

Minimal Markov chain embeddings of pattern problems

Manuel E. Lladser

Department of Applied Mathematics

University of Colorado

Boulder, CO 80309-0526, USA

Email: lladser@colorado.edu

Abstract—The Markov chain embedding technique is commonly used to study the distribution of statistics associated with regular patterns (i.e. set of strings described by a regular expression) in random strings. In this extended abstract, we formalize the concept Markov chain embedding for random strings produced by a possibly non-stationary Markov source. A notion of memory conveyed by the states of a deterministic finite automaton is introduced. This notion is used to characterize the smallest state-space size Markov chain required to specify the distribution of the count statistic of a given regular pattern. The research finds applications in problems associated with regular patterns in random strings that demand exponentially large state spaces.

I. INTRODUCTION

We first introduce some general notation and terminology. In what follows \mathcal{A} denotes a finite set (of characters) called the *alphabet*. The set of all words obtained by concatenating a finite number of characters in \mathcal{A} is denoted \mathcal{A}^* . Elements in \mathcal{A}^* are usually called *strings*. The *length* of a string x is the number of characters that form it and it is denoted $|x|$. By definition the *empty string* (denoted as ϵ) is an element of \mathcal{A}^* and it is the only string of length zero. We define $\mathcal{A}^+ := \mathcal{A}^* \setminus \{\epsilon\}$.

If $x_1, \dots, x_n \in \mathcal{A}^*$ then $x_1 \cdots x_n$ denotes the word obtained by concatenating (from left to right) the strings x_1, \dots, x_n . (For small values of n such as $n = 2$ we write $x_1 x_2$ instead of $x_1 \cdots x_2$.) For $x, y \in \mathcal{A}^*$ we write $x = y \dots$ to mean that there is a string w (possibly empty) such that $x = yw$. Similarly, we write $x = \dots y$ to mean that $x = wy$ for some string w . We say that y is a *substring* of x provided that there are strings w_1 and w_2 (possibly empty) such that $x = w_1 y w_2$.

For the remaining of this manuscript $k \geq 0$ is a fixed integer parameter. We define $\mathcal{A}^k := \mathcal{A}^{\leq k} \cap \mathcal{A}^{\geq k}$, where $\mathcal{A}^{\leq k} := \{x \in \mathcal{A}^* : |x| \leq k\}$ and $\mathcal{A}^{\geq k} := \{x \in \mathcal{A}^* : |x| \geq k\}$.

We are interested in the occurrence and frequency of patterns in random strings. A *pattern* (also called *language*) is by definition is any subset of \mathcal{A}^* . We model a *random string* of length l as $X_1 \cdots X_l$, where $X = (X_n)_{n \geq 1}$ is a sequence of \mathcal{A} -valued random variables defined on a certain probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Much of the literature of patterns in random strings is devoted to the understanding of the distribution of the *sooner-time* and *count-statistics* associated with a given pattern \mathcal{L} . These are the random variables respectively defined

as

$$\tau := \min\{n \geq 1 : X_1 \cdots X_n \in \mathcal{L}\}, \quad (1)$$

$$S_n := \sum_{i=1}^n \mathbb{I}[X_1 \cdots X_i \in \mathcal{L}], \quad (2)$$

for $n \geq 1$. Above $\mathbb{I}[\cdot]$ denotes the Iverson's bracket i.e. a quantity defined to be 1 whenever the statement within the brackets is true, and 0 otherwise.

To fix some ideas regarding the above definitions, let \mathcal{W} be a given pattern (e.g. a finite set of strings) and define $\mathcal{L} = \mathcal{A}^* \mathcal{W}$ i.e. \mathcal{L} is the set of all strings of the form xw , with $x \in \mathcal{A}^*$ and $w \in \mathcal{W}$. The sooner-statistic τ then corresponds to the smallest n such that $X_1 \cdots X_n$ has a suffix in \mathcal{W} . On the other hand, if \mathcal{W} is a *reduced set of strings* (i.e. no string in \mathcal{W} is a substring of another string in \mathcal{W}) then S_n corresponds to the total number of (possibly overlapping) substrings of $X_1 \cdots X_n$ that belong to \mathcal{W} .

