Generalizing the Blum-Elias Method for Generating Random Bits from Markov Chains

Hongchao Zhou

Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125
Email: hzhou@caltech.edu

Jehoshua Bruck

Department of Electrical Engineering
California Institute of Technology
Pasadena, CA 91125
bruck@caltech.edu

Abstract—The problem of random number generation from an uncorrelated random source (of unknown probability distribution) dates back to von Neumann’s 1951 work. Elias (1972) generalized von Neumann’s scheme and showed how to achieve optimal efficiency in unbiased random bits generation. Hence, a natural question is what if the sources are correlated? Both Elias and Samueleson proposed methods for generating unbiased random bits in the case of correlated sources (of unknown probability distribution), specifically, they considered finite Markov chains. However, their proposed methods are not efficient (Samueleson) or have implementation difficulties (Elias). Blum (1986) devised an algorithm for efficiently generating random bits from degree-2 finite Markov chains in expected linear time, however, his beautiful method is still far from optimality. In this paper, we generalize Blum’s algorithm to arbitrary degree finite Markov chains and combine it with Elias’s method for efficient generation of unbiased bits. As a result, we provide the first known algorithm that generates unbiased random bits from an arbitrary finite Markov chain, operates in expected linear time and achieves the information-theoretic upper bound on efficiency.

I. INTRODUCTION

The problem of random number generation dates back to von Neumann [1] who considered the problem of simulating an unbiased coin by using a biased coin with unknown probability. He observed that when one focuses on a pair of coin tosses, the events HT and TH have the same probability; hence, HT produces the output symbol 0 and TH produces the output symbol 1. The other two possible events, namely, HH and TT , are ignored, namely, they do not produce any output symbols. More efficient algorithms to generate random bits from a biased coin were proposed by Hoeffding and Simons [2], Stout and Warren [3] and Peres [4]. Elias [5] gave an optimal procedure such that the expected number of unbiased random bits generated per coin toss is asymptotically equal to the entropy of the biased coin. On the other hand, Knuth and Yao [6] gave a simple procedure to generate arbitrary distribution from an unbiased coin. Han and Hoshi [7] generalized this problem to consider the case that the given coin is biased with a known distribution.

In this paper, we study the problem of generating random bits from an arbitrary and unknown finite Markov chain. The input to our problem is a sequence of symbols that

represent a random trajectory through the states of the Markov chain - given this input sequence our algorithm generates an independent unbiased binary sequence called the output sequence. This problem was first studied by Samueleson [8]. His approach was to focus on a single state (ignoring the other states) treat the transitions out of this state as the input process, hence, reducing the problem of correlated sources to the problem of a single random source; obviously, this method is not efficient. Elias [5] suggested that for each state, we produce an independent output sequence, then the output sequence can be generated by pasting (concatenating) the collection of output sequences. Although this method is much more efficient than Samueleson’s, we will show that it is not always correct to paste the output sequences directly. In fact, Blum [9] probably realized it, as he mentioned that: (i) “Elias’s algorithm is excellent, but certain difficulties arise in trying to use it (or the original von Neumann scheme) to generate bits in expected linear time from a Markov chain”, and (ii) “Elias has suggested a way to use all the symbols produced by a MC (Markov Chain). His algorithm approaches the maximum possible efficiency for a one-state MC. For a multi-state MC, his algorithm produces arbitrarily long finite sequences. He does not, however, show how to paste these finite sequences together to produce *infinitely* long independent unbiased sequences.” Blum [9] worked on this problem and derived a beautiful algorithm to generate random bits from a degree-2 Markov chain *in expected linear time* by extending the single coin von Neumann scheme. While his approach can be extended to arbitrary out-degrees (the general Markov chain model used in this paper), the information-efficiency is still far from optimality due to the limitation (compared to Elias’s algorithm) of von Neumann scheme.

In this paper, we generalize Blum’s algorithm to arbitrary degree finite Markov chains and combine it with Elias’s method for efficient generation of unbiased bits. As a result, we provide the first known algorithm that generates unbiased random bits from arbitrary finite Markov chains, operates in expected linear time and achieves the information-theoretic upper bound on efficiency. Specifically, we propose Algorithm A, which is a simple modification of Elias’s suggestion to generate random bits, it operates on finite sequences and its efficiency can reach the information-theoretic upper bound in

the limit of long input sequences. In addition, we propose Algorithm *B*, it is a combination of Blum's and Elias's algorithms, it generates infinitely long sequences of random bits in expected linear time.

