Regular Patterns, Regular Languages and Context-Free Languages

Sanjay Jain*,[1] and Yuh Shin Ong[1] and Frank Stephan**,[1,2]

[1] Department of Computer Science, National University of Singapore, Singapore 117417, Republic of Singapore.
sanjay@comp.nus.edu.sg and yuhshin@gmail.com

[2] Department of Mathematics, National University of Singapore, Singapore 119076, Republic of Singapore.
fstephan@comp.nus.edu.sg

Abstract. In this paper we consider two questions. First we consider whether every pattern language which is regular can be generated by a regular pattern. We show that this is indeed the case for extended (erasing) pattern languages if alphabet size is at least four. In all other cases, we show that there are patterns generating a regular language which cannot be generated by a regular pattern. Next we consider whether there are pattern languages which are context-free but not regular. We show that, for alphabet size 2 and 3, there are both erasing and non-erasing pattern languages which are context-free but not regular. On the other hand, for alphabet size at least 4, every erasing pattern language which is context-free is also regular. It is open at present whether there exist non-erasing pattern languages which are context-free but not regular for alphabet size at least 4.

1 Introduction

Angluin [1] introduced the concept of a pattern language as an example for an interesting learnable class. A pattern is a finite string over $\Sigma \cup V$, where Σ is a finite alphabet and V is a set of variables. The language generated by the pattern is the set of strings which can be obtained by replacing each variable in the pattern by a string over Σ. If the strings are permitted to be empty, then the corresponding languages are called extended or erasing pattern languages [11]; otherwise, the corresponding languages are called non-erasing pattern languages. Angluin [1] showed that the class of non-erasing pattern languages are learnable from positive data in the limit, while the problem whether the class of erasing pattern languages is learnable remained open for many years until it was eventually resolved negatively by Reidenbach [8]. Indeed, pattern languages became a well-studied topic and various decidability questions related to them were only settled after a long time [3].

* Supported in part by NUS grant numbers R252-000-420-112 and C252-000-087-001.
** Supported in part by NUS grant numbers R146-000-114-112 and R252-000-420-112.

Angluin's algorithm for learning non-erasing pattern languages had certain weaknesses which were later improved by Lange and Wiehagen [6]. Zeugmann [13] investigated the learning properties of Lange and Wiehagen's algorithm from a statistical perspective. Other investigations on various approaches to learn pattern languages followed [2, 5, 10, 12]. It is NP-complete to test whether a pattern generates a certain string; this hinders feasible consistent learning algorithms with respect to various paradigms.

For getting better learnability properties, the learnability of subclasses of the pattern languages was studied. For example, patterns generated by only a few variables were considered [2, 5]. Another approach is due to Shinohara [11]: he introduced the concept of regular patterns and regular pattern languages, where each variable appears at most once in the pattern.

It is easy to see that the language generated by a regular pattern is regular, that is, recognizable by a deterministic finite automaton. For the converse direction, Reidenbach [9] showed that, for non-erasing pattern languages, for any non-empty alphabet, there are pattern languages which are regular but which are not generated by any regular pattern. In this paper we study this question for erasing pattern languages. We show that if alphabet Σ contains at least four letters, then every erasing pattern language which is also regular can be generated by a regular pattern. On the other hand, if alphabet Σ is non-empty and has at most three elements, then there are erasing pattern languages which are also regular, but are not generated by any regular pattern.

Reidenbach [9] asked whether there are pattern languages which generate context-free languages which are not regular. Note that for alphabet size 1, every erasing or non-erasing pattern language is regular. We show that, for alphabet size 2 or 3, for both erasing as well as non-erasing case, there exist pattern languages which are context-free but not regular. On the other hand, for alphabet size at least 4, we show that every erasing pattern language which is context-free is also regular. The corresponding question for non-erasing pattern languages for alphabet size at least 4 is open at this point.

2 Preliminaries

For a string x, we let $|x|$ denote the length of x, and $x(i)$ denote the $(i+1)$-th character in the string. Thus $x = x(0)x(1)\dots x(|x|-1)$. Symbol ϵ denotes the empty string.

Throughout the paper we let Σ be a finite and non-empty alphabet. Without loss of generality, we assume that $\Sigma = \{0, 1, \dots, n\}$ for some natural number n. Furthermore, we let V be an infinite set of variables. We often use $V = \{x_1, x_2, \dots\}$ and we might also just use x, y, z for the variables if we need only two or three of them. A *pattern* [1] is a member of $(\Sigma \cup V)^*$.

Informally, a substitution is a mapping from $(\Sigma \cup V)^*$ to Σ^* which replaces consistently every variable by a string in Σ^*. More formally, a substitution s is any mapping which is defined as follows.