Although our discussion will be specialized to the so called Markov models of random strings and regular patterns, for the sake of completeness we describe briefly other settings considered in the literature. The simplest probabilistic model for X is the so called *memoryless model* in which X_1, X_2, \dots form an i.i.d. sequence [1], [8], [10], [12]. Among the simplest models of random strings that convey a correlation structure, one finds *Markov models* [14], [15], [17], [18] and more generally *hidden Markov models* [19]. At the highest level of generality (that is analytically tractable) one finds *dynamical models* [3], [4]. In this case, for each $n \geq 1$, the distribution of X_{n+1} may depend on all the variables X_1, \dots, X_n .

The simplest type of patterns considered in the literature are *compound patterns*. This term is used to refer to a pattern consisting of a finite number of strings. However, these types of patterns are just particular cases of *regular patterns* [11], [20]. Broadly speaking, a pattern is said to be *regular* if it can be described by a regular expression over the alphabet characters of the type used by computer scientists. According to Kleene's and the Rabin-Scott theorem [11], [20], a pattern \mathcal{L} is *regular* if and only if it corresponds to the pattern recognized by a certain deterministic finite automaton. A *deterministic finite automaton* (in short, automaton) is a 5-tuple of the form $G = (V, \mathcal{A}, f, q, T)$, where V is a non-empty set (called *set of states*), $q \in V$ is a unique distinct state (called the *initial*

state), and $T \subset V$ is a distinguishable set of states (called *set of terminal states*). The term $f : V \times \mathcal{A}^* \rightarrow V$ is a function (called the *transition function*) that must satisfy the fundamental properties

$$f(v, \epsilon) = v; \quad (3)$$

$$f(v, xy) = f(f(v, x), y), \quad (4)$$

for all $v \in V$ and $x, y \in \mathcal{A}^*$. The *pattern recognized by G* is the set $\{x \in \mathcal{A}^* : f(q, x) \in T\}$.

Automata are usually represented as directed graphs with labeled edges: *an arrow labeled with the character α connecting a state u with a state v is drawn provided that $f(u, \alpha) = v$* . (See the top of Figure 1 for an example.)

A. The Markov chain embedding technique.

For the remaining of this manuscript $G = (V, \mathcal{A}, f, q, T)$ is an automaton and \mathcal{L} is the regular pattern recognized by G . We also specialize to the case of Markov models of random strings. By this we mean that X is a k -th order homogeneous Markov chain. If $k = 0$ this means that X_1, X_2, \dots is a sequence of i.i.d. random variables. In this case we talk more specifically about a *memoryless model*. Otherwise, for $k > 0$ this means that for all $n > k$ and $\alpha_1, \dots, \alpha_n \in \mathcal{A}$,

$$\begin{aligned} \mathbb{P}[X_n = \alpha_n \mid X_{n-1} = \alpha_{n-1}, \dots, X_1 = \alpha_1] \\ = \mathbb{P}[X_{k+1} = \alpha_n \mid X_k = \alpha_{n-1}, \dots, X_1 = \alpha_{n-k}]. \end{aligned}$$

Since the right-hand side above does not really depend on n , the above is equivalent to say that $((X_{n+1}, \dots, X_{n+k}))_{n \geq 0}$ is a first-order homogeneous Markov chain with state space \mathcal{A}^k (in the sense used by most introductory probability textbooks [5], [6]).

The Markov chain embedding technique is a well established technique used to understand the distribution of the statistics in (1) and (2) as well as other statistics associated with \mathcal{L} and X (e.g. the distance between two non-overlapping and consecutive occurrences of a pattern [16]). The term is attributed to Fu and Koutras [7] however the technique can be traced back to the work of Gerber and Li [8] and other authors [1], [2]. A major generalization of this technique is due to Nicodème et al. [15], [14] who formalized it in the context of regular patterns as opposed to compound or other special (yet still regular) patterns. See [13] for a self-contained discussion of the Markov chain embedding technique.

Broadly speaking, the Markov chain embedding technique consists in transforming a random string in \mathcal{A}^* into a random string in V^* , where V is the state space of an appropriate automaton (that usually recognizes the regular pattern of interest). The following definition [13] conceptualizes the technique in a probabilistic framework as used by most authors.

Definition 1.1: (Automata embeddings.) The embedding of X in G is the process $X^G = (X_n^G)_{n \geq 1}$ with state space $f(q, \mathcal{A}^+) = \{f(q, x) : x \in \mathcal{A}^+\} \subset V$, where $X_n^G := f(q, X_1 \cdots X_n)$. In particular, due to (4), it applies for all $n \geq 2$ that

$$X_n^G = f(X_{n-1}^G, X_n).$$

If X is produced by a memoryless source then X^G is clearly a first-order homogeneous Markov chain. This last feature is very useful to determine the moment generating function associated with the joint distribution of the sooner-times and count-statistics of several regular patterns [13]. However, regardless of the probabilistic nature of X , the following result applies — in general — as long as X^G turns out to be a first-order homogeneous Markov chain.