The remainder of this paper is organized as follows. Section II introduces Elias's schemes to generate random bits from any biased coin. Section III presents our main lemma that allows the generalization to arbitrary Markov chains. Algorithm *A* is presented and analyzed in Section IV, it is a simple modification of Elias's suggestion for Markov chain. Algorithm *B* is presented in Section V, it is a generalization of Blum's algorithm.

II. ELIAS'S SCHEME GENERATION OF RANDOM BITS

Consider a length N sequence generated by a biased n -face coin $X = x_1x_2\dots x_N \in \{s_1, s_2, \dots, s_n\}^*$ such that the probability to get s_i is p_i , with $\sum_{i=1}^n p_i = 1$. While we are given a sequence X the probabilities $p_1, p_2, \dots, p_n$ are unknown, the question is: How can we generate an independent and unbiased sequence of 0's and 1's from X ? The efficiency of a generation algorithm is defined as the ratio between the expected length of the output sequence and the length of the input sequence.

Elias [5] proposed an optimal (in terms of efficiency) generation algorithm; for the sake of completeness we describe it here. His method is based on the following idea: The possible n^N input sequences of length N can be partitioned into classes such that all the sequences in the same class have the same number of s_k 's with $1 \leq k \leq n$. Note that for every class, the members of the class have the same probability to be generated. Now, our goal is to assign a string of bits (the output) to each possible input sequence, such that any two output sequences Y and Y' with the same length (say k), have the same probability to be generated. In general, for a class with m members that were not assigned yet, assign 2^j possible output binary sequences of length j to 2^j distinct unassigned members, where $2^j \leq m < 2^{j+1}$. Repeat the procedure above for the rest of the members that were not assigned. Note that when a class has odd number of members, there will be one and only one member assigned to ϕ (the empty string).

The foregoing assignment algorithm is not efficient as it is using a lookup table, however, efficient assignment can be achieved by using the lexicographic order. Given input sequence X of length N , the output sequence can be written as a function of X , denoted by $f_E(X)$, called Elias's function. Pae and Loui [10] showed that Elias's function is computable in polynomial time. In the rest of this paper, we assume that $f_E(X)$ is computed using the procedure in [10]. For this $f_E(X)$, it has the following property.

Lemma 1 (Property of f_E). *Given an input sequence X of length N such that $Y = f_E(X) \in \{0, 1\}^k$. Then for any $Y' \in \{0, 1\}^k$, there exists one and only one input sequence X' , that is a permutation of X , such that $f_E(X') = Y'$.*

III. MAIN LEMMA: EQUIVALENCE OF EXIT SEQUENCES

Our goal is to efficiently generate random bits from a Markov chain with unknown transition probabilities. The

paradigm we study is that a Markov chain generates the sequence of states that it is visiting and this sequence of states is the input sequence to our algorithm for generating random bits. Specifically, we express an input sequence as $X = x_1x_2\dots x_N$ with $x_i \in \{s_1, s_2, \dots, s_n\}$, where $\{s_1, s_2, \dots, s_n\}$ indicate the states of a Markov chain. We first define the following notations:

- $x_a, X[a]$: the a^{th} element of X
- $X[a : b]$: subsequence of X from the a^{th} to b^{th} element
- X^a : $X[1 : a]$
- $Y \equiv X$: Y is a permutation of X
- $Y \doteq X$: Y is a permutation of X and $y_{|Y|} = x_{|X|}$

The key idea is that for a given Markov chain, we can treat each state as a coin and treat its exits as the corresponding coin tosses. Namely, we can generate a collection of independent sequences $\pi(X) = [\pi_1(X), \pi_2(X), \dots, \pi_n(X)]$, called exit sequences, where $\pi_i(X)$ is the sequence of states following s_i in X , namely, $\pi_i(X) = \{x_{j+1} | x_j = s_i, 1 \leq j < N\}$.

