j) \supset L_{\mathrm{NE},\Sigma}(\alpha_{j+1})$ (or $L_{\mathrm{E},\Sigma}(\alpha_j) \supset L_{\mathrm{E},\Sigma}(\alpha_{j+1})$, respectively) and $L_{\mathrm{NE},\Sigma}(\alpha_j) \supseteq S$ (or $L_{\mathrm{E},\Sigma}(\alpha_j) \supseteq S$, respectively), and*
- *an $n \in \mathbb{N}$ with $L_{\mathrm{NE},\Sigma}(\alpha_n) = L_{\mathrm{NE},\Sigma}(\alpha)$ (or $L_{\mathrm{E},\Sigma}(\alpha_n) = L_{\mathrm{E},\Sigma}(\alpha)$, respectively).*

Proof. Directly from the definition of an NE-descriptive (or E-descriptive) pattern. □

Thus, the question of whether there is a descriptive pattern for a language S requires insights into the inclusion problem for pattern languages. As partly stated in Section 2, this problem is undecidable in the general case, but it is decidable for the class of terminal-free E-pattern languages (though combinatorially complex and, according to Ehrenfeucht and Rozenberg [3], NP-complete).

In order to illustrate and substantiate Observation 1 and as a reference for further considerations in Section 4, we now give some examples of strictly decreasing chains of pattern languages. We begin with a sequence of patterns that has identical properties for both NE- and E-pattern languages:

Example 1. Let Σ be any alphabet. For every $i \in \mathbb{N}$, we define $\alpha_i := x_1^{2^i}$, i. e. $\alpha_0 = x_1$, $\alpha_1 = x_1^2$, $\alpha_2 = x_1^4$, $\alpha_3 = x_1^8$ and so on. Evidently, for every $j \in \mathbb{N}$, the morphism $h : \{x_1\}^+ \to \{x_1\}^+$, defined by $h(x_1) := x_1^2$, satisfies $h(\alpha_j) = \alpha_{j+1}$. Since, for both NE- and E-pattern languages, the existence of such a morphism is a sufficient condition for an inclusion relation (cf. Theorems 2.2 and 2.3 by Jiang et al. [5]), $L_{\mathrm{NE},\Sigma}(\alpha_j) \supseteq L_{\mathrm{NE},\Sigma}(\alpha_{j+1})$ and $L_{\mathrm{E},\Sigma}(\alpha_j) \supseteq L_{\mathrm{E},\Sigma}(\alpha_{j+1})$ are satisfied. In the given example, it is evident that all inclusions of NE-pattern languages are strict. The same holds for the inclusion of E-pattern languages; alternatively, for all but unary alphabets Σ, it is directly proved by Lemma 1 (using Theorem 2 and Theorem 3) given in Section 2. Hence, the sequence of α_i leads to an infinite strictly decreasing chain for NE-pattern languages as well as for E-pattern languages. Nevertheless, the sequence of patterns is irrelevant in the context of Observation 1, as the sets $S_{\mathrm{NE}} := \bigcap_{i=0}^{\infty} L_{\mathrm{NE},\Sigma}(\alpha_i)$ and $S_{\mathrm{E}} := \bigcap_{i=0}^{\infty} L_{\mathrm{E},\Sigma}(\alpha_i)$, i. e. those languages all patterns are consistent with, are empty.

Our next example looks quite similar to Example 1, but here a difference between NE- and E-pattern languages can be noted:

Example 2. Let Σ be an alphabet with $|\Sigma| \geq 2$. For every $i \in \mathbb{N}$, we define $\alpha_i := x_1^{2^i} y^2$, i. e. $\alpha_0 = x_1 y^2$, $\alpha_1 = x_1^2 y^2$, $\alpha_2 = x_1^4 y^2$, $\alpha_3 = x_1^8 y^2$ and so on. Referring to the same facts as mentioned in Example 1, it can be shown that the patterns again define one infinite strictly decreasing chain of NE-pattern languages and another one of E-pattern languages. However, while the set $S_{\mathrm{NE}} := \bigcap_{i=0}^{\infty} L_{\mathrm{NE},\Sigma}(\alpha_i)$ again is empty, $S_{\mathrm{E}} := \bigcap_{i=0}^{\infty} L_{\mathrm{E},\Sigma}(\alpha_i)$ now equals $L_{\mathrm{E},\Sigma}(y^2)$. Hence, we have a chain of E-pattern languages that are all consistent with a nonempty language. Nevertheless, $L_{\mathrm{E},\Sigma}(y^2)$ obviously has a descriptive pattern, namely $\delta := y^2$, and this of course holds for all infinite sequences of patterns where S_{E} equals an E-pattern language. Consequently, the existence of a single infinite strictly decreasing chain of E-pattern languages L_i satisfying, for every $i \in \mathbb{N}$, $L_i \supseteq S$, does not mean that there is no E-descriptive pattern for S. Furthermore, it is worth mentioning that we can replace S_{E} with a finite language and still preserve the above described properties of the α_i and δ. For $\Sigma \supseteq \{\mathtt{a}, \mathtt{b}\}$, this is demonstrated, e. g., by the language $S := \{\mathtt{aa}, \mathtt{bb}\}$, which satisfies, for every $i \in \mathbb{N}$, $S \subseteq L_{\mathrm{E},\Sigma}(\alpha_i)$ and has the E-descriptive pattern δ.

Our final example presents a special phenomenon of E-pattern languages, namely the existence bi-infinite strictly decreasing/increasing chains of such languages:

Example 3. Let Σ be an alphabet with $|\Sigma| \geq 2$. For every $i \in \mathbb{Z}$, we define

$$\alpha_i := \begin{cases} x_1^{2^{-i}} & \text{if } i \text{ is negative,} \\ x_1^2 x_2^2 \ \dots \ x_{i+2}^2 & \text{else.} \end{cases}$$

Hence, for example, from $i = -3$ to $i = 2$ the patterns read $\alpha_{-3} = x_1^8$, $\alpha_{-2} = x_1^4$, $\alpha_{-1} = x_1^2$, $\alpha_0 = x_1^2 x_2^2$, $\alpha_1 = x_1^2 x_2^2 x_3^2$, and $\alpha_2 = x_1^2 x_2^2 x_3^2 x_4^2$. Using Theorem 3, it is easy to show that all patterns are morphically primitive. Theorem 2 demonstrates that all morphisms mapping an α_k onto an α_j, $j < k$, are not imprimitivity morphisms. Therefore we can conclude from Lemma 1 that $L_{\mathrm{E},\Sigma}(\alpha_j) \subset L_{\mathrm{E},\Sigma}(\alpha_k)$ if and only if $j < k$. For the given patterns, $S_{\mathrm{E}} := \bigcap_{i=-\infty}^{\infty} L_{\mathrm{E},\Sigma}(\alpha_i)$ is empty, but if we define, for every $i \in \mathbb{Z}$, $\alpha_i' := y^2 \alpha_i$, then these α_i' generate a bi-infinite strictly decreasing/increasing chain of E-pattern languages where $S_{\mathrm{E}} := \bigcap_{i=-\infty}^{\infty} L_{\mathrm{E},\Sigma}(\alpha_i') = L_{\mathrm{E},\Sigma}(y^2)$ is an E-pattern language.

Note that the example patterns given above are terminal-free merely for the sake of convenience. They can be effortlessly turned into certain patterns containing terminal symbols and still showing equivalent properties.

4 The Existence of Descriptive Patterns

In the present chapter we study the existence of patterns that are descriptive of sets S of strings. According to our remarks in Section 1, four main cases can be considered, depending on whether S is finite or infinite and whether NE- or E-descriptiveness is examined. We focus on the existence of E-descriptive patterns for infinite languages since, for the other three cases, answers are absolutely straightforward or directly or indirectly provided by Angluin [1] and Jiang et al. [5]. In order to give a comprehensive description and further explain some of our formal concepts and statements we nevertheless also briefly describe the known or trivial cases.