Proposition 1.2: (Generating functions associated with Markovian embeddings.) Assume that X^G is a first-order homogeneous Markov chain. If μ denotes the row-vector associated with the initial distribution of X^G and P denotes the probability transition matrix of X^G then the following applies.

- (a) Let μ_- be the vector μ but with all entries associated with T removed. Similarly, let P_- be the matrix obtained by removing all rows and columns associated with T . Finally, let p_- be the column-vector obtained by removing all rows associated with T from the vector obtained by adding up all the columns in P associated with T . If \mathbb{I} is the identity matrix then

$$\sum_{n=2}^{\infty} \mathbb{P}[\tau = n] x^n = x^2 \cdot \mu_- \cdot (\mathbb{I} - x \cdot P_-)^{-1} \cdot p_-. \quad (5)$$

- (b) Let μ_y be the row-vector obtained by multiplying by a marker-variable y all the entries in μ associated with states in T . Similarly, let P_y be the matrix obtained by multiplying by y the columns of P associated with states in T . If \mathbb{I} is the identity matrix and $\mathbf{1}$ is a column-vector of ones then

$$\sum_{n=1, k=0}^{\infty} \mathbb{P}[S_n = k] x^n y^k = x \cdot \mu_y \cdot (\mathbb{I} - x \cdot P_y)^{-1} \cdot \mathbf{1}. \quad (6)$$

Identity (5) is equivalent to have $\mathbb{P}[\tau = n] = \mu_- \cdot P_-^{n-2} \cdot p_-$, for all $n \geq 2$. This last identity follows from elementary results on homogeneous Markov chains after taking into account the following two observations: (i) the entry in column- v of $\mu_- \cdot P_-^{n-2}$ is the probability that $\{X_1^G, \dots, X_{n-1}^G\} \cap T = \emptyset$ and $X_{n-1}^G = v$, and (ii) the entry in row- v of p_- is the probability that $X_n^G \in T$ given that $X_{n-1}^G = v$.

On the other hand, identity (6) follows by a *transfer matrix method* argument [9]. The use of this method in the context of pattern statistics was introduced by Nicodème et al. [15], [14]. See [13] for other applications. The main idea here is that $\mathbb{P}[S_n = k]$ corresponds to the coefficient of y^k of the polynomial $\mu_y \cdot P_y^{n-1} \cdot \mathbf{1}$. This is because the coefficient in column- v of $\mu_y \cdot P_y^{n-1}$ is a polynomial in the variable y , and the coefficient of y^k of this polynomial corresponds to the probability that $|\{1 \leq l \leq n : X_l^G \in T\}| = k$ and $X_n^G = v$.

II. MAIN RESULTS

Proposition 1.2 is limited to embeddings that are Markovian. However, when the infinite sequence X is not produced by a memoryless source, there is no warranty that X^G will be Markovian. This is because the Markovianity of X^G depends on an affinity between the probabilistic model of X and

the automaton G where the infinite sequence of symbols is embedded into. In this extended abstract we consider the following questions: (i) *what conditions on the automaton G ensure that X^G is a first-order homogeneous Markov chain?*, (ii) *is there an automaton G that recognizes \mathcal{L} for which X^G is also markovian?*, and (iii) *what is the smallest state space size automaton G that recognizes \mathcal{L} for which the embedding X^G is markovian?*

The first and second question are considered in Section II-A. We emphasize that question (ii) could be answered using the Markov automata proposed by Nicodème et al. in [14], [15]. In here, we provide an alternative construction of this automaton using a synchronization argument. The third question is addressed in Section II-B. The discussion that follows is part of an ongoing research project by the author as the above questions are addressed only partially in this manuscript.

A. K -th order automata

For $x \in \mathcal{A}^*$ define $x_{(k)}$ to be the longest $y \in \mathcal{A}^{\leq k}$ such that $x = \dots y$. Observe that in general $x = \dots x_{(k)}$. Furthermore, for all $x, y \in \mathcal{A}^*$, it applies that

$$x_{(k)} = x \iff |x| \leq k, \quad (7)$$

$$(x_{(k)}y)_{(k)} = (xy)_{(k)}. \quad (8)$$

We start our discussion by introducing a notion of memory conveyed by the states in G .