Lemma 2 (Uniqueness). [11] *An input sequence X can be uniquely determined by x_1 and $\pi(X)$.*

Lemma 3 (Equal-probability). [11] *Two input sequences $X = x_1x_2\dots x_N$ and $Y = y_1y_2\dots y_N$ with $x_1 = y_1$ have the same probability to be generated if $\pi_i(X) \equiv \pi_i(Y)$ for all $1 \leq i \leq n$.*

In [5], Elias has suggested to concatenate the output sequences that correspond to $\pi_1(X), \pi_2(X), \dots$ as the final output. However, Elias did not show how to correctly paste these output sequences. A straightforward method is to concatenate all these output sequences. However, the method of direct concatenation above is not always correct. Consider a simple example of a two-state Markov chain in which $P[s_2 | s_1] = p_1$ and $P[s_1 | s_2] = p_2$, as shown in Fig. 1. Assume an input

Fig. 1. An instance of Markov chain.

sequence with length $N = 4$ is generated from this Markov chain and the starting state is s_1 , then the probabilities of the possible input sequences and their corresponding output sequences (based on direct concatenation) are given in Table I. We can see that when the input sequence length $N = 4$, the probability to produce a bit 0 is $p_1(1-p_1)(2-p_1)$ while the probability to produce a bit 1 is $p_1p_2(2-p_1-p_2)$. They are different for some p_1, p_2 . So we can conclude if use the direct concatenation approach we might generate an output sequence which is not independent or unbiased.

Let's try to intuitively understand why the direct concatenation of the output sequences of $\pi_1(X), \pi_2(X), \dots$ does not work. The reason is that while the elements belonging to the same exit sequence $\pi_i(X)$ are independent, the exit sequences $\pi_1(X), \pi_2(X), \dots$ may not be independent. Namely, we cannot

Input	Probability	Output concatenation	Output Algorithm A
$s_1 s_1 s_1 s_1$	$(1-p_1)^3$	ϕ	ϕ
$s_1 s_1 s_1 s_2$	$(1-p_1)^2 p_1$	0	ϕ
$s_1 s_1 s_2 s_1$	$(1-p_1) p_1 p_2$	0	0
$s_1 s_1 s_2 s_2$	$(1-p_1) p_1 (1-p_2)$	0	ϕ
$s_1 s_2 s_1 s_1$	$p_1 p_2 (1-p_1)$	1	1
$s_1 s_2 s_1 s_2$	$p_1^2 p_2$	ϕ	ϕ
$s_1 s_2 s_2 s_1$	$p_1 (1-p_2) p_2$	1	ϕ
$s_1 s_2 s_2 s_2$	$p_1 (1-p_2)^2$	ϕ	ϕ

TABLE I
AN EXAMPLE OF DIRECT CONCATENATION AND ALGORITHM A.

treat $\pi_1(X), \pi_2(X), \dots, \pi_n(X)$ as tosses of n different coins. This motivated us to study the characterization of exit sequences that are feasible in Markov chains (the equivalence of exit sequences).

Lemma 4 (Main Lemma). *Given an input sequence $X = x_1 x_2 \dots x_N$ with $x_N = s_\chi$, produced from a Markov chain, for any $[\Lambda_1, \Lambda_2, \dots, \Lambda_n]$ with*

- 1) Λ_i is a permutation ($\equiv$) of $\pi_i(X)$ for $i = \chi$.
- 2) Λ_i is a tail-fixed permutation ($\doteq$) of $\pi_i(X)$ for $i \neq \chi$.

then there exists one sequence $X' = x'_1 x'_2 \dots x'_N$ such that $x'_1 = x_1$ and $\pi(X') = [\Lambda_1, \Lambda_2, \dots, \Lambda_n]$. For this X' , we have $x'_N = x_N$.

The proof of this lemma can be found in [11]. Now, we can demonstrate our results on the equivalence of exit sequences by considering the following example. Let

$$X = s_1 s_4 s_2 s_1 s_3 s_2 s_3 s_1 s_1 s_2 s_3 s_4 s_1$$

with $\chi = 1$ and its exit sequences is given by $[s_4 s_3 s_1 s_2, s_1 s_3 s_3, s_2 s_1 s_4, s_2 s_1]$. After permutating all the exit sequences (for $i \neq 1$, we keep the last element of the i^{th} sequence fixed), we get a new group of exit sequences $[s_1 s_2 s_3 s_4, s_3 s_1 s_3, s_1 s_2 s_4, s_2 s_1]$. Based on these new exit sequences, we can generate a new input sequence

$$X' = s_1 s_1 s_2 s_3 s_1 s_3 s_2 s_1 s_4 s_2 s_3 s_4 s_1$$

IV. ALGORITHM A : MODIFICATION OF ELIAS'S SUGGESTION

In the section above, we see that Elias suggested to paste the outputs of different exit sequences together, as the final output, but the simple direct concatenation cannot always work. By modifying the method to paste these outputs, we get Algorithm A to generate unbiased random bits from any Markov chain.

