Journal of Computer and System Sciences ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Closure properties of pattern languages ☆

Joel D. Day [a], Daniel Reidenbach [a,*], Markus L. Schmid [b]

[a] Department of Computer Science, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK
[b] Fachbereich 4 – Abteilung Informatik, Universität Trier, D-54286 Trier, Germany

ARTICLE INFO

Article history:
Received 10 March 2015
Received in revised form 26 May 2016
Accepted 7 July 2016
Available online xxxx

Keywords:
Pattern languages
Closure properties

ABSTRACT

Pattern languages are a well-established class of languages, but very little is known about their closure properties. In the present paper we establish a large number of closure properties of the terminal-free pattern languages, and we characterise when the union of two terminal-free pattern languages is again a terminal-free pattern language. We demonstrate that the equivalent question for general pattern languages is characterised differently, and that it is linked to some of the most prominent open problems for pattern languages. We also provide fundamental insights into a well-known construction of E-pattern languages as unions of NE-pattern languages, and vice versa.

1. Introduction

Pattern languages were introduced by Dana Angluin [1] in order to model the algorithmic inferrability of patterns that are common to a set of words. In this context, a pattern is a sequence of variables and terminal symbols, and its language is the set of all words that can be generated from the pattern by a substitution that replaces all variables in the pattern by words of terminal symbols. Hence, more formally, a substitution is a terminal-preserving morphism, i.e., a morphism that maps every terminal symbol to itself. For example, the pattern language of the pattern $\alpha := x_1x_1\mathtt{a}x_2\mathtt{b}$, where x_1, x_2 are variables and $\mathtt{a}, \mathtt{b}$ are terminal symbols, is the set of all words that have a square as a prefix, followed by an arbitrary suffix that begins with the letter $\mathtt{a}$ and ends with the letter $\mathtt{b}$. Thus, e.g., $\mathtt{abbabbaab}$ is contained in the language of α, whereas $\mathtt{bbbaa}$ is not. It is a direct consequence of these definitions that a pattern language is either a singleton or infinite. Furthermore, it is worth noting that two basic types of pattern languages are considered in the literature, depending on whether the variables must stand for nonempty words (referred to as nonerasing or NE-pattern languages) or whether they may represent the empty word (so-called extended, erasing or simply E-pattern languages).

While the definition of pattern languages is simple, many of their properties are known to be related to complex phenomena in combinatorics on words, such as pattern avoidability (see Jiang et al. [9]) and ambiguity of morphisms (see Reidenbach [17]). Hence, the knowledge on pattern languages is still patchy, despite recent progress mainly regarding decision problems (see, e.g., Freydenberger, Reidenbach [7], Fernau, Schmid [5], Fernau et al. [6] and Reidenbach, Schmid [18]) and the relation to the Chomsky hierarchy (see Jain et al. [8] and Reidenbach, Schmid [19]).

Establishing the closure properties of a class of formal languages is one of the most classical and fundamental research tasks in formal language theory and any respective progress normally leads to insights and techniques that yield a better understanding of the class. In the case of pattern languages, it is known since Angluin's initial work that they are not closed

☆ A preliminary version [4] of this paper was presented at the conference DLT 2014.

* Corresponding author.
E-mail addresses: J.Day@lboro.ac.uk (J.D. Day), D.Reidenbach@lboro.ac.uk (D. Reidenbach), MSchmid@uni-trier.de (M.L. Schmid).

http://dx.doi.org/10.1016/j.jcss.2016.07.003
0022-0000/

under most of the usual operations, including union, intersection and complement. However, these non-closure properties can be shown by using very basic example patterns and exploiting peculiarities of the definition of pattern languages. For example, if a pattern does not contain a variable, then its language is a singleton; hence the union of any two distinct singleton pattern languages contains two elements, and therefore it cannot be a pattern language. Furthermore, the intersection of two pattern languages given by patterns that start with different terminal symbols is empty and the empty set, although a trivial language, is not a pattern language as well. Since, apart from a strong result by Shinohara [20] on the union of NE-pattern languages, hardly anything is known beyond such immediate facts, we can observe that in the case of pattern languages the existing closure properties fail to contribute to our understanding of their intrinsic properties.

It is the main purpose of this paper to investigate the closure properties of pattern languages more thoroughly. To this end, in Section 3, we consider the closure properties of two important subclasses of pattern languages, namely the classes of terminal-free NE- and E-pattern languages, i.e., pattern languages that are generated by patterns that do not contain any terminal symbols. This choice is motivated by the fact that terminal-free patterns have been a recent focus of interest in the research on pattern languages and, furthermore, most existing examples for non-closure of pattern languages (including the two examples for union and intersection given in the previous paragraph) do not translate to the terminal-free case. In Section 3.1, we completely characterise when the union of two terminal-free pattern languages is again a terminal-free pattern language and, in Section 3.2, we prove their non-closure under intersection, for which the situation is much more complicated compared to the operation of union.