Definition 2.1: (Memory-length.) For $v \in V$, say that v conveys a memory-length of order k if there exists a $\lambda \in \mathcal{A}^k$ such that if $x \in \mathcal{A}^{\geq k}$ and $f(q, x) = v$ then $x = \dots \lambda$. We say that G conveys a memory-length of order k if every $v \in V$ conveys a memory-length of order k .

It follows from the definition that every automaton conveys a memory-length of order 0. See the top of Figure 1 for an example of an automaton that conveys a memory-length of order 1 but not of order 2.

The appropriateness of the above definition in the context of the Markov chain embedding technique is revealed by the following result.

Proposition 2.2: If the automaton G conveys a memory-length of order k and X is a k -th order homogeneous Markov chain then $(X_n^G)_{n \geq k}$ is a first-order homogeneous Markov chain, with initial distribution and probability transition matrix given, respectively, by the formulae

$$\mathbb{P}(X_k^G = v) = \sum_{\lambda \in \mathcal{A}^k} \mathbb{P}(X_1 \cdots X_k = \lambda) \cdot \llbracket f(q, \lambda) = v \rrbracket, \quad (9)$$

$$\mathbb{P}(X_n^G = v \mid X_{n-1}^G = u) = \sum_{\alpha \in \mathcal{A}} \mathbb{P}(X_{k+1} = \alpha \mid X_k \cdots X_1 = \lambda) \cdot \llbracket f(u, \alpha) = v \rrbracket, \quad (10)$$

where the string λ in (10) is the one associated with the memory-length of u according to Definition 2.1.

Proof: Formula (9) is direct. To show the markovian property about X^G , let $n > k$ and $v_1, \dots, v_n \in V$. We determine the probability of the event $[X_n^G = v_n] \cap B$, where $B = [X_{n-1}^G = v_{n-1}, \dots, X_1^G = v_1]$. For this observe that

$B \subset [f(q, X_1 \cdots X_{n-1}) = v_{n-1}]$. In particular, since G conveys a memory-length of order k , there exists $\lambda_{n-1} \in \mathcal{A}^k$ such that $B \subset [X_{n-1} \cdots X_{n-k} = \lambda_{n-1}]$. As a result, using that $B \in \sigma(X_1, \dots, X_{n-1})$ and that X is a k -th order homogeneous Markov chain, we obtain that

$$\begin{aligned} \mathbb{P}([X_n^G = v_n] \cap B) &= \mathbb{P}(B) \cdot \mathbb{P}(X_n^G = v_n \mid B), \\ &= \mathbb{P}(B) \cdot \mathbb{P}(f(v_{n-1}, X_n) = v_n \\ &\quad \mid B \cap [X_{n-1} \cdots X_{n-k} = \lambda_{n-1}]), \\ &= \mathbb{P}(B) \cdot \sum_{\alpha} \mathbb{P}(X_{k+1} = \alpha \mid X_k \cdots X_1 = \lambda_{n-1}), \end{aligned}$$

where the indices in the above summation are restricted to those $\alpha \in \mathcal{A}$ such that $f(v_{n-1}, \alpha) = v_n$. Since the summation on the right-hand side above can be regarded as a function of (v_{n-1}, v_n) that does not depend on n , the above shows that X^G is a first-order homogeneous Markov chain for $n \geq k$. Formula (10) is now immediate from the above identity. ■

Proposition 2.2 answers question (i). We may now rephrase question (ii) as follows: *are there automata that recognize the regular language \mathcal{L} however convey arbitrarily large memory lengths?* We answer this question positively. The construction we propose is based on the combinatorial concept of *de Bruijn graph* which we restate next but in the framework of automata theory. (See [14] for an essentially equivalent construction to the automaton we propose in here.)

Definition 2.3: (de Bruijn automaton.) The de Bruijn automaton of order k is the automaton $dB_k = (\mathcal{A}^{\leq k}, \mathcal{A}, g, \epsilon, \mathcal{A}^k)$ with transition function $g : \mathcal{A}^{\leq k} \times \mathcal{A}^* \rightarrow \mathcal{A}^{\leq k}$ defined as

$$g(\lambda, x) := (\lambda x)_{(k)}. \quad (11)$$

Observe that g satisfies the fundamental properties described in (3)–(4) due to (7)–(8). In particular, dB_k is a well-posed automaton. Before addressing question (ii) we introduce one more automaton.