Using Observation 1, the question of the existence of *NE-descriptive* patterns can be easily answered for all types of languages S. We begin with the case of a *finite* S. Here, it is primarily necessary to observe that a word w can only be covered by a pattern α through nonerasing substitutions if α is not longer than w. Hence, for any finite alphabet Σ and any word over Σ, there are only finitely many NE-pattern languages over Σ covering this word; this property of a class of languages is commonly referred to as *finite thickness* (cf. Wright [13]). Quite obviously, the same holds for infinite alphabets Σ, since the number of different terminal symbols that can occur in patterns covering w is limited by the number of different terminal symbols in w. With regard to infinite sequences of patterns (generating languages that all differ from each other) over a fixed alphabet, this means that none of them can contain infinitely many patterns that cover, e. g., the shortest word in a given finite set of strings. This immediately shows that, for every finite S, there exists an NE-descriptive pattern:

Proposition 1 (Angluin [1]). *Let Σ be an alphabet and $S \subseteq \Sigma^+$ a finite language. Then there is a pattern $\delta \in \mathrm{Pat}_\Sigma$ that is NE-descriptive of S.*

Note that Angluin [1] does not explicitly state Proposition 1, but directly studies more challenging questions by introducing a procedure computing an NE-descriptive pattern for any finite language S and examining the computational complexity of the problem of finding such patterns for finite languages.

With regard to NE-descriptive patterns for *infinite* languages S, the same reasoning as for finite languages S leads to the analogous result:

Proposition 2. *Let Σ be an alphabet and $S \subseteq \Sigma^+$ an infinite language. Then there is a pattern $\delta \in \mathrm{Pat}_\Sigma$ that is NE-descriptive of S.*

Proof. Directly from Observation 1 and the finite thickness of the class of NE-pattern languages. □

A closer look at the underlying reasoning proving Propositions 1 and 2 reveals that it does not need to consider whether any infinite sequence of patterns leads to an infinite strictly decreasing chain of NE-pattern languages (as featured by Observation 1), but can be completely based on the concept of finite thickness. If we nevertheless wish to examine the properties of such chains, then we can easily observe that, for every sequence of patterns α_i, $i \in \mathbb{N}$, with $L_{\mathrm{NE},\Sigma}(\alpha_i) \supset L_{\mathrm{NE},\Sigma}(\alpha_{i+1})$, the set $S_{\mathrm{NE}} := \bigcap_{i=0}^{\infty} L_{\mathrm{NE},\Sigma}(\alpha_i)$ necessarily is empty. Hence, Examples 1 and 2 illustrate the only option possible.

With regard to *E-descriptiveness*, the situation is more complex. As shown by Examples 2 and 3, the class of E-pattern languages does not have finite thickness and there are even finite and infinite languages that are contained in all E-pattern languages of an infinite strictly decreasing chain. Nevertheless, it is known that every *finite* language has an E-descriptive pattern:

Theorem 4 (Jiang et al. [5]). *Let Σ be an alphabet and $S \subseteq \Sigma^*$ a finite language. Then there is a pattern $\delta \in \mathrm{Pat}_\Sigma$ that is E-descriptive of S.*

The proof for Theorem 4 given by Jiang et al. [5] demonstrates that for every finite language S an upper bound n can be given such that, for every pattern α consistent with S, there exists a pattern β satisfying $|\beta| \leq n$ and $L_{\mathrm{E},\Sigma}(\beta) \subseteq L_{\mathrm{E},\Sigma}(\alpha)$. So if, for any finite S, there is a sequence of patterns α_i, $i \in \mathbb{N}$, leading to an infinite strictly decreasing chain of E-pattern languages comprising S – which implies that there is no upper bound for the length of the α_i – then all but finitely many of these patterns need to have variables that are not required for generating the words in S. This phenomenon is illustrated by Example 2, where only the subpattern y^2 of all patterns is necessary in order to map the patterns onto the words in S_{E}.

