Lange and Wiehagen's Pattern Language Learning Algorithm: An Average-Case Analysis with respect to its Total Learning Time*

THOMAS ZEUGMANN

Department of Informatics, Kyushu University,
Kasuga 816-8580, Japan
E-mail: thomas@i.kyushu-u.ac.jp

The present paper deals with the best-case, worst-case and average-case behavior of Lange and Wiehagen's (1991) pattern language learning algorithm with respect to its *total learning time*. Pattern languages have been introduced by Angluin (1980) and are defined as follows:

Let $\mathcal{A} = \{0, 1, \ldots\}$ be any non–empty finite alphabet containing at least two elements. Furthermore, let $X = \{x_i \mid i \in \mathbb{N}\}$ be an infinite set of variables such that $\mathcal{A} \cap X = \emptyset$. *Patterns* are non–empty strings over $\mathcal{A} \cup X$. $L(\pi)$, the language generated by pattern π is the set of strings which can be obtained by substituting non-null strings from $\mathcal{A}^*$ for the variables of the pattern π.

Lange and Wiehagen's (1991) algorithm learns the class of all pattern languages in the limit from text. We analyze this algorithm with respect to its *total learning time* behavior, i.e., the overall time taken by the algorithm until *convergence*. For every pattern π containing k different variables it is shown that the total learning time is $O(|\pi|^2 \log_{|\mathcal{A}|}(|\mathcal{A}| + k))$ in the best-case and unbounded in the worst-case. Furthermore, we estimate the expectation of the total learning time. In particular, it is shown that Lange and Wiehagen's algorithm possesses an expected total learning time of $O(2^k k^2 |\pi|^2 \log_{|\mathcal{A}|}(k|\mathcal{A}|))$ with respect to the uniform distribution.

1. Introduction

The setting we want to deal with is the average-case analysis of pattern language learning algorithms. The pattern languages have been formally introduced by Angluin [1], and have been widely investigated recently (cf., e.g., Salomaa [20, 21], and Shinohara and Arikawa [26] for an overview). Moreover, Angluin [1] also proved that the class of all pattern languages is learnable in the limit from positive data. Subsequently, Shinohara [24] dealt with polynomial time learnability of subclasses of pattern languages. Nix [18] as well as Shinohara and Arikawa [25] firstly outlined interesting applications of pattern inference algorithms. Recently, pattern language learning algorithms have been successfully applied for solving problems in molecular biology, too (cf., e.g. [23, 26]).

Additionally, the learnability of pattern languages has been considered in a variety of learning models. For example, Angluin [2] investigated the problem of using queries to learn the class *PAT* of all pattern languages. Marron [17] refined this scenario by studying their learnability from a single example and from queries. Subsequently, Lange and Wiehagen [12] showed that *PAT* can be learned from disjointness queries, too; thus solving a problem that remained open in [2]. Moreover, they also presented the first algorithm that iteratively learns all pattern languages in the limit. Wiehagen and Zeugmann [28] dealt with consistent versus inconsistent pattern language learning in the limit. Furthermore, Lange and Zeugmann [13] as well as Zeugmann, Lange and Kapur [29] investigated the learnability of pattern languages from positive data under monotonicity constraints and with respect to different sets of allowed hypothesis spaces, again considering learning in the limit.

*This work has been supported by the Grant-in-Aid for Scientific Research (C) from the Japanese Ministry of Education, Science, Sports, and Culture under grant no. 07680403.

Kearns and Pitt [9], Ko, Marron and Tzeng [11] and Schapire [22] intensively studied the learnability of pattern languages in the PAC–learning model. In particular, Schapire [22] proved that the class *PAT* is not PAC-learnable regardless of the representation used by the learning algorithm, provided only that the learner is requested to output a polynomial-size hypothesis that can be evaluated in polynomial time, unless $\mathcal{P}_{/poly} = \mathcal{NP}_{/poly}$. However, the class *Pat* of all patterns is not a polynomial time representation for *PAT*, since the membership problem for *PAT* with respect to *Pat* is $\mathcal{NP}$-complete (cf. [1]). For further research along this line, the reader is referred to [26].

Jiang *et al.* [8] proved that inclusion for pattern languages is undecidable. The latter result has some implications to the learnability of all the pattern languages, too (cf. [29]). Finally, Kilpeläinen *et al.* [10] studied the learnability of unions of simple patterns using the minimum description length principle.

This continuous interest in the pattern languages and their applications motivated us to initiate the analysis of pattern language learning algorithms with respect to their *average-case behavior.* This is important for unifying the formal mathematical and the empirical approaches to gain a better understanding of the behavior of machine learning algorithms. Moreover, research along this line seems to be inevitable for obtaining results allowing assertions concerning the efficiency of algorithms learning in the limit that are of major interest with respect to potential applications.

The present paper deals with the algorithm proposed by Lange and Wiehagen [12]. In particular, their algorithm learns the whole class of all pattern languages from positive data. Lange and Wiehagen [12] showed that their algorithm has polynomial update time. However, our goal is more ambitious. We analyze the best-case, worst-case and average-case complexity of this algorithm with respect to the *total learning time.* The total learning time is the sum of all update times taken by the algorithm until successful learning. In particular, we show that the average-case complexity of Lange and Wiehagen's [12] algorithm is $O(2^k k^2 |\pi|^2 \log_{|\mathcal{A}|}(k|\mathcal{A}|))$ with respect to the uniform distribution for any pattern π containing k different variables.

2. Preliminaries

Let $\mathbb{N} = \{0, 1, 2, \ldots\}$ be the set of all natural numbers, and let $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. For all real numbers x we define $\lfloor x \rfloor$, the *floor function*, to be the greatest integer less than or equal to x.

Following Angluin [1] we define patterns and pattern languages as follows. Let $\mathcal{A} = \{0, 1, \ldots\}$ be any non-empty finite alphabet containing at least two elements. By $\mathcal{A}^*$ we denote the free monoid over $\mathcal{A}$ (cf. Hopcroft and Ullman [7]). The set of all finite non-null strings of symbols from $\mathcal{A}$ is denoted by $\mathcal{A}^+$, i.e., $\mathcal{A}^+ = \mathcal{A}^* \setminus \{\varepsilon\}$, where ε denotes the empty string. By $|\mathcal{A}|$ we denote the cardinality of $\mathcal{A}$. Furthermore, let $X = \{x_i \mid\ i \in \mathbb{N}\}$ be an infinite set of variables such that $\mathcal{A} \cap X = \emptyset$. *Patterns* are non-empty strings over $\mathcal{A} \cup X$, e.g., 01, $0x_0111$, $1x_0x_00x_1x_2x_0$ are patterns. The length of a string $w \in \mathcal{A}^*$ and of a pattern π is denoted by $|w|$ and $|\pi|$, respectively. A pattern π is in *canonical form* provided that if k is the number of different variables in π then the variables occurring in π are precisely $x_0, \ldots, x_{k-1}$. Moreover, for every j with $0 \leq j < k-1$, the leftmost occurrence of x_j in π is left to the leftmost occurrence of x_{j+1} in π. The examples given above are patterns in canonical form. In the sequel we assume, without loss of generality, that all patterns are in canonical form. By *Pat* we denote the set of all patterns in canonical form.

Let $\pi \in Pat$, $1 \leq i \leq |\pi|$; we use $\pi(i)$ to denote the i-th symbol in π. If $\pi(i) \in \mathcal{A}$, then we refer to $\pi(i)$ as to a *constant*; otherwise $\pi(i) \in X$, and we refer to $\pi(i)$ as to a *variable.* By $\#\text{var}(\pi)$ we denote the number of different variables occurring in π, and by $\#_{x_i}(\pi)$ we denote the number of occurrences of variable x_i in π. If $\#\text{var}(\pi) = k$, then we refer to π as to a *k-variable pattern.* Let $k \in \mathbb{N}$, by Pat_k we denote the set of all *k-variable patterns.* Furthermore, let $\pi \in Pat_k$, and let $u_0, \ldots, u_{k-1} \in \mathcal{A}^+$; then we denote by

$\pi[x_0\colon u_0, \dots, x_{k-1}\colon u_{k-1}]$ the string $w \in \mathcal{A}^+$ obtained by substituting u_j for each occurrence of x_j, $j = 0, \dots, k-1$, in the pattern π. The tuple $(u_0, \dots, u_{k-1})$ is called *substitution*. Furthermore, if $|u_0| = \dots = |u_{k-1}| = 1$, then we refer to $(u_0, \dots, u_{k-1})$ as to a *shortest substitution*. Now, let $\pi \in Pat_k$, and let $S = \{(u_0, \dots, u_{k-1}) \mid u_j \in \mathcal{A}^+,\ j = 0, \dots, k-1\}$ be any finite set of substitutions. Then we set $S(\pi) = \{\pi[x_0\colon u_0, \dots, x_{k-1}\colon u_{k-1}] \mid (u_0, \dots, u_{k-1}) \in S\}$, i.e., $S(\pi)$ is the set of all strings obtained from pattern π by applying all the substitutions from S to it. For every $\pi \in Pat_k$ we define the *language generated by pattern* π by $L(\pi) = \{\pi[x_0\colon u_0, \dots, x_{k-1}\colon u_{k-1}] \mid u_0, \dots, u_{k-1} \in \mathcal{A}^+\}$. By PAT_k we denote the set of all *k-variable pattern languages*. Finally, $PAT = \bigcup_{k \in \mathbb{N}} PAT_k$ denotes the set of all pattern languages over $\mathcal{A}$. Note that for every $L \in PAT$ there is precisely one pattern $\pi \in Pat$ such that $L = L(\pi)$ (cf. Angluin [1]).

In order to deal with the learnability of pattern languages we have to specify from what information the learning algorithms should do their task. Following Gold [5] we may distinguish between learning from *positive data* or both *positive and negative data*. However, the pattern languages are a famous example for a non-trivial class of languages that can be learned from positive data. Therefore, we consider in this paper learning from positive data, only. Formally, let $L \subseteq \mathcal{A}^*$; then every mapping t from $\mathbb{N}$ onto L is called a *text* for L or, synonymously, a *positive presentation*. By *Text(L)* we denote the set of all texts for L. Furthermore, let t be a text, and let $n \in \mathbb{N}$. We set $t_n = t(0), \dots, t(n)$, and we refer to t_n as to the initial segment of t of length $n+1$. Moreover, we define t_n^+ to denote the range of t_n, i.e., $t_n^+ = \{t(i) \mid 0 \le i \le n\}$.

Intuitively, a text for L generates the language L without any information concerning the complement of L. Note that we allow a text to be non-effective.

As in Gold [5], we define an *inductive inference machine* (abbr. IIM) to be an algorithmic device which works as follows: The IIM takes as its input larger and larger initial segments of a text t and on every input it first outputs a hypothesis, i.e., a pattern, and then it requests the next input. Now we are ready to define learnability of pattern languages from positive data in the limit.

DEFINITION 1

PAT is called learnable in the limit from text (abbr. PAT $\in$ LIM) iff there is an IIM M such that for every $L \in PAT$ and every $t \in$ Text(L),

(1) *for all $n \in \mathbb{N}$, $M(t_n)$ is defined,*

(2) *there is a pattern $\pi \in Pat$ such that $L(\pi) = L$ and for almost all $n \in \mathbb{N}$, $M(t_n) = \pi$.*

Whenever one deals with the average case analysis of algorithms one has to consider probability distributions over the relevant input domain. For learning from text, we have the following scenario. Every string of a particular pattern language is generated by a substitution. Therefore, it is convenient to consider probability distributions over the set of all possible substitutions. That is, if $\pi \in Pat_k$, then it suffices to consider any probability distribution D over $\underbrace{\mathcal{A}^+ \times \dots \times \mathcal{A}^+}_{k-\text{times}}$. For $(u_0, \dots, u_{k-1}) \in \mathcal{A}^+ \times \dots \times \mathcal{A}^+$ we denote by $D(u_0, \dots, u_{k-1})$ the probability that variable x_0 is substituted by u_0, variable x_1 is substituted by u_1, ..., and variable x_{k-1} is substituted by u_{k-1}. Moreover, in order to arrive at admissible information sequences, i.e., texts, we restrict ourselves to distributions D such that $D(u_0, \dots, u_{k-1}) > 0$ for every $(u_0, \dots, u_{k-1}) \in \mathcal{A}^+ \times \dots \times \mathcal{A}^+$. We refer to any such distribution as to an *admissible distribution* for PAT_k.