Definition 2.4: (k -th order automaton.) The k -th order automaton associated with G is the automaton $G_k = (V_k, \mathcal{A}, f_k, q_k, T_k)$, where $V_k := \mathcal{A}^{\leq k} \times V$, $q_k := (\epsilon, q)$, $T_k := \mathcal{A}^k \times T$, and $f_k : V_k \times \mathcal{A}^* \rightarrow V_k$ is the transition function defined as

$$f_k((\lambda, v), x) := (g(\lambda, x), f(v, x)) = ((\lambda x)_{(k)}, f(v, x)). \quad (12)$$

(As a side remark we notice that $G_k = dB \times G$, where \times is the product operation between automata. See [11], [20] to follow up on the concept of product automata). The above automaton is well-posed as the consistency properties of g and f are inherited by f_k . Indeed, for all $(\lambda, v) \in V_k$ and $x, y \in \mathcal{A}^*$, it applies that

$$\begin{aligned} f_k(f_k((\lambda, v), x), y) &= f_k((g(\lambda, x), f(v, x)), y), \\ &= (g(g(\lambda, x), y), f(f(v, x), y)), \\ &= (g(\lambda, xy), f(v, xy)), \\ &= f_k((\lambda, v), xy). \end{aligned}$$

The positive answer to question (ii) is finally conveyed by the following result.

Proposition 2.5: If G is a deterministic finite automaton that recognizes the regular language \mathcal{L} then G_k conveys a memory-length of order k and it recognizes the language $(\mathcal{A}^{\geq k} \cap \mathcal{L})$.

Proof: To see that G_k recognizes $(\mathcal{A}^{\geq k} \cap \mathcal{L})$, consider $x \in \mathcal{A}^*$ such that $f_k(q_k, x) \in T_k$. Since $f_k(q_k, x) = (x_{(k)}, f(q, x))$ and $T_k = \mathcal{A}^k \times T$, this is equivalent to say that $x_{(k)} \in \mathcal{A}^k$ and $f(q, x) \in T$. In particular, $|x| \geq k$ and $x \in \mathcal{L}$ i.e. $x \in (\mathcal{A}^{\geq k} \cap \mathcal{L})$. Therefore, G_k recognizes $(\mathcal{A}^{\geq k} \cap \mathcal{L})$. On the other hand, to show that G_k conveys a memory-length of order k , consider $(\lambda, v) \in V_k$ and suppose that $x \in \mathcal{A}^{\geq k}$ is such that $f_k(q_k, x) = (\lambda, v)$. Then $x_{(k)} = \lambda$, but since $|x| \geq k$, we conclude that $x = \dots\lambda$ with $|\lambda| = k$. Since λ does not depend on x , (λ, v) conveys a memory-length of order k . In particular, so does G_k . ■