1. Start with a mapping s from V to Σ^*.
2. Let $s(a) = a$ for all $a \in \Sigma$.

3. Extend s inductively to a mapping from strings over $\Sigma \cup V$ to strings over Σ by taking $s(\epsilon) = \epsilon$, $s(wa) = s(w)a$ and $s(wx) = s(w)s(x)$, for all $w \in (\Sigma \cup V)^*$, $a \in \Sigma$ and $x \in V$.

A substitution s is called non-erasing if $s(x) \neq \epsilon$ for all $x \in V$, that is, if $s(w) \neq \epsilon$ for all $w \neq \epsilon$.

The definition of the language generated by a pattern depends on whether we are considering *erasing* or *non-erasing* substitutions. For a pattern π, $L_e(\pi) = \{s(\pi) : s \text{ is a substitution}\}$ and $L_{ne}(\pi) = \{s(\pi) : s \text{ is a non-erasing substitution}\}$. Angluin [1] only considered non-erasing pattern languages. Shinohara [11] first considered the concept of extended or erasing pattern languages.

A *regular pattern* [11] is a pattern in which variables do not repeat. That is, for all $i < |\pi|$, if $\pi(i) \in V$, then for all $j < |\pi|$ such that $i \neq j$, we must have $\pi(i) \neq \pi(j)$. For a regular pattern π, the language $L_e(\pi)$ and $L_{ne}(\pi)$ are respectively called regular pattern language and non-erasing regular pattern language.

3 Regular Languages Generated only by Nonregular Patterns

Clearly, a (non-erasing) regular pattern language is also a regular language. In the following, we will consider whether every pattern language which is a regular language can be generated by a regular pattern. For non-erasing pattern languages, for any alphabet, this is not possible in general [9]. For erasing pattern languages, this depends on the size of the alphabet.

We first consider alphabet size 1; that is, we consider the case that $\Sigma = \{0\}$. In this case, every pattern language (erasing or non-erasing) is regular [9]. For example, $L_e(00xx)$ is the regular language $00(00)^*$. However, $00(00)^*$ is not generated by any regular pattern, as the language generated by any regular pattern is either a singleton or a cofinite set over the alphabet Σ. The next two results look at the case of alphabet size 2 and 3.

Theorem 1. *Suppose $|\Sigma| = 2$. Then there exists a pattern π such that $L_e(\pi)$ is regular but for no regular pattern π', $L_e(\pi') = L_e(\pi)$.*

Proof. Consider the pattern $\pi = x_1x_2x_31x_2x_4x_4x_51x_6x_5x_7$ and the alphabet $\Sigma = \{0, 1\}$. We first claim that $L_e(\pi)$ is regular. In particular, we claim that the equality

$$L_e(\pi) = \Sigma^*1(00)^*1\Sigma^* \cup \Sigma^*0\Sigma^*10(00)^*1\Sigma^* \cup \Sigma^*10(00)^*1\Sigma^*0\Sigma^*$$

holds.

($\subseteq$): Clearly, $L_e(\pi) \subseteq \Sigma^*1\Sigma^*1\Sigma^*$. Any w which contains at least two 1's, must satisfy: (i) $w \in \Sigma^*1(00)^*1\Sigma^*$ or (ii) $w \in \Sigma^*0\Sigma^*10(00)^*1\Sigma^*$ or (iii) $w \in \Sigma^*10(00)^*1\Sigma^*0\Sigma^*$ or (iv) $w \in 10(00)^*1$. Note that $10(00)^*1 \cap L_e(\pi) = \emptyset$. Thus,

$$L_e(\pi) \subseteq \Sigma^*1(00)^*1\Sigma^* \cup \Sigma^*0\Sigma^*10(00)^*1\Sigma^* \cup \Sigma^*10(00)^*1\Sigma^*0\Sigma^*$$

and so the desired inclusion holds.

($\supseteq$): Any string $\alpha 1(00)^j 1\beta$ is in $L_e(\pi)$, by choosing $x_1 = \alpha, x_7 = \beta$, $x_4 = 0^j$, $x_2 = x_3 = x_5 = x_6 = \epsilon$. Any string $\alpha 0\beta 10(00)^j 1\gamma$ is in $L_e(\pi)$, by choosing $x_1 = \alpha, x_7 = \gamma$, $x_4 = 0^j$, $x_2 = 0$, $x_3 = \beta$

and $x_5 = x_6 = \epsilon$. Any string $\alpha 10(00)^j 1\beta 0\gamma$ is in $L_e(\pi)$, by choosing $x_1 = \alpha, x_7 = \gamma$, $x_4 = 0^j$, $x_2 = x_3 = \epsilon$, $x_5 = 0$ and $x_6 = \beta$. Thus, $L_e(\pi)$ is regular.