In particular, we mainly consider a special class of admissible distributions, i.e., *product distributions*. Let $k \in \mathbb{N}^+$, then the class of all product distributions for Pat_k is defined as follows. For each variable x_j, $0 \le j \le k-1$, we assume an arbitrary probability distribution D_j

over $\mathcal{A}^+$ on substitution strings. Then we call $D = D_0 \times \cdots \times D_{k-1}$ product distribution over $\mathcal{A}^+ \times \cdots \times \mathcal{A}^+$, i.e., $D(u_0, \dots, u_{k-1}) = \prod_{j=0}^{k-1} D_j(u_j)$. Moreover, we call a product distribution *regular* if $D_0 = \cdots = D_{k-1}$. As a special case of a regular product distribution we consider the *uniform* distribution over $\mathcal{A}^+$, i.e., $D_j(u) = 1/(2 \cdot |\mathcal{A}|)^\ell$ for all $j \in \{0, \cdots k-1\}$ and all strings $u \in \mathcal{A}^+$ with $|u| = \ell$. Furthermore, with respect to potential applications it is also reasonable to consider *length biased uniform* distributions over $\mathcal{A}^+$ defined as follows. Again, all strings of length ℓ, $\ell \in \mathbb{N}^+$, are defined to be equally likely but the "weight" factor for the length ℓ is not necessarily $1/2^\ell$. Instead, we allow any sequence $(\mu_\ell)_{\ell \in \mathbb{N}^+}$ satisfying $\mu_\ell > 0$ for all $\ell \in \mathbb{N}^+$, and $\sum_{\ell \geq 1} \mu_\ell = 1$ as "weight" factors.

Additionally, we assume familiarity with discrete probability theory. For the sake of completeness we recall some fundamental notions that are extensively used throughout the paper. Let X be any random variable that takes natural numbers as its values. Then it is often very convenient to study its *probability generating function* (abbr. pgf) G_X which is defined as follows:

$$G_X(z) = \sum_{\ell \geq 0} Pr(X = \ell) z^\ell \tag{1}$$

Note that all the coefficients in (1) are nonnegative, and that they sum to 1, i.e., $G_X(1) = 1$. Thus, the power series (1) is absolutely convergent for all z with $|z| \leq 1$, where $|z|$ denotes the absolute value of z. Consequently, we may compute the first derivative of G_X by differentiating its summands, i.e.,

$$G'_X(z) = \sum_{\ell \geq 0} Pr(X = \ell) \cdot \ell \cdot z^{\ell - 1} \tag{2}$$

Moreover, $G'_X(z)$ is also absolutely convergent, and the radius of convergence of G_X and G'_X coincide. Thus, the expectation and variance of X can be computed as follows:

$$E(X) = G'_X(1) \tag{3}$$

$$V(X) = G''_X(1) + G'_X(1) - G'_X(1)^2 \tag{4}$$

provided the power series obtained still do converge for $z = 1$. Furthermore, if X is any random variable that takes only nonnegative integer values, we can decompose its pgf into a sum of conditional pgf's with respect to any other discrete random variable Y as follows (cf. Graham, Knuth, Patashnik [6]):

$$G_X(z) = \sum_{y \in rg(Y)} Pr(Y = y) g_{X|y}(z) \tag{5}$$

Here $rg(Y)$ denotes the range of Y, and $g_{X|y}$ is the pgf for the random variable $X|y$, i.e., X under the knowledge that $Y = y$. Hence $g_{X|y}$ just describes all the probabilities $Pr(X = x \mid Y = y)$, $x \in rg(X)$. For any further information concerning random variables and their probability generating functions the reader is referred to Graham, Knuth and Patashnik [6].

Finally, our main goal consists in analyzing the average-case behavior of Lange and Wiehagen's [12] pattern language learning algorithm with respect to its *total learning time*. Following Daley and Smith [4] we define the total learning time as follows. Let M be any IIM that learns all the pattern languages. Then, for every $L \in PAT$ and $t \in Text(L)$, let

$$Conv(M, t) =_{df} \text{the least number } m \text{ such that for all } n \geq m,\ M(t_n) = M(t_m)$$

denote the *stage of convergence* of M on t. Moreover, by $T_M(t_n)$ we denote the time to compute $M(t_n)$. Finally, the total learning time taken by the IIM M on successive input t is defined as

$TT(M,t) =_{df} \sum_{n=0}^{Conv(M,t)} T_M(t_n)$. Assuming any fixed probability distribution D as described above, we aim to evaluate the *expectation* of $TT(M,t)$ with respect to D which we refer to as to the *average total learning time.*

Looking at the latter definition it is obvious that we have to analyze carefully the criterion of convergence of the learning algorithm we are going to consider. This is best done by initially studying the best-case as well as the worst-case behavior of the algorithm. Subsequently, our strategy for determining the average total learning time is as follows. First, we present a theorem that allows us to estimate the average total learning time in terms of the *expected stage of convergence* (cf. Theorem 8). Next, we mainly reduce the estimation of the expected stage of convergence to the estimation of the *expected number of examples* that are necessary to fulfil the criterion of convergence and a term involving the *average input length* until convergence (cf. Theorem 9). Then, we derive general formulae to determine the average input length. Finally, we evaluate the resulting formulae for the uniform distribution and estimate $E(TT(M,t))$ for the IIM realizing Lange and Wiehagen's [12] pattern language learning algorithm (cf. Theorem 11).

The model of computation as well as the representation of patterns we assume is the same as in Angluin [1]. In particular, we assume a random access machine that performs a reasonable menu of operations each in unit time on registers of length $O(\log n)$ bits, where n is the input length.

3. Lange and Wiehagen's Algorithm

In this section we analyze the pattern language learning algorithm by Lange and Wiehagen [12] (abbr. *LWA*) with respect to its worst-case and best-case behavior. For the sake of presentation, let us first recall the *LWA*. The main operation executed by the algorithm is the *union* of a pattern and a string defined as follows:

Let $\pi \in Pat$, $w \in \mathcal{A}^+$ with $|\pi| = |w|$. The union of π and w, denoted by $\pi \cup w$, is the following pattern τ. For $i = 1, \dots, |\pi|$, let

$$\tau(i) = \begin{cases} \pi(i), & \text{if } \pi(i) = w(i) \\ x_j, & \text{if } \pi(i) \neq w(i) \ \& \ \exists k < i : \\ & [\tau(k) = x_j,\ w(k) = w(i),\ \pi(k) = \pi(i)] \\ x_m, & \text{otherwise, where } m = \#var(\tau(1) \dots \tau(i-1)) \end{cases}$$

where $\tau(0) = \varepsilon$ for notational convenience.

Now, the IIM M realizing the *LWA* can be defined as follows. Let $\pi \in Pat$, and let $t = w_0, w_1, w_2, \dots$ be any text for $L(\pi)$. Set $h_{-1} = \varepsilon$. Then,

$$M(h_{-1}, w_0) = M(\varepsilon, w_0) = w_0,$$

and for all $n \geq 1$,

$$M(h_{n-1}, w_n) = h_n = \begin{cases} h_{n-1}, & \text{if } |h_{n-1}| < |w_n| \\ w_n, & \text{if } |h_{n-1}| > |w_n| \\ h_{n-1} \cup w_n, & \text{if } |h_{n-1}| = |w_n| \end{cases}$$

Note that the *LWA* exclusively uses its last guess h_{n-1} and the new string w_n for computing its actual hypothesis h_n. Algorithms behaving thus are called *iterative.* Iterative learning algorithms are of special interest with respect to potential applications, since they allow *incremental* learning, and they are clearly more efficient with respect to space than arbitrary IIMs. However, in general incremental learning constitutes a severe restriction of the learning power (cf. Lange and Zeugmann [13,16]). Moreover, note that the *LWA* outputs exclusively canonical patterns (cf. Lange and Wiehagen [12]).

In the following, we mainly study the *time complexity* of the *LWA*. However, this analysis will also provide insight into its space complexity.

3.1. BEST-CASE AND WORST-CASE ANALYSIS OF THE *LWA*

As already mentioned, we have to analyze the criterion of convergence for the *LWA*. We assume input/output operations to be performed in unit time. Due to the choice of our model of computation, the comparison of $|h_{n-1}|$ and $|w_n|$ can be performed in time $O(\min\{|h_{n-1}|, |w_n|\})$. Moreover, it is convenient to perform the desired analysis in dependence on the number of different variables the target pattern π possesses. If this number is zero, then everything is trivial, i.e., the *LWA* immediately converges. Therefore, in the following let $k \in \mathbb{N}^+$, and let $\pi \in Pat_k$. Taking into account that $|w| \geq |\pi|$ for every $w \in L(\pi)$, it is obvious that the *LWA* can only converge if it has been fed sufficiently many strings from $L(\pi)$ having minimal length. Furthermore, as a closer look to the *LWA* immediately shows, after having seen one string from $L(\pi)$ having minimal length the *LWA* exclusively uses shortest strings from $L(\pi)$ to possibly change its actual hypothesis. Therefore, let

$$L(\pi)_{min} = \{w \mid \ w \in L(\pi),\ |w| = |\pi|\}.$$

As pointed out by Marron [17] (cf. Lemma 2.1., pp. 348) $k+1$ examples from $L(\pi)_{min}$ are sufficient to achieve convergence, e.g., one may take $\pi[x_0\colon 0, \ldots, x_{k-1}\colon 0]$, $\pi[x_0\colon 1, x_1\colon 0, \ldots, x_{k-1}\colon 0]$, $\pi[x_0\colon 0, x_1\colon 1, x_2\colon 0, \ldots, x_{k-1}\colon 0]$, $\ldots$, $\pi[x_0\colon 0, x_1\colon 0, \ldots, x_{k-1}\colon 1]$. However, this bound is by no means the best possible one as we shall show. For that purpose, first we make the following important observation.

LEMMA 1

Let $k \in \mathbb{N}^+$, and let $\pi \in Pat_k$. Then we have:
Every string from $L(\pi)_{min}$ is uniquely generated by a shortest substitution.

Proof

Let $w_1, w_2 \in L(\pi)_{min}$, and let $\bar{u}_1 = (u_0^1, \ldots, u_{k-1}^1)$ as well as $\bar{u}_2 = (u_0^2, \ldots, u_{k-1}^2)$ such that $w_1 = \pi[x_0\colon u_0^1, \ldots, x_{k-1}\colon u_{k-1}^1]$ and $w_2 = \pi[x_0\colon u_0^2, \ldots, x_{k-1}\colon u_{k-1}^2]$. Now, it suffices to show that $w_1 = w_2$ implies $\bar{u}_1 = \bar{u}_2$. Suppose the converse, i.e., $\bar{u}_1 \neq \bar{u}_2$. Then there exists a $j \in \{0, \ldots, k-1\}$ such that $u_j^1 \neq u_j^2$. Let $\ell \in \{1, \ldots, |\pi|\}$ be the least number such that $\pi(\ell) = x_j$. Since $|u_0^1| = \cdots = |u_{k-1}^1| = |u_0^2| = \cdots = |u_{k-1}^2| = 1$, we directly obtain $w_1(\ell) = u_j^1$ as well as $w_2(\ell) = u_j^2$. Hence, we have $w_1(\ell) \neq w_2(\ell)$, a contradiction. □

Next, we introduce the notion of a *good sample* that will be very helpful for our intended analysis.

DEFINITION 2

Let $k \in \mathbb{N}^+$, let $\pi \in Pat_k$, and let $S = \{w_0, \ldots, w_{m-1}\} \subseteq L(\pi)_{min}$. S is said to be a good sample of size m if the LWA, when successively fed $w_0, \ldots, w_{m-1}$, converges to π.

Clearly, the latter definition requires some *justification*, since the notion of a good sample of size m may depend on the *order* in which the strings $w_0, \ldots, w_{m-1}$ are presented to the learner. However, it does not, since the *LWA* possesses another favorable property, i.e., it is *set-driven* (cf. Theorem 2 below). Set-drivenness has been introduced by Wexler and Culicover [27], and is defined as follows.