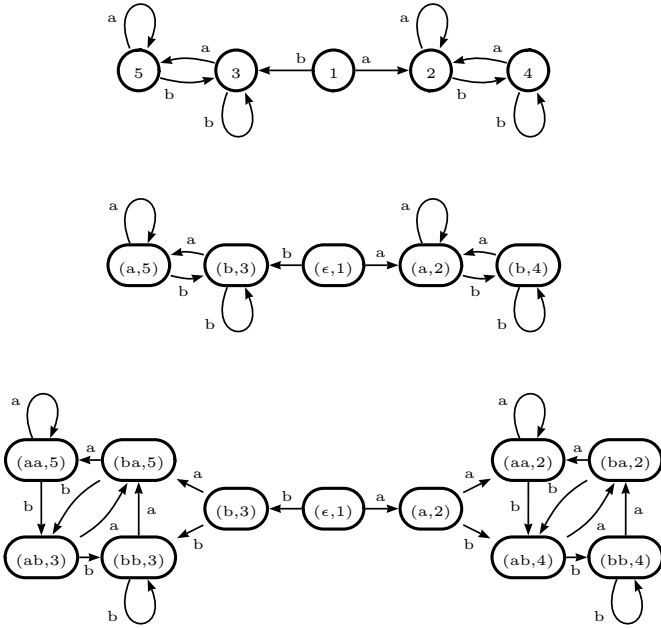


Fig. 1. Top, visual representation of automaton G that recognizes the pattern of all strings constructed with the characters a and b that have an equal number of occurrences of the patterns ab and ba as substrings. State 1 is the initial state. States 1, 2 and 3 are terminal states. According to Definition 2.1, G conveys a memory-length of order 1 but not of order 2. Middle, visual representation of the first-order automaton G_1 associated with the automaton G represented on top. Only states that are accessible from the initial state are displayed. State $(\epsilon, 1)$ is the initial state. States $(a, 2)$ and $(b, 3)$ are terminal states. According to Proposition 2.5, G_1 conveys a memory-length of order 1 and recognizes all non-empty strings with an equal number of occurrences of the patterns ab and ba as substrings. Bottom, visual representation of the second-order automaton G_2 associated with G . Only states that are accessible from the initial state are displayed. State $(\epsilon, 1)$ is the initial state. States $(aa, 2)$, $(ba, 2)$, $(ab, 3)$ and $(bb, 3)$ are terminal states. According to Proposition 2.5, G_2 conveys a memory-length of order 2 and recognizes all strings of length at least two that have an equal number of occurrences of the patterns ab and ba as substrings.

B. Minimality

According to the Mihill-Nerode theorem [11], there exists a unique automaton M that recognizes \mathcal{L} and has the smallest possible number of states. (Here uniqueness is to be interpreted up to automata isomorphisms i.e. one-to-one transformations between state spaces that preserve initial and terminal states as well as transitions between states.) However, when confronted with a k -th order homogeneous Markov chain X , there is no warranty that the embedding X^M will turn out to be a first-order homogeneous Markov chain. The goal of this section is to characterize the smallest possible state space size Markov chain that can be used to study the sooner-time or count statistics of \mathcal{L} in the context of Markov models. In what follows, we reserve the scripts $(U, \mathcal{A}, \delta, m, F)$ to refer to M . Our main result is the following one.

Theorem 2.6: If M is the minimal automaton that recognizes \mathcal{L} and $G = (V, \mathcal{A}, f, q, T)$ is any automaton that recognizes \mathcal{L} that conveys a memory-length of order k then $|f(q, \mathcal{A}^{\geq k})| \geq |\delta_k(m_k, \mathcal{A}^{\geq k})|$.

Above M_k is the k -th order automaton associated with M . In particular, m_k is the initial state of M_k and δ_k is the transition function of M_k . Theorem 2.6 asserts that the number of states in G accessible from q using words of length at least k is as least as large as the number of states in M_k that are accessible from m_k also using words of length at least k . Since the states in $f(q, \mathcal{A}^{\geq k})$ are all the states that $(X_n^G)_{n \geq k}$ could ever visit, the following result follows.

Corollary 2.7: If M is the minimal automaton that recognizes \mathcal{L} then, for all automaton G that recognizes \mathcal{L} and conveys a memory-length of order k , there exists a k -th order homogeneous Markov chain X with state space \mathcal{A} for which the state space of $(X_n^G)_{n \geq k}$ is at least as large as that of $(X_n^{M_k})_{n \geq k}$.

To fix ideas consider the automaton G on the top of Figure 1. This is the minimal automaton that recognizes the regular pattern \mathcal{L} of binary strings that have an equal number of occurrences of ab and ba as substrings. According to the above discussion, if X is a 2-nd order homogeneous Markov chain of $\{a, b\}$ -valued random variables such that $0 < \mathbb{P}(X_n = a \mid X_{n-1}X_{n-2} = \lambda) < 1$, for all $\lambda \in \{a, b\}^2$, then the state space $(X_n^G)_{n \geq k}$ — where G is any automaton that recognizes \mathcal{L} and conveys a memory-length of order 2 — will have at least 8 states (corresponding to the set of states $\{(aa, 2), (ab, 4), (bb, 4), (ba, 2), (ba, 5), (aa, 5), (ab, 3), (bb, 3)\}$ of the automaton at the bottom of Figure 1).