Now consider any regular pattern π' such that the shortest word in $L_e(\pi')$ is 11, that is, the shortest word of $L_e(\pi')$ coincides with the shortest word of $L_e(\pi)$. It follows that $\pi' \in V^*1V^*1V^*$. If π' has a variable between the two 1's, then $101 \in L_e(\pi') - L_e(\pi)$; otherwise $1001 \in L_e(\pi) - L_e(\pi')$. Thus, for any regular pattern π', $L_e(\pi) \neq L_e(\pi')$. $\square$

Theorem 2. *Suppose $|\Sigma| = 3$. Then there exists a pattern π such that $L_e(\pi)$ is regular but for no regular pattern π', $L_e(\pi') = L_e(\pi)$.*

Proof. Consider the pattern $\pi = x_1x_2x_30x_2x_4x_4x_51x_6x_5x_7$ and the alphabet $\Sigma = \{0, 1, 2\}$. We first claim that $L_e(\pi)$ is regular. In particular, we claim that the equality

$$L_e(\pi) = \Sigma^*0\Sigma^*1\Sigma^* - 1^*0^*02(22)^*11^*0^*$$

holds.

($\subseteq$): Suppose $w \in L_e(\pi)$. Then, clearly, $w \in \Sigma^*0\Sigma^*1\Sigma^*$. Suppose, by way of contradiction that $w = s(\pi) \in 1^*0^*02(22)^*11^*0^*$, for some substitution s. Thus, $s(x_1x_2x_3) \in 1^*0^*$ and $s(x_6x_5x_7) \in 1^*0^*$. However, then only $s(x_4)$ contains a 2, which can allow only even number of 2's in $s(\pi)$, a contradiction.

($\supseteq$) : Suppose $w \in \Sigma^*0\Sigma^*1\Sigma^* - 1^*0^*02(22)^*11^*0^*$. Let β be the shortest string such that there are α, γ with $w = \alpha 0\beta 1\gamma$; note that β does not contain 0 or 1, as otherwise one could replace β by a part of itself and choose α, γ accordingly. One of the following two cases applies.

First assume that $\beta = 2^{2n}$ for some n. Now let $s(x_2) = s(x_3) = s(x_5) = s(x_6) = \epsilon$, $s(x_4) = 2^n$, $s(x_1) = \alpha$ and $s(x_7) = \gamma$. Thus $w = s(\pi) \in L_e(\pi)$.

Second assume that $\beta = 2^{2n+1}$ for some n. Now, at least one of α or γ contains a 2, as otherwise, we will have that $w \in 1^*0^*02(22)^*11^*0^*$. Suppose $\alpha = \alpha' 2\alpha''$. Then, the substitution $s(x_2) = 2$, $s(x_4) = 2^n$, $s(x_1) = \alpha'$, $s(x_3) = \alpha''$, $s(x_5) = s(x_6) = \epsilon$, $s(x_7) = \gamma$, witnesses that $w \in L_e(\pi)$. One can similarly show that if γ contains a 2, then $w \in L_e(\pi)$.

Thus, $L_e(\pi)$ is regular.

Now consider any regular pattern π' such that the shortest word in $L_e(\pi')$ is 01, that is, the shortest word of $L_e(\pi')$ coincides with the shortest word of $L_e(\pi)$. It follows that $\pi' \in V^*0V^*1V^*$. If π' has a variable between 0 and 1 then $021 \in L_e(\pi') - L_e(\pi)$; otherwise $0221 \in L_e(\pi) - L_e(\pi')$. Thus, for any regular pattern π', $L_e(\pi) \neq L_e(\pi')$. $\square$

4 Block-Regular Patterns and Alphabet Size 4 or Greater

Given a pattern π, a *variable block* of π is a substring $\pi(i)\pi(i+1)\dots\pi(j)$ of π such that (a) $\pi(r)$ is a variable for $i \leq r \leq j$, (b) $i = 0$ or $\pi(i-1) \in \Sigma$ and (c) $j = |\pi| - 1$ or $\pi(j+1) \in \Sigma$. Let $NRV(\pi) = \{x : x$ appears in π exactly once $\}$ (NRV stands for non-repeating variables).

Definition 3. A *block-regular pattern* is a pattern π in which every variable block contains a variable which appears only once in the pattern π, that is, for every variable block $\pi(i)\pi(i+1)$ $\dots\pi(j)$ of π, there is a k such that $i \leq k \leq j$ and $\pi(k) \in NRV(\pi)$.

For the following we use the following version of the pumping lemma and the corresponding corollary for regular languages.