DEFINITION 3

An IIM is said to be set-driven with respect to PAT iff its output depends only on the range of its input; that is, iff $M(t_x) = M(\hat{t}_y)$ *for all* $x, y \in \mathbb{N}$, *and all texts* t, $\hat{t} \in \bigcup_{L \in PAT} Text(L)$ *provided* $t_x^+ = \hat{t}_y^+$.

Note that in general one cannot expect to learn set-drivenly. For more information concerning this subject the reader is referred to Lange and Zeugmann [15]. Now we are ready to present the announced theorem.

THEOREM 2

The LWA is set-driven with respect to PAT.

Proof

Let π_1, $\pi_2 \in Pat$, let $t \in Text(L(\pi_1))$, $\hat{t} \in Text(L(\pi_2))$, and let $x, y \in \mathbb{N}$ such that $t_x^+ = \hat{t}_y^+$. We have to show that the IIM M realizing the *LWA*, when successively fed t_x and $\hat{t}_y$, respectively, outputs the same hypothesis, say π.

Let $\ell = \min\{|w| \mid w \in t_x^+\}$. Because of $t_x^+ = \hat{t}_y^+$, we get $\ell = \min\{|w| \mid w \in \hat{t}_y^+\}$, too. Taking M's definition into account, it obviously suffices to consider M's behavior when successively fed $\sigma = w_0, \ldots, w_m$, and $\hat{\sigma} = \hat{w}_0, \ldots, \hat{w}_n$, respectively, where w_j, $0 \leq j \leq m$ and $\hat{w}_j$, $0 \leq j \leq n$, are all strings of length ℓ enumerated in t_x and $\hat{t}_y$, respectively. Moreover, it is not hard to see that σ and $\hat{\sigma}$ can be assumed to be repetition free, too, i.e., $m = n$. Note that $range(\sigma) = range(\hat{\sigma})$, since $t_x^+ = \hat{t}_y^+$.

Now, assume that π and $\hat{\pi}$ are output by M when successively fed σ and $\hat{\sigma}$, respectively. Then, obviously we have $|\pi| = |\hat{\pi}|$. If $\pi = \hat{\pi}$, we are done. Thus, suppose $\pi \neq \hat{\pi}$. Furthermore, let $i \in \{1, \ldots, |\pi|\}$ be the least $\tilde{\imath}$ such that $\pi(\tilde{\imath}) \neq \hat{\pi}(\tilde{\imath})$.

Case 1. $\pi(i) \in \mathcal{A}$

By the transitivity of the equality relation we may conclude that $\pi(i) \in \mathcal{A}$ can happen if and only if $\pi(i) = w_j(i)$ for all $j = 0, \ldots, m$. However, if $\pi(i) \neq \hat{\pi}(i)$ then there must be a string $\hat{w} \in range(\hat{\sigma})$ such that $\hat{w}(i) \neq \pi(i)$. Consequently, $\hat{w}(i) \neq w_j(i)$ for all $j = 0, \ldots, m$. But this is a contradiction to $range(\sigma) = range(\hat{\sigma})$.

Hence, we already know that $\pi(i) = \hat{\pi}(i)$ provided i is such that $\pi(i) \in \mathcal{A}$ or $\hat{\pi}(i) \in \mathcal{A}$, since the same argument applies to $\hat{\pi}$.

Case 2. $\pi(i) \in X$

Taking the latter remark into account we directly get $\hat{\pi}(i) \in X$, too. Hence, $\pi(i) \neq \hat{\pi}(i)$ implies that there are x_j, x_z, $j \neq z$ such that $\pi(i) = x_j$ and $\hat{\pi}(i) = x_z$. Without loss of generality, we may assume $j < z$. Then there exists a position $p < i$ such that $\hat{\pi}(p) = x_j$, since the *LWA* exclusively outputs canonical patterns. Therefore, by the choice of i we can conclude $\pi(p) = x_j$, too. Furthermore, let $\pi_0, \ldots, \pi_m$ be the sequence of hypotheses produced by the *LWA* when successively fed σ. Then we denote by r the least $\tilde{r} \in \{0, \ldots, m\}$ such that $\pi_{\tilde{r}-1}(p) \neq x_j$ and $\pi_{\tilde{r}}(p) = x_j$. Consequently, $\pi_r(i) = x_j$, too. This is an immediate consequence of the definition of the union operation, since it directly shows that variables distinguished once remain distinguished. Thus, we immediately obtain $w_{r+1}(p) = w_{r+1}(i), \ldots,$ $w_m(p) = w_m(i)$, since otherwise $\pi(p) \neq \pi(i)$. Hence, it remains to consider $w_0, \ldots, w_r$.

Case 2.1. $\pi_{r-1}(p) = a \in \mathcal{A}$

In this case we can further conclude that $w_0(p) = \cdots = w_{r-1}(p) = a$. Moreover, we also have $\pi_{r-1}(i) = b \in \mathcal{A}$, since otherwise $\pi_r(p) \neq \pi_r(i)$. Consequently, $w_0(i) = \cdots = w_{r-1}(i) = b$. Moreover, $w_r(p) \neq a$ and $w_r(i) \neq b$, since $\pi_r(p) \in X$. On the other hand, $\pi_r(p) = \pi_r(i)$, and thus $a = b$. To see this, suppose the converse, i.e., $a \neq b$. As we have seen $w_r(p)$, $w_r(i) \notin \{a, b\}$. But then $\pi_r(p) \neq \pi_r(i)$, by the definition of the union operation. Finally, $a = b$ immediately implies $w_r(p) = w_r(i) \neq a$, since otherwise again $\pi_r(p) \neq \pi_r(i)$. This proves $w(p) = w(i)$ for all $w \in range(\sigma)$. Now, an easy inductive argument directly yields $\hat{\pi}(p) = \hat{\pi}(i)$, a contradiction.

Case 2.2. $\pi_{r-1}(p) = \hat{x} \in X$

Again, one easily verifies $\pi_{r-1}(i) = \hat{x}$. Analogously as above one can go back to the first hypothesis $r' < r$ that contains for the first time at position p the variable $\hat{x}$. Therefore, the same arguments apply. In case $\pi_{r'}(p) \in \mathcal{A}$ we are done as above. Otherwise, we iterate the argument *mutatis mutandis*. Since $\pi_0(p) \in \mathcal{A}$, the modified Subcase 2.1. must eventually happen. □

The proof of the latter theorem directly implies the following corollary.

COROLLARY 3

Let $k \in \mathbb{N}^+$, let $\pi \in Pat_k$ be arbitrarily fixed, and let $S \subseteq L(\pi)_{min}$ be any good sample of size m. Furthermore, let $t \in Text(L(\pi))$ and $x \in \mathbb{N}$ such that $S \subseteq t_x^+$. Then the LWA converges to π when successively fed t_x.

Next, we present a lemma that helps to keep the subsequent proofs technically simpler.

LEMMA 4

Let $k \in \mathbb{N}^+$, let $\pi \in Pat_k$ be arbitrarily fixed, and let $\rho = x_0 \ldots x_{k-1}$. Moreover, let $S = \{(u_0, \ldots, u_{k-1}) \mid u_i \in \mathcal{A},\ i = 0, \ldots, k-1\}$ be any set of shortest substitutions. Then we have: The LWA converges on $S(\rho)$ to ρ if and only if it converges on $S(\pi)$ to π.

Proof

First of all note that any set S of shortest substitutions contains at most $|\mathcal{A}|^k$ many elements, i.e., S is finite. Moreover, by Lemma 1 we additionally know that $|S| = |S(\pi)|$ for every $\pi \in Pat_k$. Furthermore, it is easy to see that $S(\rho) = S$. By Theorem 2 we know that the *LWA* is set-driven. Hence, the union operation defined above canonically extends to sets of strings. Now, assume $\cup S = \rho$. We have to show that $\cup S(\pi) = \pi$. Let $\hat{\pi} = \cup S(\pi)$; then $|\hat{\pi}| = |\pi|$, since S is a set of shortest substitutions. Suppose there exists an $n \in \{1, \ldots, |\pi|\}$ such that $\hat{\pi}(n) \neq \pi(n)$. Let i be the least number n satisfying $\hat{\pi}(n) \neq \pi(n)$. Taking into account that $w(i) = \pi(i)$ for all $w \in S(\pi)$ provided $\pi(i)$ is a constant, by the definition of the union operation, we may directly conclude that $\hat{\pi}(i) \neq \pi(i)$ can only happen if $\pi(i) \notin \mathcal{A}$.

Claim 1. $\hat{\pi}(i) \notin \mathcal{A}$

Suppose the converse, i.e., $\hat{\pi}(i) = a \in \mathcal{A}$. By the definition of the union operation this can happen if and only if $a = \pi[x_0\colon u_0, \ldots, x_{k-1}\colon u_{k-1}](i)$ for all substitutions $(u_0, \ldots, u_{k-1}) \in S$. Furthermore, since $\pi(i) \in X$, say $\pi(i) = x_j$ for some $j \in \{0, \ldots, k-1\}$, we may immediately conclude that $u_j = a$ for all substitutions $(u_0, \ldots, u_{k-1}) \in S$. Thus, $\cup S(j) = a$ in accordance with the definition of the union operation; a contradiction to $\cup S = \rho$. This proves Claim 1.

Consequently, $\hat{\pi}(i) \in X$, too. Moreover, by Lemma 2 of Lange and Wiehagen [12], we furthermore know that

(α) $\hat{\pi}$ is a canonical pattern, and

(β) $\#var(\hat{\pi}) < \#var(\pi)$.

Let $\hat{\pi}(i) = x_m$ and $\pi(i) = x_j$. Then, (α) and (β) imply $m < j$, since π is also a canonical pattern. Moreover, there must be an $\ell < i$ such that $\hat{\pi}(\ell) = x_m$, too. Furthermore, since i is the least number n satisfying $\hat{\pi}(n) \neq \pi(n)$, we additionally have $\pi(\ell) = x_m$. Again, taking the definition of the union operation into account, one can easily prove that $u_m = u_j$ for all substitutions $(u_0, \dots, u_{k-1}) \in S$. However, this would directly imply $\cup S(m) = \cup S(j)$; a contradiction to $\cup S = \rho$.

The converse direction can be proved *mutatis mutandis*, and is thus omitted. □

By Lemma 4, whenever dealing with the number of strings from $L(\pi)_{min}$, $\pi \in Pat_k$, that are necessary and sufficient, respectively, for the *LWA* to converge, it suffices to consider exclusively the pattern $\rho = x_0 \dots x_{k-1}$. The next theorem establishes a lower bound for the number of examples from $L(\pi)_{min}$ needed by the *LWA* to converge. This number exclusively depends the number k of different variables occurring in the target pattern π and on the alphabet size $|\mathcal{A}|$.

THEOREM 5

Let $k \in \mathbb{N}^+$, let $\pi \in Pat_k$, and let $|\mathcal{A}| \geq 2$. Then, at least $\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1) \rfloor + 1$ examples from $L(\pi)_{min}$ are necessary in order to achieve convergence of the LWA.

Proof

By Lemma 4 it suffices to consider the target pattern $\rho = x_0 \dots x_{k-1}$, only. Now, given m shortest substitutions $(u_0^1, \dots, u_{k-1}^1), \dots, (u_0^m, \dots, u_{k-1}^m)$, we may write them in a table having m rows and k columns as follows:

	x_0	$\dots$	x_{k-1}
1	u_0^1	$\dots$	u_{k-1}^1
2	u_0^2	$\dots$	u_{k-1}^2
.	.	$\dots$	.
.	.	$\dots$	.
.	.	$\dots$	.
m	u_0^m	$\dots$	u_{k-1}^m

As the proof of Lemma 4 shows, in order to achieve convergence it is necessary that all columns are pairwise different and that there is no constant column, i.e., no column j such that $u_j^1 = \dots = u_j^m$. Now, there are $\mathcal{N} = (|\mathcal{A}|^m - |\mathcal{A}|)(|\mathcal{A}|^m - (|\mathcal{A}| + 1)) \cdot \dots \cdot (|\mathcal{A}|^m - (|\mathcal{A}| + k - 1))$ possibilities for k such columns of length m. Hence, the minimal m is determined by the condition $\mathcal{N} \neq 0$. This condition is equivalent to $|\mathcal{A}|^m - (|\mathcal{A}| + k - 1) > 0$. Thus, we obtain $m > \lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1) \rfloor$. Consequently, at least $\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1) \rfloor + 1$ examples from $L(\pi)_{min}$ are necessary in order to achieve convergence of the *LWA*. □

At this point, it is only natural to ask whether or not the lower bound established by Theorem 5 is tight. The answer to this question is also of particular importance for the average-case analysis to be performed later. The affirmative answer is provided by our next theorem, and we thus establish the announced improvement compared with Marron's [17] bound of $k + 1$.