In the proof for Theorem 4, the upper bound n equals the sum of the lengths of the words in S. Thus, this method cannot be adopted when investigating the existence of E-descriptive patterns for *infinite* sets of words. In fact, as to be demonstrated below, we here need to consider two subcases depending on the number of different letters occurring in the words of S. If the underlying alphabet is unary, then the descriptiveness of a pattern is related to the inclusion relation of E-pattern languages over this unary alphabet. The structure of such E-pattern languages, however, is significantly simpler than that of E-pattern languages over larger alphabets; in particular, the full class of these languages is a specific subclass of the regular languages (namely the linear unary languages). Therefore, and just as in the previous cases, it can be shown that, for every sequence of patterns $(\alpha_i)_{i \in \mathbb{N}}$ leading to a infinite strictly decreasing chain of E-pattern languages over a unary alphabet, the language $S_{\mathrm{E}} := \bigcap_{i=0}^{\infty} L_{\mathrm{E},\Sigma}(\alpha_i)$ is empty. Referring to Observation 1, this directly leads to the following result:

Theorem 5. *Let Σ be an alphabet, $|\Sigma| = 1$, and $S \subseteq \Sigma^*$ an infinite language. Then there is a pattern $\delta \in \mathrm{Pat}_\Sigma$ that is E-descriptive of S.*

The proof for Theorem 5 is omitted due to space constraints.

In contrast to this, Example 2 demonstrates that, for alphabets with at least two letters, there is an infinite strictly decreasing chain of E-pattern languages such that the intersection of all these languages is nonempty. Since this intersection is an E-pattern language, Example 2 can nevertheless not be used to establish a result that differs from those given for the other cases. In order to answer the question of whether this holds true for all such chains, we now consider a more sophisticated infinite sequence of patterns, that is defined as follows:

Definition 1. *We define the pattern $\alpha_0 := y^2 z^2$ and the morphism $\phi : X^* \to X^*$ (note that we assume $X \supseteq \{y, z, x_0, x_1, x_2 \ldots\}$) through, for every $i \in \mathbb{N}$,*

$$\phi(x_i) := x_{i+1},\ \phi(y) := y^2 x_1,\ \phi(z) := x_1 z^2.$$

Then, for every $i \in \mathbb{N}$, the pattern α_{i+1} is given by $\alpha_{i+1} := \phi(\alpha_i) = \phi^i(\alpha_0)$.

This means that, for example,

$$\begin{aligned}
\alpha_1 &= y^2x_1\, y^2x_1\, x_1z^2\, x_1z^2,\\
\alpha_2 &= (y^2x_1y^2x_1x_2)\,(y^2x_1y^2x_1x_2)\,(x_2x_1\, z^2x_1z^2)\,(x_2x_1z^2x_1z^2),\\
\alpha_3 &= (y^2x_1y^2x_1x_2\, y^2x_1y^2x_1x_2\, x_3)\,(y^2x_1y^2x_1x_2\, y^2x_1y^2x_1x_2\, x_3)\\
&\quad (x_3\, x_2x_1z^2x_1z^2\, x_2x_1z^2x_1z^2)\,(x_3\, x_2x_1z^2x_1z^2\, x_2x_1z^2x_1z^2).
\end{aligned}$$

It can be shown that this sequence $(\alpha_i)_{i\in\mathbb{N}}$ defines an infinite strictly decreasing chain of E-pattern languages. Furthermore, if we define the morphism $\psi : X^* \to X^*$ through $\psi(x_i) := x_i$ and $\psi(y) := \psi(z) := x_0$, then, for every alphabet Σ with $|\Sigma| \geq 2$, $L_\Sigma := \bigcup_{i=0}^{\infty} L_{\mathrm{E},\Sigma}(\psi(\alpha_i))$ satisfies $L_\Sigma \subseteq \bigcap_{i=0}^{\infty} L_{\mathrm{E},\Sigma}(\alpha_i)$. Finally, it can be demonstrated that the sequence $(\alpha_i)_{i\in\mathbb{N}}$ has a very particular property, since for every pattern γ with $L_{\mathrm{E},\Sigma}(\gamma) \supseteq L_\Sigma$ there exists an α_i satisfying $L_{\mathrm{E},\Sigma}(\gamma) \supseteq L_{\mathrm{E},\Sigma}(\alpha_i)$. Referring to Observation 1, this implies the main result of our paper:

Theorem 6. *For every alphabet Σ with $|\Sigma| \geq 2$ there is an infinite language $L_\Sigma \subset \Sigma^*$ that has no E-descriptive pattern.*

The proof for Theorem 6 is omitted due to space constraints.