Lemma 4 (Pumping Lemma). *Suppose L is a regular language. Then, there exists a constant n such that, for all strings $z \in L$ and all of its splittings $z_1 z_2 z_3 = z$ with $|z_2| \geq n$, there exists a splitting of z_2 as $z_2 = z_{21} z_{22} z_{23}$ such that $z_{22} \neq \epsilon$ and $z_1 z_{21} z_{22}^m z_{23} z_3 \in L$, for all natural numbers m.*

Corollary 5. *Suppose L is a regular language. Then, there exists a constant n such that, for all strings $z \in L$ and all of its splittings $z_1 z_2 z_3 = z$, there exists a z_2' such that $|z_2'| \leq n$ and $z_1 z_2' z_3 \in L$.*

Theorem 6. (a) *Suppose $|\Sigma| \geq 4$. If π is not a block-regular pattern, then $L_e(\pi)$ and $L_{ne}(\pi)$ are not regular.*

(b) *Suppose π is a block-regular pattern. Then $L_e(\pi) = L_e(\pi')$, for the regular pattern π' which is obtained from π by dropping all variables not in $NRV(\pi)$.*

Proof. (a) Suppose by way of contradiction that π is not a block-regular pattern but $L_e(\pi)$ (respectively, $L_{ne}(\pi)$) is regular.

As π is not a block-regular pattern, there exists a variable block of π which consists only of variables which are repeated in the pattern. Pick such a variable block $\pi(i)\pi(i+1)\ldots\pi(j)$. Let $a, b \in \Sigma$ be such that $a \neq b$ and $a, b \notin \{\pi(i-1), \pi(j+1)\}$ (here if $i = 0$ or $j + 1 = |\pi|$, then we just ignore the corresponding entry).

Now let w be the string obtained by using a substitution s which replaces every variable in π by $a^{n+1}b^{n+1}$, where n is as in the pumping lemma for the regular language $L_e(\pi)$ (respectively, $L_{ne}(\pi)$). Let $w_1 = s(\pi(0)\ldots\pi(i-1))$, $w_2 = s(\pi(i)\ldots\pi(j))$ and $w_3 = s(\pi(j+1)\ldots\pi(|\pi|-1))$.

Note that the number of characters in π which belong to $\Sigma - \{a, b\}$ is exactly the same as the number of characters in w which belong to $\Sigma - \{a, b\}$. Now, by using pumping lemma, we have that, for some substitution s' (which is non-erasing in the case of $L_{ne}(\pi)$), $w' = w_1' w_2' w_3' = s'(\pi)$, where

- $s'(\pi(0)\ldots\pi(i-1)) = w_1'$, $s'(\pi(i)\ldots\pi(j)) = w_2'$, $s'(\pi(j+1)\ldots\pi(|\pi|-1)) = w_3'$,
- w_1' does not end in a or b and w_3' does not start with a or b,
- $|w_1'| \leq n$, $|w_3'| \leq n$ and
- $w_2' = a^{k_i} b^{\ell_i} a^{k_{i+1}} b^{\ell_{i+1}} \ldots a^{k_j} b^{\ell_j}$, where $n < k_i < \ell_i < k_{i+1} < \ell_{i+1} < \ldots < k_j < \ell_j$.

Such w_1', w_2', w_3' can be obtained as follows. Initially, by taking $z = w$, with $z_1 = \epsilon$, $z_2 = w_1$ and $z_3 = w_2 w_3$, in the corollary to the pumping lemma, one can obtain $w_1' = z_2'$ of length at most n. Then, by taking $z = w_1' w_2 w_3$, with $z_1 = w_1' w_2$, $z_2 = w_3$ and $z_3 = \epsilon$, in the corollary to the pumping lemma, one can obtain $w_3' = z_2'$ of length at most n. Then, by starting with $z = w_1' w_2 w_3'$, and repeatedly using pumping lemma, with z_2 being the different segments of a^{n+1} or b^{n+1} in w_2 and taking large enough m for pumping, one can obtain the appropriate w_2' as needed above.

However this leads to a contradiction as follows. Note that $s'(\pi(r))$, where $i \le r \le j$, cannot be of the form $a^+b^+a^+$ or $b^+a^+b^+$, as each $\pi(r)$ is a repeated variable and a substring of form $a^+b^{\ell_m}a^+$, $i \le m \le j$ (respectively, $b^+a^{k_m}b^+$, $i \le m \le j$) does not appear twice in w'. This implies that $s'(\pi(r)) = a^{k_r}b^{\ell_r}$. But then this contradicts the fact that $\pi(j)$ appears at least twice in π, as the string $a^{k_j}b^{\ell_j}$ does not appear twice in w'.

Thus, $L_e(\pi)$ and $L_{ne}(\pi)$ are not regular.