THEOREM 6

Let $k \in \mathbb{N}^+$, let $\pi \in Pat_k$, and let $|\mathcal{A}| \geq 2$. Then, there always exists a set S of $\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1) \rfloor + 1$ examples from $L(\pi)_{min}$ such that $\cup S = \pi$.

Proof

Let $k \in \mathbb{N}^+$, $\pi \in Pat_k$, let $|\mathcal{A}| \geq 2$, and set $m = \lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1) \rfloor + 1$. We have to construct a set S of m examples from $L(\pi)_{min}$ such that $\cup S = \pi$. Again, by Lemma 4 it suffices to construct a set S of shortest substitutions having cardinality m such that $\cup S = \rho^k$, where $\rho^k = x_0 \dots x_{k-1}$. Moreover, by Theorem 2 we are done, if we could prove that the IIM realizing the *LWA* converges to ρ^k when fed the examples of S in a particular order.

Let $n = |\mathcal{A}| \geq 2$, and let $a_0, \dots, a_{n-1}$ denote the elements of $\mathcal{A}$. Clearly, the hardest cases occur for $k = |\mathcal{A}|^m - |\mathcal{A}|$, $m = 2, 3, \dots$. Next, we inductively describe how the wanted m examples can be constructed.

We start with $m = 2$. Hence, $k = |\mathcal{A}|^2 - |\mathcal{A}| = |\mathcal{A}|(|\mathcal{A}| - 1)$. The first example $u_1 = (u_0^1, \dots, u_{k-1}^1)$ is obtained by setting $u_j^1 = a_{j \bmod |\mathcal{A}|}$ for $j = 0, \dots, k-1$. The second example $u_2 = (u_0^2, \dots, u_{k-1}^2)$ is constructed as follows. We just take the $|\mathcal{A}| - 1$ many cyclical shifts of $a_0, \dots, a_{n-1}$ that are different from $a_0, \dots, a_{n-1}$ and write them one behind the other, i.e., $u_2 = (a_1, \dots, a_{n-1}, a_0, \dots, a_{n-1}, a_0, \dots, a_{n-2})$. Now, it is easy to see that in the computation of $u_1 \cup u_2$ always the "otherwise" case happens, i.e., $u_1 \cup u_2 = x_0, \dots, x_{k-1} = \rho^k$ (cf. Figure 1 for the $\mathcal{A} = \{0, 1, 2, 3\}$ case).

u_1	0	1	2	3	0	1	2	3	0	1	2	3
u_2	1	2	3	0	2	3	0	1	3	0	1	2
$u_1 \cup u_2$	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}

Figure 1

We proceed inductively over m. Hence, we assume that for $k = |\mathcal{A}|^m - |\mathcal{A}|$ there is a set $S_m = \{u_1, \dots, u_m\}$ of m shortest substitutions such that $\cup S = \rho^k$. Now, let $k_{ind} = |\mathcal{A}|^{m+1} - |\mathcal{A}|$. The desired $m + 1$ examples are constructed as follows. First, we take into account that $|\mathcal{A}|^{m+1} - |\mathcal{A}| = |\mathcal{A}||\mathcal{A}|^m - |\mathcal{A}| = (|\mathcal{A}| - 1)|\mathcal{A}|^m + |\mathcal{A}|^m - |\mathcal{A}| = (|\mathcal{A}| - 1)|\mathcal{A}|^m + k$. In order to simplify notation, we set $\ell = (|\mathcal{A}| - 1)|\mathcal{A}|^m$. The first example v^1 is again defined to be $v_j^1 = a_{j \bmod |\mathcal{A}|}$ for $j = 0, \dots, k_{ind} - 1$. However, for the remaining m examples we clearly aim to apply the induction hypothesis. Therefore, we distinguish between the first ℓ positions of the shortest substitutions to be defined and the remaining k ones. The k rightmost positions of $v_2, \dots, v_{m+1}$ are defined to be $u_1, \dots, u_m$, respectively. Furthermore, the leftmost ℓ positions of $v_2, \dots, v_{m+1}$ are defined as follows:

Observing that $\ell = (|\mathcal{A}| - 1)|\mathcal{A}||\mathcal{A}|^{m-1}$, we define the leftmost $|\mathcal{A}|(|\mathcal{A}| - 1)$ positions of v_2 to be the $|\mathcal{A}| - 1$ many cyclical shifts of $a_0, \dots, a_{n-1}$ that are different from $a_0, \dots, a_{n-1}$ written one behind the other. Furthermore, the remaining positions are just defined by repeating the block of the leftmost $|\mathcal{A}|(|\mathcal{A}| - 1)$ positions of v_2 just $|\mathcal{A}|^{m-1} - 1$ many times. That is,

$$v_2 =$$

$$\underbrace{\underbrace{(a_1, \dots, a_{n-1}, a_0,}_{\text{the first cyclical shift}} \dots, \underbrace{a_{n-1}, a_0, \dots, a_{n-2},}_{\text{the } (|\mathcal{A}|-1)\text{th cyclical shift}}}_{\text{the leftmost block of length } |\mathcal{A}|(|\mathcal{A}|-1)} \underbrace{\underbrace{a_1, \dots, a_{n-1}, a_0,}_{\text{the first cyclical shift}} \dots, \underbrace{a_{n-1}, a_0, \dots, a_{n-2},}_{\text{the } (|\mathcal{A}|-1)\text{th cyclical shift}}}_{\text{the second block of length } |\mathcal{A}|(|\mathcal{A}|-1)}$$

$$, \dots, \underbrace{a_1, \dots, a_{n-1}, a_0, \dots, a_{n-1}, a_0, \dots, a_{n-2},}_{\substack{\text{the } |\mathcal{A}|^{m-1}\text{th repetition} \\ \text{of the block of all cyclical shifts}}} \underbrace{u_0^1, \dots, u_{k-1}^1}_{\substack{\text{the } k \text{ rightmost} \\ \text{positions}}})$$

Next, we define v_3 as follows. The leftmost $|\mathcal{A}|(|\mathcal{A}| - 1)$ positions of v_3 are set to be equal to a_0, the next block of length $|\mathcal{A}|(|\mathcal{A}| - 1)$ is set to be equal to a_1, ..., the $|\mathcal{A}|$th block of length

$|\mathcal{A}|(|\mathcal{A}|-1)$ is set to be equal to a_{n-1}. This defines a block of length $(|\mathcal{A}|-1)|\mathcal{A}||\mathcal{A}|$, i.e., if $m=2$ we are done. If $m>2$, we fill the remaining $\ell-(|\mathcal{A}|-1)|\mathcal{A}|^2$ positions by just repeating this block $|\mathcal{A}|^{m-2}-1$ times. That is, let $z=|\mathcal{A}|(|\mathcal{A}|-1)$, then

$$v_3 =$$

$$(\underbrace{\underbrace{a_0,\ldots,a_0,}_{\substack{\text{the first}\\ z\text{ positions}}}\ldots,\underbrace{a_{n-1},\ldots,a_{n-1},}_{\substack{\text{the }|\mathcal{A}|\text{th block}\\ \text{of length }z}}}_{\text{the first block of length }(|\mathcal{A}|-1)|\mathcal{A}|^2}\ldots,\underbrace{a_0,\ldots,a_0,\ldots,a_{n-1},\ldots,a_{n-1},}_{\substack{\text{the }|\mathcal{A}|^{m-2}\text{ block}\\ \text{of length }(|\mathcal{A}|-1)|\mathcal{A}|^2}}\underbrace{u_0^2,\ldots,u_{k-1}^2}_{\substack{\text{the }k\text{ rightmost}\\ \text{positions}}}))$$

Subsequently, $v_4,\ \ldots,\ v_{m+1}$ are analogously defined as v_3. The only difference consists in augmenting the number of repetitions of $a_0,\ldots,a_{n-2}$, and a_{n-1}, respectively, each time by the factor $\mathcal{A}$. Figure 2 displays the corresponding examples and hypotheses for the case $\mathcal{A}=\{0,1\}$, $k=30$, and $m=5$. The vertical line in the table at position $\ell=16$ has been drawn to clearly separate the recursively handled part.

v_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
v_2	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1
π_1	x_0	x_1	x_0	x_1	x_0	x_1	x_0	x_1	x_0	x_1	x_0	x_1	x_0	x_1	x_0	x_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
v_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	1	0	1	0	1	0	1	0	0	1	0	1	0	1
π_2	x_0	x_1	x_2	x_3	x_0	x_1	x_2	x_3	x_0	x_1	x_2	x_3	x_0	x_1	x_2	x_3	x_5	x_6	x_5	x_6	x_5	x_6	x_5	x_6	0	1	0	1	0	1
v_4	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0	1	1	0	0	1	1	1	0	1	0	0	1
π_3	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{12}	x_{13}	0	1
v_5	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1	0	0	1	1	1	0
π_4	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}	x_{21}	x_{22}	x_{23}	x_{24}	x_{25}	x_{26}	x_{27}	x_{28}	x_{29}

Figure 2

Finally, in accordance with our construction it is easy to verify that the first two examples force the *LWA* to introduce $|\mathcal{A}|(|\mathcal{A}|-1)$ variables. Subsequently, each example augments the number of variables occurring in the ℓ leftmost positions by the factor $|\mathcal{A}|$. Moreover, by the definition of our examples, one easily verifies that the variables introduced in the k rightmost positions must have different names than those ones introduced in the ℓ leftmost positions. Hence, applying the induction hypothesis we are done. This proves the theorem for the hardest cases.

The remaining cases are handled *mutatis mutandis.* Suppose $|\mathcal{A}|^{m-1}-|\mathcal{A}|<k<|\mathcal{A}|^m-|\mathcal{A}|$. Then, we perform the same construction as in the $k=|\mathcal{A}|^m-|\mathcal{A}|$ case, except that in the rightmost part of the examples the positions not needed are deleted. □

Now we are ready to characterize the best-case and worst-case behavior of the *LWA*. This is done by the next theorem.

THEOREM 7

Let $k\in\mathbb{N}^+$, and let $|\mathcal{A}|\geq 2$. Then we have:

(1) *For every pattern $\pi\in Pat_k$ the LWA needs in the best-case simultaneously total learning time $O(|\pi|^2\log_{|\mathcal{A}|}(|\mathcal{A}|+k))$ and space $O(|\pi|)$ in order to infer the language $L(\pi)$.*

(2) *For every pattern $\pi\in Pat_k$ and every $n\in\mathbb{N}$ there exists a text $t\in Text(L(\pi))$ such that simultaneously $TT(M,t)>n$ and the space needed by the LWA to learn the language $L(\pi)$ exceeds n, i.e., the worst-case total learning time and the worst-case space complexity of the LWA are unbounded.*

Proof

As we have seen, at least $\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}|+k-1)\rfloor+1$ examples are always necessary and in the best case sufficient to learn every pattern $\pi \in Pat_k$. Hence, in the best-case the *LWA* has to perform $\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}|+k-1)\rfloor+1$ union operations over strings from $L(\pi)_{min}$. Each of them costs at most $O(|\pi|^2)$ time. Therefore, for every text $t \in Text(L(\pi))$ starting with strings obtained by the substitutions presented in the proof of Theorem 6 we have $TT(L(\pi),t) = O(|\pi|^2 \log_{|\mathcal{A}|}(|\mathcal{A}|+k))$. Moreover, the algorithm has to store exclusively its last hypothesis and the new string fed in order to compute its actual guess. Thus, the overall space complexity is $O(|\pi|)$. This proves (1).