Algorithm A

Input: A sequence $X = x_1 x_2 \dots x_N$ produced by a Markov chain, where $x_i \in S = \{s_1, s_2, \dots, s_n\}$.

Output: A sequence of 0's and 1's.

Main Function:

```

Suppose  $x_N = s_\chi$ .
for  $i := 1$  to  $n$  do
  if  $i = \chi$  then
    Output  $f_E(\pi_i(X))$ .
  else

```

```

    Output  $f_E(\pi_i(X)^{|\pi_i(X)|-1})$ 
  end if
end for
Comment:  $f_E(X)$  is Elias's function.

```

The only difference between Algorithm A and direct concatenation is that: Algorithm A ignores the last symbols of some exit sequences. Let's go back to the example of a two-state Markov chain in the last section and apply Algorithm A. We can see that when the input length $N = 4$, a bit 0 and a bit 1 have the same probability to be generated and no longer sequences can be generated, see Table I. In this case, the output sequence is independent and unbiased.

Lemma 5 (Mapping for Algorithm A). *In Algorithm A, assume input sequence $X = x_1 x_2 \dots x_N$ underlies an output sequence $Y \in \{0, 1\}^k$ for some k , then for any $Y' \in \{0, 1\}^k$, there exists one and only one sequence $X' = x'_1 x'_2 \dots x'_N$ which underlies Y' such that*

- 1) $x'_1 = x_1$ and $x'_N = x_N = s_\chi$ for some χ .
- 2) If $i = \chi$, $\pi_i(X') \equiv \pi_i(X)$ and

$$|f_E(\pi_i(X'))| = |f_E(\pi_i(X))|$$

- 3) For all $i \neq \chi$, $\pi_i(X') \doteq \pi_i(X)$ and

$$|f_E(\pi_i(X')^{|\pi_i(X')|-1})| = |f_E(\pi_i(X)^{|\pi_i(X)|-1})|$$

Proof: First we prove that for any Y' with $|Y'| = |Y|$, there exists such input sequence $X' = x'_1 x'_2 \dots x'_N$ satisfying all the requirements. Now, given a desired sequence Y' , we try to construct X' . Let's split Y' into several segments, $Y'_1, Y'_2, \dots, Y'_n$, namely $Y' = Y'_1 Y'_2 \dots Y'_n$ such that the length of Y'_i for $1 \leq i \leq n$ is given by

$$|Y'_i| = \begin{cases} |f_E(\pi_i(X))| & \text{if } i = \chi \\ |f_E(\pi_i(X)^{|\pi_i(X)|-1})| & \text{if } i \neq \chi \end{cases}$$

According to Lemma 1, if $i = \chi$, we can find $\Lambda_i \equiv \pi_i(X)$ such that $f_E(\Lambda_i) = Y'_i$. If $i \neq \chi$, we can find Λ_i such that $\Lambda_i^{|\Lambda_i|-1} \equiv \pi_i(X)^{|\pi_i(X)|-1}$, $\Lambda_i[\Lambda_i] = \pi_i(X)[\pi_i(X)]$ and $f_E(\Lambda_i^{|\Lambda_i|-1}) = Y'_i$. Based on Lemma 4, there exists a sequence $X' = x'_1 x'_2 \dots x'_N$ with $x'_1 = x_1, x'_N = x_N$ and its exit sequences are $(\Lambda'_1, \Lambda'_2, \dots, \Lambda'_n)$. It is not hard to check that X' satisfies all the requirements in the lemma.