We consider general pattern languages in Section 4, and we provide complex examples demonstrating that it is probably a very hard task to obtain full characterisations of those pairs of pattern languages whose unions or intersections are again a pattern language. In Section 4.3, we also study the question whether an E-pattern language can be expressed by the union of *nonerasing* pattern languages and, likewise, whether an NE-pattern language can be expressed by the union of *erasing* pattern languages. This question is slightly at odds with the classical investigation of closure properties, since we apply a language operation to members of one class and ask whether the resulting language is a member of another class. However, in the case of pattern languages, this makes sense, since every NE-pattern language is a finite union of E-pattern languages and every E-pattern language is a finite union of NE-pattern languages (see Jiang et al. [9]), a phenomenon that has been widely utilised in the context of inductive inference of pattern languages (see, e.g., Wright [22], Shinohara, Arimura [21]).

2. Definitions and preliminary results

The symbols $\cup$, $\cap$ and $\setminus$ denote the set operations of *union*, *intersection* and *set difference*, respectively. For sets U and B with $B \subseteq U$, $\overline{B} := U \setminus B$ is the *complement* of B.

Let $\mathbb{N} := \{1, 2, 3, \ldots\}$ and let $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$. For an arbitrary alphabet A, a *word* (*over* A) is a finite sequence of symbols from A, and ε stands for the *empty word*. The notation A^+ denotes the set of all nonempty words over A, and $A^* := A^+ \cup \{\varepsilon\}$. For the *concatenation* of two words w_1, w_2 we write $w_1 \cdot w_2$ or simply $w_1 w_2$, and w^n stands for the n-fold concatenation of the word w. We say that a word $v \in A^*$ is a *factor* of a word $w \in A^*$ 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 word, then v is a *prefix* (or a *suffix*, respectively) of w. If $w = w_0 v_1 w_1 v_2 \cdots v_n w_n$ and $v = v_1 v_2 \cdots v_n$, for some $w_0, w_n \in A^*$, $w_i, v_j \in A^+$, $1 \leq i \leq n-1$, $1 \leq j \leq n$, then v is a *subsequence* of w. The notation $|K|$ stands for the size of a set K or the length of a word K. For a $w \in A^*$ and $a \in A$, $|w|_a$ denotes the number of occurrences of the symbol a in w. A word w is *primitive* if, for any u such that $w = u^k$, $k = 1$. The *primitive root* of a word w is the primitive word u such that $w = u^k$, $k \in \mathbb{N}$.

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, for every $a \in A$, $h(a) \neq \varepsilon$. A morphism h is *ambiguous (with respect to a word w)* if there exists a morphism g satisfying $g(w) = h(w)$ and, for a letter a in w, $g(a) \neq h(a)$. If such a morphism g does not exist, then h is called *unambiguous (with respect to w)*. A morphism $\sigma : A^* \to B^*$ is *periodic* if for some (primitive) word $w \in B^*$, $\sigma(x) \in \{w\}^*$ for every $x \in A$. The word w will be referred to as the *primitive root* of σ. If $|\sigma(x)| = 1$ for every $x \in A$, then σ is *1-uniform*.

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 word over $\Sigma \cup X$, a *terminal-free pattern* is a nonempty word over X; if a word contains symbols from Σ only, then we occasionally call it a *terminal word*. For any pattern α, we refer to the set of variables in α as $\mathrm{var}(\alpha)$. If the variables in a pattern α are labelled in the natural way, then it is said to be in *canonical form*, i.e., α is in canonical form if, for some $n \in \mathbb{N}$, $\mathrm{var}(\alpha) = \{x_1, x_2, \ldots, x_n\}$ and, for any $x_i, x_j \in \mathrm{var}(\alpha)$ with $i < j$, there is a prefix β of α such that $x_i \in \mathrm{var}(\beta)$ and $x_j \notin \mathrm{var}(\beta)$. A pattern α is a *one-variable* pattern if $|\mathrm{var}(\alpha)| = 1$. A pattern α is *periodicity forcing* if for any alphabet Σ and morphisms $g, h : \mathrm{var}(\alpha)^* \to \Sigma^*$, $g(\alpha) = h(\alpha)$ implies g and h are periodic or $g = h$. A morphism $h : (\Sigma \cup X)^* \to (\Sigma \cup X)^*$ is *terminal-preserving* if $h(a) = a$ for every $a \in \Sigma$. The *residual* of a pattern α is the word $h_\varepsilon(\alpha)$, where $h_\varepsilon : (\Sigma \cup X)^* \to (\Sigma \cup X)^*$ is a terminal preserving morphism with $h_\varepsilon(x) := \varepsilon$ for every $x \in \mathrm{var}(\alpha)$. A terminal-preserving morphism $h : (\Sigma \cup X)^* \to \Sigma^*$ is called a *substitution*.

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

Note that we call a pattern language terminal-free if there exists a terminal-free pattern that generates it.

Please cite this article in press as: J.D. Day et al., Closure properties of pattern languages, J. Comput. Syst. Sci. (2016), http://dx.doi.org/10.1016/j.jcss.2016.07.003