Let $n \in \mathbb{N}$ be arbitrarily fixed. Then, for every text $t \in Text(L(\pi))$ starting with a string $w_0 \in L(\pi)$ such that $|s| > n$ we already exceed the space bound n. Moreover, if t continues with a string w_1 satisfying $|w_0| = |w_1|$, then the *LWA* has to compute $w_0 \cup w_1$. Hence, this computation already exceeds the time bound n. Consequently, $TT(L(\pi),t) > n$. □

The latter theorem offers already some insight into the complexity behavior of the *LWA* with respect to the total learning time and the amount of space needed by the *LWA*. However, there is a giant gap between the best-case and worst-case behavior. Therefore, it is of particular interest to analyze the average-case behavior of the LWA. This is done in the next section.

4. Average-Case Analysis of the *LWA*

In this section we study the average case behavior of the *LWA*. Since we want to compute the average total learning time, we start with a closer look at it. Let $k \in \mathbb{N}^+$, $\pi \in Pat_k$ be any pattern, and let $t = (w_n)_{n\in\mathbb{N}}$ range over all randomly generated texts with respect to some admissible distribution for Pat_k. Then we want to compute $E(TT(M,(w_n)))$. By definition, $TT(M,(w_n)) = \sum_{n=0}^{Conv(M,t)} T_M(h_{n-1},w_n)$, since the *LWA* is iterative. However, the expectation of $TT(M,(w_n))$ is *not* just the sum of $E(T_M(h_{n-1},w_n))$, since $Conv(M,t)$ is itself a random variable. Therefore, we first derive a formula to estimate $E(TT(M,(w_n)))$. For simplifying notation, we use C to denote the random variable $Conv(M,t)$. Clearly, C takes only natural numbers as its values.

THEOREM 8

Let $k \in \mathbb{N}^+$, let $\pi \in Pat_k$ be any pattern, and let $t = (w_n)_{n\in\mathbb{N}}$ range over all randomly generated texts $t \in Text(L(\pi))$ with respect to some admissible distribution for Pat_k. Then the expectation of $TT(M,(w_n))$ can be estimated as follows:

$$E(TT(M,(w_n))) = O\big(E(C)(V(|w_0|) + E^2(|w_0|))\big) \tag{6}$$

Proof

For the sake of presentation, we set $X = TT(M,(w_n))$. Next we apply Formula (5) to deduce the pgf for X. Hence, we may write

$$G_X(z) = \sum_{c\geq 0} Pr(C=c)\cdot g_{X|c}(z)$$

where

$$g_{X|c}(z) = \sum_{\nu\geq 0} Pr(X|c=\nu)z^\nu = \sum_{\nu\geq 0} Pr\Big(\sum_{n=0}^{c} T_M(h_{n-1},w_n) = \nu\Big)z^\nu$$

Moreover, in accordance with (3) we know that $E(X) = G'_X(1)$ provided $G'_X(z)$ converges for $z = 1$. Furthermore,

$$G'_X(1) = \sum_{c \geq 0} Pr(C = c) \cdot g'_{X|c}(1)$$

Thus, we next compute $g'_{X|c}(1)$.

$$g'_{X|c}(z) = \sum_{\nu \geq 0} \nu \cdot Pr(X|c = \nu) z^{\nu - 1}$$

and hence

$$\begin{aligned} g'_{X|c}(1) &= \sum_{\nu \geq 0} \nu \cdot Pr(X|c = \nu) = \sum_{\nu \geq 0} \nu \cdot Pr\Big(\sum_{n=0}^{c} T_M(h_{n-1}, w_n) = \nu\Big) \\ &= E\Big(\sum_{n=0}^{c} T_M(h_{n-1}, w_n)\Big) = \sum_{n=0}^{c} E(T_M(h_{n-1}, w_n)) \end{aligned}$$

which is obviously convergent provided $E(T_M(h_{n-1}, w_n))$ exists for all $n = 0, \ldots, n$. A closer look at the *LWA* immediately shows that $T_M(h_{-1}, w_0) = |w_0|$. Furthermore, we may use the following obvious worst-case bound: $T_M(h_{n-1}, w_n) = O(\min\{|h_{n-1}|, |w_n|\}^2)$ for all $n > 0$. Therefore, we can easily estimate

$$E(T_M(h_{n-1}, w_n)) = O\big(E(|w_0|^2)\big) = O\big(V(|w_0|) + E^2(|w_0|)\big) \tag{7}$$

If this term is infinite, we are already done, since the statement of the theorem becomes trivial. Assuming $E(T_M(h_{n-1}, w_n))$ to be finite, we can put it all together, and we get:

$$\begin{aligned} E(X) &= G'_X(1) = \sum_{c \geq 0} Pr(C = c) \cdot c \cdot \frac{1}{c} \sum_{n=0}^{c} E(T_M(h_{n-1}, w_n)) \\ &\leq E(C) \cdot \max_{c > 0} \Big\{ \frac{1}{c} \sum_{n=0}^{c} E(T_M(h_{n-1}, w_n)) \Big\} \end{aligned}$$

Next, we estimate the term $\max_{c>0}\{\frac{1}{c}\sum_{n=0}^{c} E(T_M(h_{n-1}, w_n))\}$. Using the Estimate (7), we obviously have

$$\max_{c > 0} \Big\{ \frac{1}{c} \sum_{n=0}^{c} E(T_M(h_{n-1}, w_n)) \Big\} = O(V(|w_0|) + E^2(|w_0|))$$

and hence the theorem is proved. □

Now, Theorem 8 tells us what we have to compute in order to estimate the average total learning time. Namely, we have to determine $E(C)$, i.e., the expectation of the stage of convergence as well as $E(|w_0|)$ and $V(|w_0|)$. This is done distribution independent as long as possible. Subsequently, we consider in particular the uniform distribution and evaluate the derived terms.

In order to analyze $E(|w_0|)$ and $V(|w_0|)$, one can proceed as follows. Let $u = (u_0, \ldots, u_{k-1})$ be any substitution. Because of

$$|\pi[x_0 \colon u_0, \ldots, x_{k-1} \colon u_{k-1}]| = |\pi| + \sum_{i=0}^{k-1} \#_{x_i}(\pi)(|u_i| - 1) \leq |\pi| + |\pi| \sum_{i=0}^{k-1} (|u_i| - 1),$$

we additionally have

$$
\begin{aligned}
E(|\pi[x_0\colon u_0,\dots,x_{k-1}\colon u_{k-1}]|) &\le |\pi| + |\pi| E\Big(\sum_{i=0}^{k-1}(|u_i|) - 1)\Big) \\
V(|\pi[x_0\colon u_0,\dots,x_{k-1}\colon u_{k-1}]|) &\le |\pi|^2 V\Big(\sum_{i=0}^{k-1}(|u_i| - 1)\Big)
\end{aligned}
$$

For the particular interesting case of product distributions the latter formulae further simplify as follows.

$$
E(|\pi[x_0\colon u_0,\dots,x_{k-1}\colon u_{k-1}]|) \le |\pi| + |\pi| \sum_{i=0}^{k-1}(E(|u_i|) - 1) \tag{8}
$$

$$
V(|\pi[x_0\colon u_0,\dots,x_{k-1}\colon u_{k-1}]|) \le |\pi|^2 \sum_{i=0}^{k-1} V(|u_i|) \tag{9}
$$

Consequently, for the application of (8) and (9) it suffices to study the pgfs $G_{|u_i|}$ for the random variables $|u_i|$ ranging over all possible lengths. That is, we have to study

$$
G_{|u_i|}(z) = \sum_{\ell \ge 1} Pr(|u_i| = \ell) z^\ell \tag{10}
$$

However, this study requires additional assumptions concerning the relevant probability distributions. Therefore, we postpone this task until Subsection 4.2.

4.1. ESTIMATING $E(C)$

We continue with the estimation of $E(C)$. By Theorem 5 we already know that for every $\pi \in Pat_k$ at least $\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1)\rfloor + 1$ examples from $L(\pi)_{min}$ are necessary in order to achieve convergence of the *LWA*. Furthermore, Theorem 6 shows that this number is sometimes sufficient, too. On the other hand, one can construct samples $S \subseteq L(\pi)_{min}$ of size $|\mathcal{A}|^{k-1}$ that are *not good*. This can be seen as follows. By Lemma 4 it suffices to consider $\rho = x_0,\dots,x_{k-1}$. As the proof of Theorem 5 shows, in order to achieve convergence it is in particular necessary that the sample S of shortest substitutions does not contain a constant column. However, we may fix the first component of all shortest substitutions in S to be equal to a_0. Since there are precisely $|\mathcal{A}|^{k-1}$ shortest substitutions for $x_1,\dots,x_{k-1}$, the resulting sample of $|\mathcal{A}|^{k-1}$ many shortest substitutions is not good for $\rho = x_0,\dots,x_{k-1}$.

Finally, it is easy to see that every sample of elements from $L(\pi)_{min}$ that has at least size $|\mathcal{A}|^{k-1} + 1$ is good. Consequently, the number of elements from $L(\pi)_{min}$ needed to achieve convergence of the *LWA* may considerably vary. Therefore, it is convenient to introduce another random variable N for this number. As we have seen, N may take as values natural numbers from $\{\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1)\rfloor + 1, \dots, |\mathcal{A}|^{k-1} + 1\}$.

Hence, we may write the pgf for C as follows:

$$
G_C(z) = \sum_{n=\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}|+k-1)\rfloor+1}^{|\mathcal{A}|^{k-1}+1} Pr(N = n) \cdot g_{C|n}(z) \tag{11}
$$

where

$Pr(N = n)$ denotes the probability that precisely n elements from $L(\pi)_{min}$ are needed

$g_{C|n}(z)$ denotes the cpgf for $C|n$, i.e., the pgf for C under the *knowledge* that $N = n$.

Now, it turned out to be convenient to express the cpgf $g_{C|n}(z)$ as follows:

$$g_{C|n}(z) = \sum_{m=1}^{n} g_{T_m}(z) \tag{12}$$

where the functions g_{T_m} have the following meaning:

g_{T_1} describes the probabilities for the appearance of the first string w_1 from $L(\pi)_{min}$ in a randomly generated text.

g_{T_2} describes the *conditional probabilities* in dependence on the possible w_1 for the appearance of the *second* string w_2 from $L(\pi)_{min}$ in a randomly generated text that fulfills $w_1 \neq w_2$.

.

.

.

g_{T_n} describes the *conditional probabilities* in dependence on the possible $w_1, \dots w_{n-1}$ for the appearance of the nth string w_n from $L(\pi)_{min}$ in a randomly generated text that fulfills $w_n \neq w_m$ for all $m = 1, \dots, n-1$.

The random variables T_m themselves refer to the lengths of the corresponding segments in a randomly generated text. That is, T_1 describes the possible lengths of initial segments of a randomly generated text t until the appearance of the first element w_1 from $L(\pi)_{min}$. Moreover, T_2 expresses the possible lengths of the next segment in t until the appearance of an element w_2 from $L(\pi)_{min}$ that is different from w_1. In general T_m describes the possible lengths of the mth segment. The starting point of this segment is determined by the event that already $m-1$ pairwise different strings from $L(\pi)_{min}$ appeared. The end point of the mth segment is defined by the appearance of the mth shortest string w_m from $L(\pi)_{min}$ in the randomly generated text t that is pairwise different to all other strings from $L(\pi)_{min}$ seen so far.

The next theorem shows why the approach undertaken turns out to be useful. In particular, it reduces the estimate of $E(C)$ to the computation of the expected number of elements from $L(\pi)_{min}$ necessary for the *LWA* to converge and to the computation of the expectations for the random variables T_m introduced above.