Next, we prove that there are at most one sequence satisfying the requirements in the lemma. Assume there are two (or more) sequences U, V satisfying the requirements. According to Lemma 2, we know that they have different exit sequences. Then there exists a state s_i , such that $\pi_i(U) \neq \pi_i(V)$. If $i = \chi$, we also have $\pi_i(U) \equiv \pi_i(V)$, so we have $f_E(\pi_i(U)) \neq f_E(\pi_i(V))$, that means different Y'_i are produced, which contradicts with our assumption. If $i \neq \chi$, we will have $\pi_i(U) \doteq \pi_i(V)$, so we have $f_E(\pi_i(U)^{|\pi_i(U)|-1}) \neq f_E(\pi_i(V)^{|\pi_i(V)|-1})$, which also lead to contradictions. ■

Theorem 6 (Algorithm A). *Let the sequence generated by a Markov chain be used as input to algorithm A, then the output of Algorithm A is an independent unbiased sequence.*

Proof: Assume the length of input sequence is N , then we want to show that for any $Y, Y' \in \{0, 1\}^k$, Y and Y' have the same probability to be generated.

Let $S_u(Y)$ and $S_u(Y')$ denote the sets of input sequences underlying output sequences Y and Y' . According to Lemma 5 above, there is a one-to-one mapping between the elements in $S_u(Y)$ and $S_u(Y')$. Based on Lemma 3, we know that this mapping does not change the probability for sequences to be generated. As a result, from a Markov chain, the probability to generate a sequence in $S_u(Y)$ is equal to that to generate a sequence in $S_u(Y')$. So Y and Y' have the same probability to be generated. It means that any sequence of the same length k will be generated with the same probability. The result in the theorem is immediate from this conclusion. ■

Theorem 7 (Efficiency). *Let X be a sequence of length N generated by a Markov chain, which is used as input to algorithm A. Suppose the length of its output sequence is M , then the limiting efficiency $\eta_N = \frac{E[M]}{N}$ as $N \rightarrow \infty$ realizes the upper bound $\frac{H(X)}{N}$.*

The proof of this theorem can be found in [11].

V. ALGORITHM B : GENERALIZATION OF BLUM'S ALGORITHM

Given an input sequence, it is efficient to generate independent unbiased sequences using Algorithm A. However, it still has some limitations: (1) The whole input sequence has to be stored. (2) The input cannot be a stream or infinitely long, that is because no bits can be generated until the whole input sequence is provided. (3) It is not computable in expected linear time. In [9], Blum showed how to extend von Neumann scheme to generate an independent unbiased sequence of 0's and 1's from any Markov chain in expected linear time. As a generalization of Blum's algorithm, we propose Algorithm B by applying Elias scheme instead of von Neumann scheme.

Algorithm B

Input: A sequence (or a stream) $x_1x_2\dots$ produced by a Markov chain, where $x_i \in \{s_1, s_2, \dots, s_n\}$.

Parameter: n positive integer functions (window size) $\varpi_i(k)$ with $k \geq 1$ for $1 \leq i \leq n$.

Output: A sequence (or a stream) of 0's and 1's.

Main Function:

$E_i = \phi$ (empty) for all $1 \leq i \leq n$.

$k_i = 1$ for all $1 \leq i \leq n$.

c : the index of current state, namely, $s_c = x_1$.

while next input symbol is s_j ($\neq \text{null}$) **do**

if $|E_j| \geq \varpi_j(k_j)$ **then**

 Output $f_E(E_j)$.

$E_j = \phi$.

$k_j = k_j + 1$.

end if

$E_c = E_c s_j$ (Add s_j to E_c).

$c = j$.

end while

Note that Blum's algorithm is a special case of Algorithm B by setting the window size functions $\varpi_i(k) = 2$ for all $1 \leq i \leq n$ and $k \in \{1, 2, \dots\}$. Namely, Algorithm B is a generalization of Blum's algorithm. Assume a sequence of symbols $X = x_1x_2\dots x_N$ with $x_N = s_\chi$ have been read by the algorithm above, we want to show that for any N , the output sequence is always independent and unbiased. For all i with $1 \leq i \leq n$, we can write

$$\pi_i(X) = F_{i1}F_{i2}\dots F_{im_i}E_i$$

where F_{ij} with $1 \leq j \leq m_i$ are the segments used to generate outputs. For all i, j , we have $|F_{ij}| = \varpi_i(j)$ and

$$\begin{cases} 0 \leq |E_i| < \varpi_i(m_i + 1) & \text{if } i = \chi \\ 0 < |E_i| \leq \varpi_i(m_i + 1) & \text{otherwise} \end{cases}$$

Lemma 8 (Mapping for Algorithm B). *In Algorithm B, assume input sequence $X = x_1x_2\dots x_N$ underlies an output sequence $Y \in \{0, 1\}^k$ for some k , then for any $Y' \in \{0, 1\}^k$, there exists one and only one sequence $X' = x'_1x'_2\dots x'_N$ which underlies Y' such that*

1) $x'_1 = x_1$.