The proof of Theorem 2.6 follows closely the construction of the minimal automaton that recognizes \mathcal{L} provided by the Mihill-Nerode Theorem. Indeed, following the lines of the proof of this theorem in [11], consider in $\mathcal{A}^{\geq k}$ the equivalence relations

$$\begin{aligned} xR^{\mathcal{L}}y &\Leftrightarrow (\forall z \in \mathcal{A}^*) : (xz \in \mathcal{L} \Leftrightarrow yz \in \mathcal{L}) \\ &\quad \text{and } x_{(k)} = y_{(k)}; \\ xR^Gy &\Leftrightarrow f(q, x) = f(q, y) \text{ and } x_{(k)} = y_{(k)}. \end{aligned}$$

The number of equivalence classes of $R^{\mathcal{L}}$ and R^G are respectively denoted $|R^{\mathcal{L}}|$ and $|R^G|$. Furthermore, in what follows, a relation over $\mathcal{A}^{\geq k}$ is thought of as a set whose elements are the equivalent classes of the relation. Theorem 2.6 is almost a direct consequence of the following intermediate result.

Lemma 2.8: The following applies:

- (a) $R^{\mathcal{L}} = R^{M_k}$.
- (b) If G recognizes \mathcal{L} then $|R^{\mathcal{L}}| \leq |R^G|$.
- (c) If G conveys a memory-length of order k then

$$|R^G| = |f(q, \mathcal{A}^{\geq k})|.$$

Proof: To show (a) consider $x, y \in \mathcal{A}^{\geq k}$. Notice that

$$xR^{M_k}y \Leftrightarrow x_{(k)} = y_{(k)} \text{ and } \delta(m, x) = \delta(m, y). \quad (13)$$

In particular, if $xR^{M_k}y$ then, according to (4), the following logic equivalences hold for all $z \in \mathcal{A}^*$:

$$\begin{aligned} xz \in \mathcal{L} &\Leftrightarrow \delta(m, xz) \in T \\ &\Leftrightarrow \delta(\delta(m, x), z) \in T \\ &\Leftrightarrow \delta(\delta(m, y), z) \in T \\ &\Leftrightarrow \delta(m, yz) \in T \\ &\Leftrightarrow yz \in \mathcal{L}. \end{aligned} \quad (14)$$

As a result, $xR^{M_k}y$ implies $xR^{\mathcal{L}}y$. To show the converse, we see from (13) that it is enough to verify that $xR^{\mathcal{L}}y$ implies $\delta(m, x) = \delta(m, y)$. For this we appeal to the Mihill-Nerode theorem which asserts that up to an isomorphism $M = (U, \mathcal{A}, \delta, m, F)$, where U is the set of equivalence classes of $R^{\mathcal{L}}$ when regarded as a relation over \mathcal{A}^* . Furthermore, if $[x]$ denotes the equivalence class of $x \in \mathcal{A}^*$, then $m = [\epsilon]$, $F = \{[x] : x \in \mathcal{L}\}$ and, for all $x \in \mathcal{A}^*$ and $\alpha \in \mathcal{A}$, $\delta([x], \alpha) = [x\alpha]$. Thus suppose that $x, y \in \mathcal{A}^{\geq k}$ and $xR^{\mathcal{L}}y$. Then, in particular, $[x] = [y]$ but then $\delta(m, x) = [x] = [y] = \delta(m, y)$. Hence $xR^{M_k}y$. This completes the proof of part (a).

To prove (b) it is enough to show that R^G is a refinement of $R^{\mathcal{L}}$ i.e. that xR^Gy implies $xR^{\mathcal{L}}y$. Indeed, since xR^Gy implies that $f(q, x) = f(q, y)$, it follows from the same argument used in (14) that for all $z \in \mathcal{A}^*$, $xz \in \mathcal{L}$ if and only if $yz \in \mathcal{L}$. Hence $xR^{\mathcal{L}}y$ and this shows (b).

To verify (c), consider the transformation $\Phi : R^G \rightarrow f(q, \mathcal{A}^{\geq k})$ defined as $\Phi(E) := f(q, x)$, for $x \in E \in R^G$. Observe that $\Phi(E)$ is uniquely defined because if $x, y \in E \in R^G$ then, in particular, $f(q, x) = f(q, y)$. Furthermore, $\Phi(E) \in f(q, \mathcal{A}^{\geq k})$ because R^G is an equivalence relation in $\mathcal{A}^{\geq k}$. To demonstrate (c) we show that Φ is a bijection. First we show that Φ is one-to-one. Indeed, if $E_1, E_2 \in R^G$ are such that $\Phi(E_1) = \Phi(E_2)$ then $f(q, x_1) = f(q, x_2)$, for all $x_1 \in E_1$ and $x_2 \in E_2$. Since G conveys a memory-length of order k and $|x_1|, |x_2| \geq k$, it follows that $x_{1(k)} = x_{2(k)}$. Therefore $x_1R^Gx_2$, in particular, $E_1 = E_2$. This shows that Φ is one-to-one. Finally, to verify that Φ is surjective, consider $v \in f(q, \mathcal{A}^{\geq k})$. In particular, there exists $x \in \mathcal{A}^{\geq k}$ such that $v = f(q, x)$. Since R^G is an equivalence class in $\mathcal{A}^{\geq k}$, there exists $E \in R^G$ such that $x \in E$. Since $\Phi(E) = v$, it follows