THEOREM 9

Let $k \in \mathbb{N}^+$, $\pi \in Pat_k$ be any pattern, and let $t = (w_n)_{n\in\mathbb{N}}$ range over all randomly generated texts with respect to some admissible distribution for Pat_k. Then the expectation of the stage of convergence can be estimated as follows:

$$E(C) \leq E(N) \cdot \max\Big\{E(T_1),\ \frac{1}{2}\sum_{j=1}^{2} E(T_j),\ \dots,\ \frac{1}{|\mathcal{A}|^{k-1}+1}\sum_{m=1}^{|\mathcal{A}|^{k-1}+1} E(T_m)\Big\}$$

Proof

In accordance with Formula (3) we obtain from (11)

$$E(C) = G'_C(1) = \sum_{n=\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}|+k-1)\rfloor+1}^{|\mathcal{A}|^{k-1}+1} Pr(N=n) \sum_{m=1}^{n} g'_{T_m}(1)$$

Taking into account that $g'_{T_m}(1) = E(T_m)$, and setting $Pr(N = n) = 0$ for all $n = 1, \ldots, \lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1) \rfloor$, we obtain:

$$\begin{aligned}
E(C) &= \sum_{n=1}^{|\mathcal{A}|^{k-1}+1} Pr(N = n) \sum_{m=1}^{n} E(T_m) \\
&= \sum_{n=1}^{|\mathcal{A}|^{k-1}+1} Pr(N = n) \cdot n \cdot \frac{1}{n} \sum_{m=1}^{n} E(T_m) \\
&\leq \sum_{n=1}^{|\mathcal{A}|^{k-1}+1} Pr(N = n) \cdot n \cdot \max\Big\{E(T_1),\ \ldots,\ \frac{1}{|\mathcal{A}|^{k-1}+1} \sum_{m=1}^{|\mathcal{A}|^{k-1}+1} E(T_m)\Big\} \\
&= E(N) \cdot \max\Big\{E(T_1),\ \ldots,\ \frac{1}{|\mathcal{A}|^{k-1}+1} \sum_{m=1}^{|\mathcal{A}|^{k-1}+1} E(T_m)\Big\}
\end{aligned}$$

□

Next, we are going to derive formulae for the cpgf g_{T_m}. Again, we perform the wanted derivation in dependence on the number k of different variables in the target pattern π. Moreover, by Lemma 1 it suffices to deal with the probabilities of the shortest substitutions. Let $\mathcal{A} = \{0, 1, \ldots, a-1\}$. Then, we use b_i to denote the shortest substitution $(b_i^0, \ldots, b_i^{k-1})$, where $b_i^j \in \mathcal{A}$, $j = 0, \ldots, k-1$, and $i = b_i^0 \ldots b_i^{k-1}$. That is, i is expressed as a–ary number including leading zeros. For example, for $\mathcal{A} = \{0, 1, \ldots, 9\}$ and $k = 4$ we have $b_0 = (0, 0, 0, 0)$, and $b_{9999} = (9, 9, 9, 9)$. Now, let D be any fixed probability distribution. Then, $p = \sum_{i=0}^{|\mathcal{A}|^k-1} D(b_i)$ is clearly the probability of success for the first shortest substitution. Hence, we obtain:

$$\begin{aligned}
g_{T_1}(z) &= \sum_{\nu \geq 0} Pr(T_1 = \nu) z^\nu = \sum_{\nu \geq 1} (1-p)^{\nu-1} p z^\nu \\
&= \frac{pz}{1-(1-p)z}
\end{aligned}$$

Consequently, by Formula (3)

$$E(T_1) = g'_{T_1}(1) = \frac{1}{p} \tag{13}$$

This was quiet easily done. However, the derivation of expressions for the remaining g_{T_m} again involves conditional probabilities. For the sake of presentation, we first handle the case $m = 2$, and show subsequently how to generalize it. We use Formula (5) and express the pgf for T_2 as follows:

$$g_{T_2}(z) = \sum_{b_i \in \mathcal{A}^k} Pr(Y = b_i) g_{T_2|b_i}(z) \tag{14}$$

where

$$g_{T_2|b_i}(z) = \sum_{\nu \geq 1} \underbrace{\big(1 - p + D(b_i)\big)}_{\substack{\text{failure probability} \\ \text{increases}}}{}^{\nu-1} \underbrace{\big(p - D(b_i)\big)}_{\substack{\text{success prob.} \\ \text{decreases}}} z^\nu \tag{15}$$

$$= \frac{(p - D(b_i))z}{1 - (1 - p + D(b_i))z} \tag{16}$$

It remains to compute $Pr(Y = b_i)$. This is done by Bayes' Theorem. Let $H_i = \{b_i\}$, i.e., H_i is the hypothesis that the first shortest element w_1 from $L(\pi)_{min}$ seen so far has been generated by the shortest substitution b_i. Setting $B = \bigcup_{j=0}^{|\mathcal{A}|^k-1} H_j$ the probability $Pr(Y = b_i)$ is clearly equal to $Pr(H_i|B)$. Furthermore, the *a posteriori* probabilities $Pr(H_i|B)$ are obtained as follows:

$$Pr(Y = b_i) = Pr(H_i|B) = \frac{Pr(B|H_i)Pr(H_i)}{\sum\limits_{j=0}^{|\mathcal{A}|^k-1} Pr(B|H_j)Pr(H_j)} \tag{17}$$

Now, taking into account that $Pr(H_i) = D(b_i)$ and that $Pr(B|H_i) = \dfrac{Pr(B \cap H_i)}{Pr(H_i)} = 1$ for all $i \in \{0, \dots, |\mathcal{A}|^k - 1\}$, Equation (17) simplifies to

$$Pr(Y = b_i) = \frac{D(b_i)}{\sum\limits_{j=0}^{|\mathcal{A}|^k-1} D(b_j)} = \frac{D(b_i)}{p} \tag{18}$$

Incorporating (16) and (18) into (14) and applying again (3) we obtain:

$$E(T_2) = g'_{T_2}(1) = \frac{1}{p} \cdot \sum_{b_i \in \mathcal{A}^k} \frac{D(b_i)}{p - D(b_i)} \tag{19}$$

Now, it is not hard to see how to generalize the latter derivation. Let $b_{i_1}, \dots, b_{i_{m-1}}$ denote the shortest substitutions that generated the $m-1$ pairwise different strings $w_1, \dots, w_{m-1}$ from $L(\pi)_{min}$ already seen. Then, Equations (14), (15) and (16) generalize as follows:

$$g_{T_m}(z) = \sum_{\substack{(b_{i_1},\dots,b_{i_{m-1}}) \in (\mathcal{A}^k)^{m-1} \\ b_{i_\ell} \neq b_{i_j},\ \ell \neq j}} Pr(Y = (b_{i_1}, \dots, b_{i_{m-1}}))\, g_{T_m|(b_{i_1},\dots,b_{i_{m-1}})}(z) \tag{20}$$

where

$$g_{T_m|(b_{i_1},\dots,b_{i_{m-1}})}(z) = \sum_{\nu \geq 1} \underbrace{\Big(1 - p + \sum_{j=1}^{m-1} D(b_{i_j})\Big)}_{\substack{\text{failure probability} \\ \text{increases}}}{}^{\nu-1} \underbrace{\Big(p - \sum_{j=1}^{m-1} D(b_{i_j})\Big)}_{\substack{\text{success prob.} \\ \text{decreases}}} z^{\nu} \tag{21}$$

$$= \frac{(p - \sum\limits_{j=1}^{m-1} D(b_{i_j}))z}{1 - (1 - p + \sum\limits_{j=1}^{m-1} D(b_{i_j}))z} \tag{22}$$

For computing the probabilities $Pr(Y = (b_{i_1}, \dots, b_{i_{m-1}}))$ we again apply Bayes' Theorem. We set $H_{(i_1,\dots,i_{m-1})} = \{(b_{i_1}, \dots, b_{i_{m-1}})\}$ for all tuples $(b_{i_1}, \dots, b_{i_{m-1}}) \in (\mathcal{A}^k)^{m-1}$ satisfying $b_{i_\ell} \neq b_{i_j}$ for all $\ell, j \in \{1, \dots, m-1\}$, $\ell \neq j$. The set B is again the union of all hypotheses $H_{(i_1,\dots,i_{m-1})}$. Furthermore, $Pr(H_{(i_1,\dots,i_{m-1})}) = \prod_{j=1}^{m-1} D(b_{i_j})$, since all substitutions are drawn independently. Finally, taking into account that $Pr(B|H_{(i_1,\dots,i_{m-1})}) = 1$, we obtain

$$Pr(H_{(i_1,\dots,i_{m-1})}|B) = \frac{\prod\limits_{j=1}^{m-1} D(b_{i_j})}{\sum\limits_{\substack{(b_{j_1},\dots,b_{j_{m-1}}) \in (\mathcal{A}^k)^{m-1} \\ b_{j_\ell} \neq b_{j_i},\ \ell \neq i}} \prod\limits_{z=1}^{m-1} D(b_{j_z})} \tag{23}$$

Finally, incorporating (22) and (23) into (20) and applying again (3) we obtain:

$$E(T_m) = \frac{1}{\sum\limits_{\substack{(b_{j_1},\ldots,b_{j_{m-1}})\in(\mathcal{A}^k)^{m-1}\\ b_{j_\ell}\neq b_{j_i},\ \ell\neq i}} \prod\limits_{z=1}^{m-1} D(b_{j_z})} \cdot \sum_{\substack{(b_{i_1},\ldots,b_{i_{m-1}})\in(\mathcal{A}^k)^{m-1}\\ b_{i_\ell}\neq b_{i_j},\ \ell\neq j}} \frac{\prod\limits_{j=1}^{m-1} D(b_{i_j})}{p - \sum\limits_{j=1}^{m-1} D(b_{i_j})} \tag{24}$$

The latter formula directly allows the derivation of lower and upper bounds for $E(T_m)$. Let $b_{min_1},\ldots,b_{min_m}$ denote the shortest substitutions that satisfy $D(b_{min_1}) = \min\{D(b_i) \mid b_i \in \mathcal{A}^k\},\ldots,D(b_{min_m}) = \min\{D(b_i) \mid b_i \in \mathcal{A}^k\backslash\{b_{min_1},\ldots,b_{min_{m-1}}\}$, respectively. Furthermore, we analogously define $b_{max_1},\ldots,b_{max_m}$ by replacing "min" by "max." Then we have the following corollary.

COROLLARY 10

For all $m \in \mathbb{N}$, $m \geq 2$, the expectation of T_m can be estimated as follows:

$$\frac{1}{p - \sum\limits_{j=1}^{m-1} D(b_{min_j})} \ \leq\ E(T_m) \ \leq\ \frac{1}{p - \sum\limits_{j=1}^{m-1} D(b_{max_j})}$$

Proof

By (24) we have:

$$\begin{aligned}
E(T_m) &= \frac{1}{\sum\limits_{\substack{(b_{j_1},\ldots,b_{j_{m-1}})\in(\mathcal{A}^k)^{m-1}\\ b_{j_\ell}\neq b_{j_i},\ \ell\neq i}} \prod\limits_{z=1}^{m-1} D(b_{j_z})} \cdot \sum_{\substack{(b_{i_1},\ldots,b_{i_{m-1}})\in(\mathcal{A}^k)^{m-1}\\ b_{i_\ell}\neq b_{i_j},\ \ell\neq j}} \frac{\prod\limits_{j=1}^{m-1} D(b_{i_j})}{p - \sum\limits_{j=1}^{m-1} D(b_{i_j})} \\
&\geq \frac{1}{\sum\limits_{\substack{(b_{j_1},\ldots,b_{j_{m-1}})\in(\mathcal{A}^k)^{m-1}\\ b_{j_\ell}\neq b_{j_i},\ \ell\neq i}} \prod\limits_{z=1}^{m-1} D(b_{j_z})} \cdot \sum_{\substack{(b_{i_1},\ldots,b_{i_{m-1}})\in(\mathcal{A}^k)^{m-1}\\ b_{i_\ell}\neq b_{i_j},\ \ell\neq j}} \frac{\prod\limits_{j=1}^{m-1} D(b_{i_j})}{p - \sum\limits_{j=1}^{m-1} D(b_{min_j})} \\
&= \frac{1}{p - \sum\limits_{j=1}^{m-1} D(b_{min_j})} \cdot \frac{\sum\limits_{\substack{(b_{i_1},\ldots,b_{i_{m-1}})\in(\mathcal{A}^k)^{m-1}\\ b_{i_\ell}\neq b_{i_j},\ \ell\neq j}} \prod\limits_{j=1}^{m-1} D(b_{i_j})}{\sum\limits_{\substack{(b_{j_1},\ldots,b_{j_{m-1}})\in(\mathcal{A}^k)^{m-1}\\ b_{j_\ell}\neq b_{j_i},\ \ell\neq i}} \prod\limits_{z=1}^{m-1} D(b_{j_z})} \\
&= \frac{1}{p - \sum\limits_{j=1}^{m-1} D(b_{min_j})}
\end{aligned}$$

The stated upper bound can be analogously proved. □

This finishes the distribution independent estimate of $E(C)$. Clearly, in order to arrive at better interpretable estimates of $E(C)$ one has to evaluate $E(T_1),\dots,E(T_n)$ as well as $E(N)$ for particular distributions. This is done in the next subsection.