(b) Clearly, $L_e(\pi') \subseteq L_e(\pi)$. We now show that $L_e(\pi) \subseteq L_e(\pi')$. Suppose $w = s(\pi)$, for some substitution s. Let s' be a substitution such that $s'(x) = s(\pi(i)\pi(i+1)\dots\pi(j))$, where $\pi(i)\pi(i+1)\dots\pi(j)$ is a variable block of π and x is the first variable in $\pi(i)\pi(i+1)\dots\pi(j)$ from $NRV(\pi)$. All other variables are mapped by s' to ϵ. Then, it is easy to verify that $w = s'(\pi')$. Thus, $L_e(\pi)$ is a regular pattern language. $\Box$

Corollary 7. *Suppose $|\Sigma| \ge 4$. Then the following three statements are equivalent*

(a) $L_e(\pi)$ *is regular;*

(b) π *is a block-regular pattern;*

(c) $L_e(\pi) = L_e(\pi')$ *for some regular pattern* π'.

5 Non-Erasing Pattern Languages and Regular Patterns

For non-erasing pattern languages, for every non-empty alphabet Σ there is a regular language which is generated by a pattern but not by a regular pattern. This result follows from the work of Reidenbach [9]. Here, we give the details for completeness. We additionally consider block-regular patterns and mention an open problem.

If $\Sigma = \{0\}$, then the pattern xx generates the regular language $00(00)^*$. However, this language cannot be generated by a regular pattern or a block-regular pattern. Furthermore, for $\Sigma = \{0\}$, every block-regular pattern language is generated by a regular pattern.

If $|\Sigma| \ge 2$, that is, if $0, 1 \in \Sigma$, then $L_{ne}(1xy1xz) = \bigcup_{a \in \Sigma} 1a\Sigma^+1a\Sigma^+$ is regular. On the other hand, by [1], we have that $L_{ne}(\pi) = L_{ne}(1xy1xz)$ iff $\pi = 1xy1xz$, except for renaming of variables. Thus, $L_{ne}(1xy1xz)$ is not generated by a regular pattern.

Note that, by Theorem 6(a), for alphabet size at least 4, if π is not a block-regular pattern, then $L_{ne}(\pi)$ is not regular. On the other hand, for block-regular pattern $0x_1x_2x_30x_2x_1x_4$, $L_{ne}(0x_1x_2x_30x_2x_1x_4) \cap 001^*01^*01 = \{001^m01^n01 : m > n \ge 1\}$ is non-regular. Thus, for $|\Sigma| \ge 2$, block-regular patterns can generate languages which are non-regular. Hence, for $|\Sigma| \ge 4$, the class of languages generated by block-regular patterns is a strict generalization of the class of pattern languages which are regular.

One might ask whether a language is regular if every block starts and ends with a variable occurring only once. However, this is also false as shown by the following example. Consider the language

$$B = L_{ne}(x_1y_1y_3x_20x_3y_1y_2y_3x_40x_5y_1y_2y_3x_60x_7y_1y_2y_3x_8)$$

with non-repeating x-variables and repeating y-variables. Every block in the pattern above starts and ends with a non-repeating variable. Consider the intersection

$$A = B \cap 11110112^+110112^+110112^+11 = \{11110112^n110112^n110112^n11 : n \in \{1, 2, \dots\}\}$$

of B with a regular language. The language A is neither regular nor context-free and therefore B is neither regular nor context-free.

An interesting question for non-erasing pattern languages is whether there exist regular languages that can be generated only by patterns which are not block-regular. As already seen, this is impossible for alphabet size at least 4 and possible for alphabet size 1. While the next result provides a solution for alphabet size 2, the question remains open for alphabet size 3.

Proposition 8. *Suppose $\Sigma = \{0, 1\}$. There exists a non-block regular pattern π such that,*
(a) *$L_{ne}(\pi)$ is regular and*
(b) *$L_{ne}(\pi) \neq L_{ne}(\pi')$ for any block regular pattern π'.*

Proof. Let $\pi = x_1 0 y 1 x_2 0 y 1 x_3$. $L_{ne}(\pi)$ is only generated by patterns which are equal to π up to a renaming of the variables [1], thus part (b) follows.

We now show that

$$L_{ne}(\pi) = \Sigma^+ 001 \Sigma^+ 001 \Sigma^+ \cup \Sigma^+ 011 \Sigma^+ 011 \Sigma^+ \cup \Sigma^+ 0101 \Sigma^+ 0101 \Sigma^+$$

and hence $L_{ne}(\pi)$ is regular. To see

$$L_{ne}(\pi) \subseteq \Sigma^+ 001 \Sigma^+ 001 \Sigma^+ \cup \Sigma^+ 011 \Sigma^+ 011 \Sigma^+ \cup \Sigma^+ 0101 \Sigma^+ 0101 \Sigma^+,$$

let $w \in L_{ne}(\pi)$ and choose the values of the variables such that y is as short as possible. Then y cannot be of the form $y' 0 y''$ with $|y''| \geq 1$ — as otherwise one could replace y by y'' and x_1, x_2 by $x_1 0 y'$ and $x_2 0 y'$, respectively, so that y would not be as short as possible. Similarly, if $y = y' 1 y''$ and $|y'| \geq 1$, then one can replace y by y', x_2 by $y'' 1 x_2$ and x_3 by $y'' 1 x_3$; again y would not have been of the shortest possible form. Therefore 0 can occur in y only at the last position and 1 can occur only in the first position. It follows that $w \in \Sigma^+ 0 y 1 \Sigma^+ 0 y 1 \Sigma^+$ for some $y \in \{0, 1, 10\}$. This establishes that $L_{ne}(\pi)$ is a subset of the given regular language.