that Φ is surjective and this completes the proof of part (c). \blacksquare

The proof of Theorem 2.6 goes as follows. If G is any automaton that recognizes \mathcal{L} and conveys a memory-length of order k , it follows from the above lemma that $|R^{M_k}| \leq |f(q, \mathcal{A}^{\geq k})|$. Since, according to Proposition 2.5, M_k conveys a memory-length of order k , part (c) above implies that $|R^{M_k}| = |\delta_k(m_k, \mathcal{A}^{\geq k})|$ and the theorem follows.

ACKNOWLEDGMENT

The author would like to thank Robert S. Maier and Pierre Nicodème for some insightful conversations about automata theory.

REFERENCES

- [1] E. A. Bender and F. Kochman. The distribution of subword counts is usually normal. *Eur. J. Comb.*, 14(4):265–275, 1993.
- [2] J. D. Biggins and C. Cannings. Markov renewal processes, counters and repeated sequences in Markov chains. *Adv. Appl. Prob.*, 19:521–545, 1987.
- [3] J. Bourdon and B. Vallée. Generalized pattern matching statistics. In *Colloquium on Mathematics and Computer Science : Algorithms and Trees*, Trends in Mathematics, pages 249–265. Birkhauser, 2002.
- [4] J. Bourdon and B. Vallée. Pattern matching statistics on correlated sources. In *Proc. of the 7th Latin American Symposium on Theoretical Informatics (LATIN'06)*, pages 224–237, Valdivia, Chile, 2006.
- [5] R. Durrett. *Essentials of stochastic processes*. Springer, 1999.
- [6] R. Durrett. *Probability: theory and examples*. Duxbury Press, third edition, 2004.
- [7] J. C. Fu and M. V. Koutras. Distribution theory of runs: a Markov chain approach. *J. Amer. Statist. Assoc.*, 89(427):1050–1058, 1994.
- [8] H. U. Gerber and S.-Y. R. Li. The occurrence of sequence patterns in repeated experiments and hitting times in a Markov chain. *Stochastic Processes and their Applications*, 11(1):101–108, 1981.
- [9] I. P. Goulden and D. M. Jackson. *Combinatorial Enumeration*. Dover Publications, 2004.
- [10] L. J. Guibas and A. M. Odlyzko. String overlaps, pattern matching, and nontransitive games. *J. Comb. Theory, Ser. A*, 30(2):183–208, 1981.
- [11] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 1979.
- [12] S.-Y. R. Li. A martingale approach to the study of occurrence of sequence patterns in repeated experiments. *The Annals of Probability*, 8(6):1171–1176, 1980.
- [13] M. Lladser, M. D. Betterton, and R. Knight. Multiple pattern matching: A Markov chain approach. 2006. To appear in the *J. Math. Bio.*
- [14] P. Nicodème. Regexpcount, a symbolic package for counting problems on regular expressions and words. *Fundamenta Informaticae*, 56(1-2):71–88, 2003.
- [15] P. Nicodème, B. Salvy, and P. Flajolet. Motif statistics. *Theoretical Computer Scienc*, 287(2):593–617, 2002.
- [16] Y. Park and J. L. Spouge. Searching for multiple words in a Markov sequence. *INFORMS Journal on Computing*, 16(4):341–347, 2004.
- [17] M. Régnier. A unified approach to word occurrences probabilities. *Discrete Applied Mathematics*, 104(1):259–280, 2000. Special issue on Computational Biology.
- [18] M. Régnier and W. Szpankowski. On pattern frequency occurrences in a Markovian sequence. *Algorithmica*, 22(4):631–649, 1998.
- [19] S. Robin, F. Rodolphe, and S. Schbath. *DNA, Words and Models*. Cambridge University Press, New York, NY, USA, 2005.
- [20] M. Sipser. *Introduction to the Theory of Computation*. International Thomson Publishing, 1996.