4.2. RESULTS CONCERNING THE UNIFORM DISTRIBUTION

In this subsection we apply the Theorems 8 and 9 to the uniform distribution. The following theorem expresses the average-case behavior of the *LWA* for this particular case.

THEOREM 11

Let $k \in \mathbb{N}^+$, let $|\mathcal{A}| \geq 2$, let $\pi \in Pat_k$ be any pattern, and let $t = (w_n)_{n\in\mathbb{N}}$ range over all randomly generated texts $t \in Text(L(\pi))$ with respect to the uniform distribution. Then, we have:

$$E(TT(M,(w_n))) = O\big(2^k k^2 |\pi|^2 \log_{|\mathcal{A}|}(k|\mathcal{A}|)\big)$$

Proof

First of all, we deal with the pgfs $G_{|u_i|}$. Since the distribution under consideration is the uniform one, the pgfs $G_{|u_i|}$ are the same for all $i = 0,\dots,k-1$. Taking into account that $Pr(|u_i| = \ell) = |\mathcal{A}|^\ell/(2^\ell|\mathcal{A}|^\ell) = 1/2^\ell$ for all $i \in \{0,\dots,k-1\}$ and $\ell \in \mathbb{N}^+$, we may rewrite Equation (10) as follows

$$G_{|u_i|}(z) = \sum_{\ell\geq 1} \frac{z^\ell}{2^\ell} = \frac{2}{2-z} - 1$$

Hence, by Equations (3) and (4) we obtain:

$$\begin{aligned} E(|u_i|) &= 2 \text{ for all } i = 0,\dots,k-1 \\ V(|u_i|) &= 2 \text{ for all } i = 0,\dots,k-1 \end{aligned}$$

Now, applying (8) and (9) we have $E(|w_0|) \leq (k+1)|\pi|$ and $V(|w_0|) \leq 2k|\pi|^2$, respectively. Therefore, we get:

$$O(V(|w_0|) + E^2(|w_0|)) = O(k^2|\pi|^2) \tag{25}$$

Next, we can directly apply Corollary 10 in order to compute the $E(T_m)$s, since the lower and upper bound stated there clearly match for the uniform distribution.

Since $p = \sum_{i=0}^{|\mathcal{A}|^k-1} 1/(2|\mathcal{A}|)^k = 1/2^k$, by (13) we have $E(T_1) = 2^k$. Furthermore, an easy calculation yields

$$E(T_m) = (2|\mathcal{A}|)^k/(|\mathcal{A}|^k - m + 1). \tag{26}$$

We continue with the evaluation of $\max\Big\{E(T_1),\ \frac{1}{2}\sum_{j=1}^{2} E(T_j),\dots,\frac{1}{|\mathcal{A}|^{k-1}+1}\sum_{m=1}^{|\mathcal{A}|^{k-1}+1} E(T_m)\Big\}$ in order to apply Theorem 9.

Claim 1. $\max\Big\{E(T_1),\ \dots,\ \frac{1}{|\mathcal{A}|^{k-1}+1}\sum_{m=1}^{|\mathcal{A}|^{k-1}+1} E(T_m)\Big\} = \frac{1}{|\mathcal{A}|^{k-1}+1}\sum_{m=1}^{|\mathcal{A}|^{k-1}+1} E(T_m)$

Obviously, it suffices to show that $\frac{1}{n+1}\sum_{m=1}^{n+1} E(T_m) > \frac{1}{n}\sum_{m=1}^{n} E(T_m)$ for all $n \geq 1$. Since

$E(T_{n+1}) > E(T_m)$ for all $m = 1, \dots, n$, we know that $n \cdot E(T_{n+1}) > \sum_{m=1}^{n} E(T_m)$. Therefore,

$$n \cdot \sum_{m=1}^{n} E(T_m) + n \cdot E(T_{n+1}) \quad > \quad n \cdot \sum_{m=1}^{n} E(T_m) + \sum_{m=1}^{n} E(T_m)$$

and hence

$$n \cdot \sum_{m=1}^{n+1} E(T_m) \quad > \quad (n+1) \sum_{m=1}^{n} E(T_m)$$

This proves Claim 1.

Now, it is not hard to estimate the term $\frac{1}{|\mathcal{A}|^{k-1}+1} \sum_{m=1}^{|\mathcal{A}|^{k-1}+1} E(T_m)$. Looking at (26) we see that the biggest summand equals $(2|\mathcal{A}|)^k/(|\mathcal{A}|^k - |\mathcal{A}|^{k-1})$. Thus, we estimate the maximum by replacing all summands by the maximal one. Thus we obtain:

$$\begin{aligned} \frac{1}{|\mathcal{A}|^{k-1}+1} \sum_{m=1}^{|\mathcal{A}|^{k-1}+1} E(T_m) &= \frac{1}{|\mathcal{A}|^{k-1}+1} \sum_{m=1}^{|\mathcal{A}|^{k-1}+1} \frac{2^k |\mathcal{A}|^k}{|\mathcal{A}|^k - m + 1} \\ &< \frac{1}{|\mathcal{A}|^{k-1}+1} \sum_{m=1}^{|\mathcal{A}|^{k-1}+1} \frac{2^k |\mathcal{A}|^k}{|\mathcal{A}|^k - |\mathcal{A}|^{k-1}} \\ &= \frac{1}{|\mathcal{A}|^{k-1}+1} \cdot (|\mathcal{A}|^{k-1}+1) \cdot \frac{2^k |\mathcal{A}|^k}{|\mathcal{A}|^k - |\mathcal{A}|^{k-1}} \\ &= \frac{2^k |\mathcal{A}|}{|\mathcal{A}|-1} \le 2^{k+1} \end{aligned} \tag{27}$$

where the latter estimate is due to $|\mathcal{A}| \ge 2$.

We finish the proof by estimating the expectation of the number of elements from $L(\pi)_{min}$ needed by the *LWA* to converge, i.e., we deal with $E(N)$.

LEMMA 12

Let $k \in \mathbb{N}^+$, and let $\pi \in Pat_k$ be any pattern. Then, the average number of examples from $L(\pi)_{min}$ needed by the LWA to converge is of order $\log_{|\mathcal{A}|}(k \cdot |\mathcal{A}|)$, *i.e.,* $E(N) = O(\log_{|\mathcal{A}|}(k \cdot |\mathcal{A}|))$.

First of all, by Corollary 3 we know that $Pr(N = n)$ equals the ratio of all good samples of size n and all samples $S \subseteq L(\pi)_{min}$ of size n. Moreover, by Lemma 4 it again suffices to deal with $\rho = x_0 \dots x_{k-1}$. Hence, we have to study the probabilities that a randomly chosen subset of n pairwise different shortest substitutions constitutes a good sample of size n. This is done by applying the principle of inclusion and exclusion (cf., e.g., Pólya, Tarjan and Woods [19]). Now, the proof of Lemma 4 shows how to chose the relevant properties. As we have seen, a sample of size n is not good if and only if it contains a constant column or at least two columns of it are identical. Hence, we may define the following properties.

(α) $x_i = const$ for $i = 0, \dots, k-1$,

(β) $x_i = x_j$ for all $i, j \in \{0, \dots, k-1\}$ with $i \ne j$.

Therefore, in total we have $z = k + \binom{k}{2}$ many properties. By N_i we denote the number of samples fulfilling property $i = 0, \dots, z$, by N_{i_1,i_2} we denote the number of samples satisfying simultaneously the properties i_1 and i_2, $i_1 \neq i_2$, and so on. Then, the number of good samples of size n is obtained by

$$N^* = \binom{|\mathcal{A}|^k}{n} - \sum_{i=0}^{z} N_i + \sum_{i_1=1}^{z-1} \sum_{i_2=i_1+1}^{z} N_{i_1,i_2} - \sum_{i_1=1}^{z-2} \sum_{i_2=i_1+1}^{z-1} \sum_{i_3=i_2+1}^{z} N_{i_1,i_2,i_3} + \ldots + (-1)^z N_{0,\ldots,z-1}.$$

Note that $\binom{|\mathcal{A}|^k}{n}$ refers to the number of all possible samples of size n. However, the precise computation of all those numbers $N_{i_1,\ldots i_j}$ is quite complicated. Therefore, we restrict ourselves to calculate the rather rough estimate $N^* \geq \binom{|\mathcal{A}|^k}{n} - \sum\limits_{i=0}^{z} N_i$. In order to simplify notation we set $a = |\mathcal{A}|$.

We continue with the calculation of N_i for $i = 0, \ldots k-1$. If $x_i = const$, then there are $\binom{a^{k-1}}{n}$ possibilities to choose the remaining free positions in the shortest substitutions. Moreover, each resulting sample of shortest substitutions can be varied by choosing a different constant for x_i. Therefore, there we have $N_i = a\binom{a^{k-1}}{n}$. Since there are k possible choices for i, we obtain:

$$\sum_{i=0}^{k-1} N_i = k \cdot a \binom{a^{k-1}}{n} \tag{28}$$

Next, we consider N_i for $i = k, \ldots, z-1$. Let x_i, x_j with $i \neq j$ be arbitrarily fixed. Then there are a^{k-1} many possibilities to choose the values of all $x_0, \ldots, x_{k-1}$ except x_j. Clearly, x_j is already defined by specifying x_i. Hence, there are $\binom{a^{k-1}}{n}$ samples of size n fulfilling $x_i = x_j$. Finally, since there are $\binom{k}{2}$ many choices for pairs x_i, x_j we have:

$$\sum_{i=k}^{z-1} N_i = \binom{k}{2}\binom{a^{k-1}}{n} \tag{29}$$

Putting (28) and (29) together and taking into account that $Pr(N \leq n) \geq N^*/\binom{a^k}{n}$, we obtain the following estimate:

$$Pr(N \leq n) \geq \frac{\binom{a^k}{n} - k \cdot a\binom{a^{k-1}}{n} - \binom{k}{2}\binom{a^{k-1}}{n}}{\binom{a^k}{n}} = 1 - \frac{k \cdot a\binom{a^{k-1}}{n} - \binom{k}{2}\binom{a^{k-1}}{n}}{\binom{a^k}{n}}$$

Now, it suffices to estimate the rightmost term in the latter equation. Applying the definition of the Binomial coefficients and reducing the resulting fraction, we get:

$$\begin{aligned} \frac{k \cdot a\binom{a^{k-1}}{n} - \binom{k}{2}\binom{a^{k-1}}{n}}{\binom{a^k}{n}} &= \frac{\left(k \cdot a + \binom{k}{2}\right)(a^{k-1}-1)\ldots(a^{k-1}-n+1)}{(a^k-1)\ldots(a^k-n+1)} \\ &\leq \frac{k \cdot a + \binom{k}{2}}{a^n} \end{aligned}$$

The latter inequality is easily obtained by applying $(a^{k-1} - \ell)/(a^k - \ell) \leq 1/a$ for all $\ell = 1, \ldots, n-1$. Summarizing, we already know that

$$Pr(N \leq n) \geq 1 - \frac{k \cdot a + \binom{k}{2}}{a^n}$$

Therefore, we directly obtain:

$$Pr(N > n) = 1 - Pr(N \leq n) \leq \frac{k \cdot a + \binom{k}{2}}{a^n} \tag{30}$$

This is nice, since $E(N) = \sum_{n\geq 0} Pr(N > n)$ (cf., e.g., [6]). However, in order to derive the desired bound we have to be careful. That means, as long as the term in (30) is worse than the trivial estimate $Pr(N > n) = 1$, we better sum the 1s. Obviously, $\left(k \cdot a + \binom{k}{2}\right)/a^n \leq 1$ iff $n \geq \lfloor \log_a(k \cdot a + \binom{k}{2})\rfloor + 1$. In order to simplify notion, we set $m = \left\lfloor \log_a \left(k \cdot a + \binom{k}{2}\right)\right\rfloor + 1$. Then, we have:

$$\begin{aligned}
E(N) &= \sum_{n\geq 0} Pr(N > n) \\
&\leq \sum_{n=0}^{m} 1 + \sum_{n\geq m+1} Pr(N > n) \\
&\leq m + 1 + \left(k \cdot a + \binom{k}{2}\right) \sum_{n\geq m+1} \frac{1}{a^n} \\
&= m + 1 + \left(k \cdot a + \binom{k}{2}\right) \left(\sum_{n\geq 0} \frac{1}{a^n} - \sum_{n=0}^{m} \frac{1}{a^n}\right) \\
&= m + 1 + \left(k \cdot a + \binom{k}{2}\right) \left(\frac{a}{a-1}\left(1 - 1 + \frac{1}{a^{m+1}}\right)\right) \\
&\leq m + 1 + \left(k \cdot a + \binom{k}{2}\right) \frac{1}{a-1} \cdot \frac{1}{k \cdot a + \binom{k}{2}} \\
&= m + 1 + \frac{1}{a-1} = \left\lfloor \log_a \left(k \cdot a + \binom{k}{2}\right)\right\rfloor + 2 + \frac{1}{a-1} \\
&= O(\log_a(k \cdot a))
\end{aligned}$$

This proves Lemma 12.