For the converse direction, assume that $w \in \Sigma^+ 0 y 1 \Sigma^+ 0 y 1 \Sigma^+$ and $y \in \{0, 1, 10\}$. Then it is immediate that one can choose x_1, x_2, x_3 such that $w = x_1 0 y 1 x_2 0 y 1 x_3$ and $w \in L_{ne}(\pi)$. This completes the proof. □

Open Problem 9. *Suppose the alphabet size is 3. Is there a pattern π which is not block-regular such that $L_{ne}(\pi)$ is a regular language?*

6 Pattern Languages Which Are Context-Free But Not Regular

Reidenbach [9] asked whether there are pattern languages which are context-free but not regular. Note that, if alphabet size is 1, then every pattern language is regular. Theorem 10 below shows that, for alphabet size 2 or 3, for both erasing as well as non-erasing case, there are pattern languages which are context-free but not regular. On the other hand, Theorem 12 below shows that, for alphabet size at least 4, every erasing pattern language which is context-free is also regular.

Theorem 10. (a) *Suppose* $\Sigma = \{0, 1\}$*. Then* $L_e(1x1y1x1z)$ *and* $L_{ne}(1x1y1x1z)$ *are context-free but not regular.*

(b) *Suppose* $\Sigma = \{0, 1, 2\}$*. Then* $L_e(y0x1z0x1w)$ *and* $L_{ne}(y0x1z0x1w)$ *are context-free but not regular.*

Proof. (a) Note that

$$\begin{aligned} L_e(1x1y1x1z) &= \{10^n1y10^n1z : y, z \in \Sigma^*, n \geq 0\} \text{ and} \\ L_{ne}(1x1y1x1z) &= \{10^n1y10^n1z : y, z \in \Sigma^+, n > 0\} \cup \{110^n1y110^n1z : y, z \in \Sigma^+, n \geq 0\}. \end{aligned}$$

The above can be verified by using the shortest possible x which witnesses a string to be in $L_e(1x1y1x1z)$ or $L_{ne}(1x1y1x1z)$. Each of these two languages is easily seen to be context-free. However, their intersection with 10^+1010^+10 gives the language $\{10^n1010^n10 : n > 0\}$, which is not regular.

(b) Note that

$$\begin{aligned} L_e(y0x1z0x1w) &= \{y02^n1z02^n1w : y, z, w \in \Sigma^*, n \geq 0\} \text{ and} \\ L_{ne}(y0x1z0x1w) &= \{y02^n1z02^n1w : y, z, w \in \Sigma^+, n > 0\} \cup \\ &\quad \{y012^n1z012^n1w : y, z, w \in \Sigma^+, n \geq 0\} \cup \\ &\quad \{y02^n01z02^n01w : y, z, w \in \Sigma^+, n \geq 0\} \cup \\ &\quad \{y012^n01z012^n01w : y, z, w \in \Sigma^+, n \geq 0\}. \end{aligned}$$

The above can be verified by using the shortest possible x which witnesses a string to be in $L_e(y0x1z0x1w)$ or $L_{ne}(y0x1z0x1w)$. Each of these two languages is easily seen to be context-free. However, their intersection with 202^+1202^+12 gives the language $\{202^n1202^n12 : n > 0\}$, which is not regular. $\square$

We now consider the case of alphabet size at least 4, for erasing pattern languages. Let occur(π, x) (respectively, occur(π, a)) denote the number of times that variable x (respectively, character a) occurs in pattern π. The following lemma is useful to show that pattern languages of certain form are not context-free.

Lemma 11. *Suppose* $\Sigma = \{2, 3\}$*. Suppose* π *is a non-empty pattern consisting of only variables and for each variable* x *in* π*, occur*$(\pi, x) \geq 2$*. Then* $L_e(\pi)$ *is not context-free.*

Proof. Suppose by way of contradiction that π is as in the hypothesis of the lemma, but $L_e(\pi)$ is context-free. Consider a Chomsky Normal Form grammar G for $L_e(\pi) - \{\epsilon\}$. Let x be a variable in π for which occur(π, x) is minimized. Let $k = \text{occur}(\pi, x)$. Now consider any sufficiently long Kolmogorov random string w, that is a string w of Kolmogorov complexity at least $|w|$ (see [7]). Below we will give a description of w which is of length approximately $\frac{(2k-1)|w|}{2k}$, contradicting randomness of w.