Consequently, when searching for descriptive patterns, the case of E-descriptive patterns of infinite languages over alphabets of at least two letters is the only one where the existence of such patterns is not always guaranteed. This directly answers a question posed by Jiang et al. [5].

Finally, it can be shown that, while the proof of Theorem 6 is based on the particular shape of the infinite union L_Σ of E-pattern languages described above, L_Σ can be replaced by a language L^t_Σ which, for every pattern $\psi(\alpha_i)$, $i \geq 0$, contains just a single word. In order to describe this insight more precisely, we have to introduce the following concept:

Definition 2. *A language L is called* properly thin *if, for every $n \geq 0$, L contains at most one word of length n.*

Referring to this definition, we can strengthen Theorem 6 as follows:

Corollary 1. *For every alphabet Σ with $|\Sigma| \geq 2$, there is an infinite properly thin language $L^t_\Sigma \subset \Sigma^*$ that has no E-descriptive pattern.*

The proof for Corollary 1 is omitted due to space constraints.

5 Conclusions and Further Directions of Research

In the present paper, we have studied the existence and nonexistence of patterns that are descriptive of a set of strings. We have explained that this question is related to the existence of infinite strictly decreasing chains of pattern languages. Our main result has demonstrated that there exist infinite languages over alphabets of at least two letters that do not have an E-descriptive pattern.

This insight leads to the question of characteristic criteria describing infinite languages without an E-descriptive pattern. We have referred to one example of such languages, namely a particular infinite union of E-pattern languages. Although we have mentioned that an infinite properly thin language can be substituted for this union, we anticipate that only very special languages (and very special infinite strictly decreasing chains of E-pattern languages) can be used for the proof of our main result. Thus, we expect the nonexistence of E-descriptive patterns to be a rare phenomenon. In addition to the said criteria, we consider it worthwhile to further investigate the existence of efficient procedures finding descriptive patterns of given languages (for those cases where descriptive patterns exist). So far, this question has only been answered for NE-descriptive patterns of finite languages (see Angluin [1]), demonstrating that no such procedure can have polynomial runtime (provided that P≠NP). We feel that a more pleasant result might be possible for E-descriptive patterns.

References

1. D. Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.
2. A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert. Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5:279–305, 1998.
3. A. Ehrenfeucht and G. Rozenberg. Finding a homomorphism between two words is NP-complete. *Information Processing Letters*, 9:86–88, 1979.
4. D.D. Freydenberger and D. Reidenbach. Bad news on decision problems for patterns. In *Proc. 12th International Conference on Developments in Language Theory, DLT 2008*, volume 5257 of *Lecture Notes in Computer Science*, 2008.
5. 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.
6. T. Jiang, A. Salomaa, K. Salomaa, and S. Yu. Decision problems for patterns. *Journal of Computer and System Sciences*, 50:53–63, 1995.
7. A. Mateescu and A. Salomaa. Patterns. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1, chapter 4.6, pages 230–242. Springer, 1997.
8. 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.
9. D. Reidenbach and J.C. Schneider. Morphically primitive words. *Theoretical Computer Science*, 410:2148–2161, 2009.
10. G. Rozenberg and A. Salomaa. *Handbook of Formal Languages*, volume 1. Springer, Berlin, 1997.
11. K. Salomaa. Patterns. In C. Martin-Vide, V. Mitrana, and G. Păun, editors, *Formal Languages and Applications*, number 148 in Studies in Fuzziness and Soft Computing, pages 367–379. Springer, 2004.
12. T. Shinohara. Polynomial time inference of extended regular pattern languages. In *Proc. RIMS Symposia on Software Science and Engineering, Kyoto*, volume 147 of *Lecture Notes in Computer Science*, pages 115–127, 1982.
13. K. Wright. Identification of unions of languages drawn from an identifiable class. In *Proc. 2nd Annual Workshop on Computational Learning Theory, COLT 1989*, pages 328–333, 1989.