Finally, incorporating Lemma 12 and the Estimation (27) into Theorem 9 as well as (25) into Theorem 8 we directly obtain $E(TT(M, (w_n)) = O(2^k k^2 |\pi|^2 \log_{|\mathcal{A}|}(k|\mathcal{A}|))$ and hence the theorem is proved. □

5. Conclusions and Open Problems

The present paper dealt with the best-case, worst-case and average-case analysis of Lange and Wiehagen's [12] pattern language learning algorithm with respect to its total learning time. As far as we know, this is the first paper that completely analyzes a concrete algorithm that learns a non-trivial class of objects in the limit.

In particular, we proved the matching upper and lower bound of $\lfloor \log_{|\mathcal{A}|}(|\mathcal{A}| + k - 1)\rfloor + 1$ many examples that are necessary and sufficient for the *LWA* to learn any pattern $\pi \in Pat_k$ in the best-case. Note that this number decreases if the alphabet size increases. Moreover, for the uniform distribution the expected number of $O(\log_{|\mathcal{A}|}(k|\mathcal{A}|))$ many examples needed by the *LWA* for converging is not too far from the best-case bound (cf. Lemma 12). Thus, despite the fact that the *LWA* may behave arbitrarily complex in the worst-case, we could establish the bound $O(2^k k^2 |\pi|^2 \log_{|\mathcal{A}|}(k|\mathcal{A}|))$ for its expected total learning time with respect to the uniform distribution. However, this required some effort, and hence the question arises whether or not the work invested has been worth the trouble.

It has been suggested that a reasonable bound on the average-case behavior of the *LWA* can be obtained by using Theorem 6. Assuming again the uniform distribution, one could expect to see the first element of S roughly within the first $|\mathcal{A}|^k$ inputs. Hence, all the elements of S can be expected to have been in the text roughly within the first $|S|\cdot|\mathcal{A}|^k$ inputs. Consequently, the the average-case complexity could be estimated by $O(|w_0|^2|\mathcal{A}|^k|S|) = O(|\pi|^2|\mathcal{A}|^k\log_{|\mathcal{A}|}(|\mathcal{A}|+k))$. This is a nice heuristic argument, and filling in all the remaining details may result in an easier proof. However, the bound obtainable along this line of reasoning is not only worse than the one established in Theorem 11, it is also qualitatively quit different. In particular, the bound $O(|\pi|^2|\mathcal{A}|^k\log_{|\mathcal{A}|}(|\mathcal{A}|+k))$ considerably increases if $|\mathcal{A}|$ increases. In contrast, the bound established in Theorem 11 clearly decreases if the alphabet $\mathcal{A}$ gets larger. Additionally, our average-case analysis has been to a large extent distribution independent, and can thus be easily extended to other interesting distributions.

We continue with a short analysis of the average-case bound obtained. First, if the number of different variables in the target patterns one wants to learn is upper bounded by some constant, then the average-case behavior of the *LWA* is quadratically bounded in $|\pi|$ and logarithmically in the alphabet size $|\mathcal{A}|$. The latter result remains clearly true, if we replace "uniform distribution" by "length biased uniform distribution." As an easy inspection of the proof presented above shows, the only term changing is 2^k to μ_1^{-k}. Therefore, it would be desirable to compare the average-case behavior of the *LWA* to the average-case behavior of other algorithms that learn PAT_k.

Nevertheless, when applied to learn the class of all pattern languages, the expected total learning time is in both cases exponential in the reciprocal value of the relevant weight factor μ_0 assigned to all shortest strings over $\mathcal{A}$. Thus, if k becomes larger the expected total learning time of the *LWA* fastly becomes impractical.

Finally, we discuss further applications of the results obtained. Lange and Wiehagen [12] also considered pattern inference from good examples. In this setting, the teacher provides sets of good examples. However, in order to avoid simple coding tricks, the learner is required to learn from every superset of every set of good examples. Our results apply to this setting as well. Our best-case analysis drastically improves the corresponding assertion concerning the minimal size of sets of good examples (cf. [12], Theorem 3, Assertion (1)).

Moreover, the established tight bound for the size of good samples improves the complexity estimates of other algorithms as well. For example, the number of queries needed in Marron's [17] Algorithm 2.2. also considerably reduces from $k+1$ to $\lfloor\log_{|\mathcal{A}|}(|\mathcal{A}|+k-1)\rfloor+1$. The construction outlined in the proof of Theorem 6 has been also successfully applied in Arimura *et al.* [3] to drastically decrease the number of membership queries in an algorithm that learns unions of tree patterns.

Acknowledgement

The author is very grateful to Takeo Okazaki and Peter Rossmanith for many helpful and enlightening discussions. In particular, I would like to thank Peter Rossmanith for a suggestion in refining the estimate outlined in Theorem 11.

Finally, I heartily thank the anonymous referees for their careful reading and for many valuable comments on the preparation of this paper.

6. References

[1] D. Angluin, Finding patterns common to a set of strings, Journal of Computer and System Sciences 21 (1980) 46 – 62.

[2] D. Angluin, Queries and concept learning, Machine Learning 2 (1988) 319 – 342.

[3] H. Arimura, H. Ishizaka and T. Shinohara, Learning unions of tree patterns using queries, Theoretical Computer Science 185 (1997) 47 – 62.

[4] R. Daley and C.H. Smith, On the complexity of inductive inference, Information and Control 69 (1986) 12 – 40.

[5] E.M. Gold, Language identification in the limit, Information and Control 10 (1967) 447 – 474.

[6] R.L. Graham, D.E. Knuth and O. Patashnik, *Concrete Mathematics* (Addison-Wesley, Reading, Massachusetts, 1989).

[7] J.E. Hopcroft and J.D. Ullman, *Formal Languages and their Relation to Automata* (Addison-Wesley, Reading, Massachusetts, 1969).

[8] T. Jiang, A. Salomaa, K. Salomaa and S. Yu, (1993), Inclusion is undecidable for pattern languages, in: *Proc. 20th International Colloquium on Automata, Languages and Programming*, eds. A. Lingas, R. Karlsson and S. Carlsson (Springer, Lecture Notes in Computer Science 700, 1993) pp. 301 – 312.

[9] M. Kearns and L. Pitt (1989), A polynomial-time algorithm for learning k–variable pattern languages from examples, in: *Proc. 2nd Annual ACM Workshop on Computational Learning Theory*, eds. R. Rivest, D. Haussler and M.K. Warmuth (Morgan Kaufmann Publishers Inc., San Mateo, 1989) pp. 57 – 71.

[10] P. Kilpeläinen, H. Mannila and E. Ukkonen, (1995), MDL Learning of unions of simple pattern languages from positive examples, in: *Proc. 2nd European Conference on Computational Learning Theory – EuroCOLT'95*, ed. P. Vitanyi, (Springer, Lecture Notes in Artificial Intelligence 904, 1995) pp. 252 – 260.

[11] Ker-I Ko, A. Marron and W.G. Tzeng, Learning string patterns and tree patterns from examples, in: *Proc. 7th Conference on Machine Learning*, eds. B.W. Porter and R.J. Mooney (Morgan Kaufmann Publishers Inc., San Mateo, 1990) pp. 384 – 391.

[12] S. Lange and R. Wiehagen, Polynomial-time inference of arbitrary pattern languages, New Generation Computing 8 (1991) 361 – 370.

[13] S. Lange and T. Zeugmann, Types of monotonic language learning and their characterization, in: *Proc. 5th Annual ACM Workshop on Computational Learning Theory*, ed. D. Haussler (ACM Press, New York, 1992) pp. 377 – 390.

[14] S. Lange and T. Zeugmann, Monotonic versus non-monotonic language learning, in: *Proc. 2nd International Workshop on Nonmonotonic and Inductive Logic*, eds. G. Brewka, K.P. Jantke and P.H. Schmitt (Springer, Lecture Notes in Artificial Intelligence 659, 1993) pp. 254 – 269.

[15] S. Lange and T. Zeugmann, Set-driven and rearrangement-independent learning of recursive languages, Mathematical Systems Theory **29**, No. 6, 1996, 599 – 634.

[16] S. Lange and T. Zeugmann, Incremental learning from positive data, Journal of Computer and System Sciences **53**, No. 1, 1996, 88 – 103.

[17] A. Marron, Learning pattern languages from a single initial example and from queries, in: *Proc. 1st Annual ACM Workshop on Computational Learning Theory*, eds. D. Haussler and L. Pitt (Morgan Kaufmann Publishers Inc., San Mateo, 1988) pp. 345 – 358.

[18] R.P. Nix, Editing by examples, Technical Report 280, Department of Computer Science, Yale University, New Haven, USA (1983).

[19] G. Pólya, R.E. Tarjan and D.R. Woods, *Notes on Introductory Combinatorics*, (Birkhäuser, Basel-Boston-Stuttgart, 1983).

[20] A. Salomaa, Patterns (The Formal Language Theory Column), EATCS Bulletin 54 (1994), 46 – 62.

[21] A. Salomaa, Return to patterns (The Formal Language Theory Column), EATCS Bulletin 55 (1994), 144 – 157.

[22] R.E. Schapire, Pattern languages are not learnable, in: *Proc. 3rd Annual ACM Workshop on Computational Learning Theory*, eds. M.A. Fulk and J. Case (Morgan Kaufmann Publishers Inc., San Mateo, 1990) pp. 122 – 129.

[23] S. Shimozono, A. Shinohara, T. Shinohara, S. Miyano, S. Kuhara, and S. Arikawa. Knowledge acquisition from amino acid sequences by machine learning system BONSAI, *Trans. Information Processing Society of Japan* **35** (1994), 2009 – 2018.

[24] T. Shinohara, Polynomial time inference of extended regular pattern languages, in: *Proc. RIMS Symposia on Software Science and Engineering*, eds. E. Goto, K. Furukawa, R. Nakajima, I. Nakata, and A. Yonezawa, (Springer, Lecture Notes in Computer Science 147, 1983) pp. 115 – 127.

[25] T. Shinohara and S. Arikawa, Learning data entry systems: An application of inductive inference of pattern languages, Research Report 102, Research Institute of Fundamental Information Science, Kyushu University, Fukuoka, Japan (1983).

[26] T. Shinohara and S. Arikawa, Pattern inference, *in* “Algorithmic Learning for Knowledge-Based Systems” eds. K.P. Jantke and S. Lange, (Springer, Lecture Notes in Artificial Intelligence 961, 1995) pp. 259 – 291.

[27] K. Wexler and P. Culicover, *Formal Principles of Language Acquisition*, (MIT Press, Cambridge, Massachusetts, 1980).

[28] R. Wiehagen and T. Zeugmann, Ignoring data may be the only way to learn efficiently, Journal of Experimental and Theoretical Artificial Intelligence 6 (1994), 131 – 144.

[29] T. Zeugmann, S. Lange and S. Kapur, Characterizations of monotonic and dual monotonic language learning, Information and Computation 120 (1995), 155 – 173.