The string w can be described as follows. Let $w' = s(\pi)$, where s maps x to w and rest of the variables to ϵ. Consider the derivation tree of w' in G. Then, there exists a node corresponding to a non-terminal A in the tree such that, in this tree, A derived a substring w_2 of $w' = w_1 w_2 w_3$, where $|w|/2 \leq |w_2| \leq |w|$. Let u be the shortest string that can be generated by the non-terminal A in grammar G. Then, $w'' = w_1 u w_3 \in L_e(\pi)$. Thus, as w'' can be described using the substitution needed to obtain w'', complexity of w'' is at most $|w''|/k$ plus a constant. Thus, w'' has complexity at most $\frac{(2k-1)|w|}{2k}$ plus a constant. Note that $w = w_1' w_3'$, for some appropriate prefix w_1' of w_1 and suffix w_3' of w_3. Thus, w can be obtained from w'' by just giving the location where the w_1' ends and w_3' starts in w''. Thus, w has complexity at most $\frac{(2k-1)|w|}{2k} + 2\log_2(|w|) + c$, for a constant c. For sufficiently long w, this contradicts the randomness of w. $\square$

Theorem 12. *Suppose $\Sigma \supseteq \{0,1,2,3\}$ and let a pattern π be given. If $L_e(\pi)$ is context-free, then it is regular.*

Proof. Without loss of generality assume that π starts and ends with a member of Σ (this is just for ease of writing the proof). Suppose $L_e(\pi)$ is not regular. Then, by Theorem 6(b), there exists a variable block $\pi(i)\pi(i+1)\ldots\pi(j)$, with $i \leq j$, such that $\mathrm{occur}(\pi, \pi(r)) \geq 2$, for $i \leq r \leq j$. Fix one such i, j. Without loss of generality assume that $\pi(i-1) = 0$ and $\pi(j+1) = 1$.

Let $Y = \{\pi(r) : i \leq r \leq j\}$.

Let $X = \{x \in Y : \mathrm{occur}(\pi, x) = \mathrm{occur}(\pi(i)\pi(i+1)\ldots\pi(j), x)\}$. That is, X is the set of variables x that appear in π but neither in $\pi(0)\ldots\pi(i-1)$ nor in $\pi(j+1)\ldots\pi(|\pi|-1)$.

We will now show that $L_e(\pi)$ is not context-free. So suppose by way of contradiction that $L_e(\pi)$ is context-free. We consider the following cases.

Case 1: $X \neq \emptyset$.

Let π' be the pattern formed from $\pi(i)\ldots\pi(j)$ by dropping the variables $x \notin X$.

Consider the language $L = \{s(\pi) : s(x) \in \{2,3\}^*$, if $x \in X$, and $s(x) = \epsilon$ otherwise$\}$. Then L is also context-free (as it can be obtained by taking intersection of $L_e(\pi)$ with a regular set formed by replacing each variable $x \in X$ (treating each occurrence of a variable as distinct) by $\{2,3\}^*$, and rest of the variables by ϵ). But then, this would imply that $L(\pi')$ is context-free, contradicting Lemma 11.

Case 2: $X = \emptyset$.

Let q be the maximum value of $\frac{\mathrm{occur}(\pi(i)\pi(i+1)\ldots\pi(j),x)}{\mathrm{occur}(\pi,x)}$, for the variables $x \in Y$.

Let $Z = \{x \in Y : \frac{\mathrm{occur}(\pi(i)\pi(i+1)\ldots\pi(j),x)}{\mathrm{occur}(\pi,x)} = q\}$.

Consider the language $L = \{s(\pi) : s(x) \in 2^*3^*$ for each variable $x \in Z$ and $s(x) = \epsilon$ for $x \notin Z\}$. Then L is also context-free (as it can be obtained by taking intersection of $L_e(\pi)$ with a regular set formed from π by replacing each variable $x \in Z$ (treating each occurrence of a variable as distinct) by 2^*3^* and each variable $x \notin Z$ by ϵ). Let $n_2 = \mathrm{occur}(\pi, 2)$ and $n_3 = \mathrm{occur}(\pi, 3)$. Note that, for each $w \in L$, the number of 2's (3's) in w occurring in between the characters corresponding to $\pi(i-1)$ and $\pi(j+1)$ is at most $q*(\mathrm{occur}(w,2)-n_2)$ (respectively, $q*(\mathrm{occur}(w,3)-n_3)$).

Let n be as in the pumping lemma for context-free languages for L (see [4]). Consider string $w = s(\pi)$ obtained by using the substitution s which maps $x \in Z$ to 2^n3^n and all other variables to ϵ. Now by pumping lemma for context-free languages, there exists a splitting of w into $uvzv'u'$

such that $|vzv'| \leq n$, $|vv'| \geq 1$, and both $uvvzv'v'u'$ and uzu' are in L. Thus, vv' cannot contain 0 or 1, and

(i) both v and v' lie fully within the portion of w corresponding to $\pi(i) \ldots \pi(j)$ or
(ii) both v and v' do not contain any part of $\pi(i) \ldots \pi(j)$ or
(iii) both v and v' do not contain any 2's from the portion of w corresponding to $\pi(i) \ldots \pi(j)$ or
(iv) both v and v' do not contain any 3's from the portion of w corresponding to $\pi(i) \ldots \pi(j)$.

If (i) holds, then $w' = uvvzv'v'u'$ will have more than $q*(\text{occur}(w', 2) - n_2)$ many 2's or more than $q*(\text{occur}(w', 3) - n_3)$ many 3's between the characters corresponding to $\pi(i-1)$ and $\pi(j+1)$. If (ii) holds, then $w' = uzu'$ will have more than $q*(\text{occur}(w', 2) - n_2)$ many 2's or more than $q*(\text{occur}(w', 3) - n_3)$ many 3's between the characters corresponding to $\pi(i-1)$ and $\pi(j+1)$.

So suppose (i) and (ii) do not hold and (iii) holds (case of (iv) holds is similar). If vv' contains a 2, then $w' = uzu'$ will have more than $q*(\text{occur}(w', 2) - n_2)$ many 2's between the characters corresponding to $\pi(i-1)$ and $\pi(j+1)$. On the other hand, if vv' does not contain a 2, then it consists only of 3's, including at least one 3 corresponding to the portion of w in $\pi(j+1) \ldots \pi(|\pi|-1)$. But then, this implies that the string uzu' does not have the appropriate sequence of constants following the character corresponding to $\pi(j)$, as required for all strings in L. $\square$

For alphabet size at least 4, it is unknown whether every context-free non-erasing pattern language is also regular; the main reason is that there are languages generated by block-regular patterns which are not regular languages; it might be that some of them are context-free, although we think at the moment that this is unlikely.

Open Problem 13. *Suppose $|\Sigma| \geq 4$. Is there a pattern π such that $L_{ne}(\pi)$ is context-free but not regular?*

Acknowledgment

We would like to thank Daniel Reidenbach, Yang Yue and the anonymous referees for useful information and providing various pointers to the literature.

References

1. Dana Angluin. Finding patterns common to a set of strings. *Journal of Computer and System Sciences*, 21:46–62, 1980.
2. Thomas Erlebach, Peter Rossmanith, Hans Stadtherr, Angelika Steger and Thomas Zeugmann. Learning one-variable pattern languages very efficiently on average, in parallel and by asking queries. *Theoretical Computer Science*, 261:119–156, 2001.
3. Dominik D. Freydenberger and Daniel Reidenbach. Bad news on decision problems for patterns. *Information and Computation*, 208:83–96, 2010.

4. John E. Hopcroft, Rajeev Motwani and Jeff D. Ullman. Introduction to Automata Theory, Languages and Computation, 2nd edition. Addison Wesley, 2001.
5. Michael Kearns and Leonard Pitt. A polynomial-time algorithm for learning k-variable pattern languages from examples. *Proceedings of the Second Annual Workshop on Computational Learning Theory*, Santa Cruz, California, USA, pages 57–71, 1989.
6. Steffen Lange and Rolf Wiehagen. Polynomial time inference of arbitrary pattern languages. *New Generation Computing*, 8:361–370, 1991.
7. Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and its Applications.* Third Edition. Springer, 2008.
8. Daniel Reidenbach. A non-learnable class of E-pattern languages. *Theoretical Computer Science*, 350:91–102, 2006.
9. Daniel Reidenbach. The ambiguity of morphisms in free monoids and its impacts on algorithmic properties of pattern languages. PhD Thesis, University of Kaiserslautern, Germany, 2006.
10. Peter Rossmanith and Thomas Zeugmann. Learning k-variable pattern languages efficiently stochastically finite on average from positive data. *Grammatical Inference, Fourth International Colloquium*, ICGI 1998, Ames, Iowa, USA, July 1998, Proceedings. *Springer LNAI*, 1433:13–24, 1998.
11. Takeshi Shinohara. Polynomial time inference of extended regular pattern languages. *RIMS Symposia on Software Science and Engineering*, Kyoto, Japan, Proceedings. *Springer LNCS*, 147:115–127, 1982.
12. Takeshi Shinohara and Setsuo Arikawa. Pattern inference. *Algorithmic Learning for Knowledge-Based Systems*, Springer LNAI 961:259–291, 1995.
13. Thomas Zeugmann. Lange and Wiehagen's pattern language learning algorithm: An average-case analysis with respect to its total learning time. *Annals of Mathematics and Artificial Intelligence*, 23:117–145, 1998.