2) For all i with $1 \leq i \leq n$

$$\pi_i(X) = F_{i1}F_{i2}\dots F_{im_i}E_i$$

$$\pi_i(X') = F'_{i1}F'_{i2}\dots F'_{im_i}E_i$$

3) For all i, j , we have

$$F_{ij} \equiv F'_{ij} \text{ and } |f_E(F_{ij})| = |f_E(F'_{ij})|$$

Proof: Let's construct such a sequence X' satisfying all the requirements in the lemma:

1) Let $S = \{(i, j) | \forall 1 \leq i \leq n, 1 \leq j \leq m_i\}$. Let $w = \sum_i m_i$ be the number of segments generated from X .

2) Construct a sequence $\bar{X}$ with $\pi_i(\bar{X}) = \bar{F}_{i1}\bar{F}_{i2}\dots\bar{F}_{im_i}E_i$ such that

a) $\bar{x}_1 = x_1$.

b) For all i, j , $\bar{F}_{ij} = \begin{cases} F_{ij} & \text{if } (i, j) \in S \\ F'_{ij} & \text{otherwise} \end{cases}$

According to the main lemma, we know that such sequence $\bar{X}$ exists and it is unique.

3) Let $\bar{X}$ be the input sequence, and $\bar{F}_{i_1j_1}, \bar{F}_{i_2j_2}, \dots, \bar{F}_{i_wj_w}$ be the ordered segments used to produce outputs, i.e., when the input sequence is $\bar{X}$, the output sequence is

$$f_E(\bar{F}_{i_1j_1})f_E(\bar{F}_{i_2j_2})\dots f_E(\bar{F}_{i_wj_w})$$

4) Let $k = \max\{m | 1 \leq m \leq w, (i_m, j_m) \in S\}$, then we do the following operations:

a) Let $S = S / (i_k, j_k)$.

b) Let $l = |f_E(F_{i_kj_k})|$, then set $F'_{i_kj_k}$ such that $F'_{i_kj_k} \equiv F_{i_kj_k}$ and

$$f_E(F'_{i_kj_k}) = Y'[|Y'| - l + 1 : |Y'|]$$

c) Let $Y' = Y'[1 : |Y'| - l]$.

5) Repeat step 2-4) until S becomes empty (ϕ).

6) Repeat step 2) and 3) and let $X' = \bar{X}$.

We want to prove that the X' constructed using the procedure above satisfies all the requirements in Lemma 6. It is obvious that X' satisfies the conditions 1) to 3) in Lemma 6, the only thing that we need to prove is that X' underlies Y' .

Note that we have to repeat step 2) and 3) for $w + 1$ times. We want to prove that after the k^{th} iteration of step 3), the following conclusion is true:

- 1) $G = f_E(\bar{F}_{i_{w-k+2}j_{w-k+2}}) \dots f_E(\bar{F}_{i_w j_w})$ is a suffix of Y' .
- 2) $S = \{(i_m, j_m) | 1 \leq m \leq w - k + 1\}$

Let's prove the conclusion above by induction. First, after the first iteration, we can get that $G = \phi$, so the conclusion is true. Now, let's assume that this conclusion is true after the k^{th} iteration, we will prove that this conclusion is also true after the $k+1^{th}$ iteration when $k \leq w$. In the following proof, all the symbols are for the k^{th} iteration. In order to distinguish with the k^{th} iteration, we use $*$ to mark the symbols for the $k+1^{th}$ iteration.

According to our assumption, when input sequence is $\bar{X}$, the output is

$$f_E(\bar{F}_{i_1 j_1}) f_E(\bar{F}_{i_2 j_2}) \dots f_E(\bar{F}_{i_w j_w})$$

where $\{(i_m, j_m) | 1 \leq m \leq w - k + 1\} = S$ and G is a suffix of Y' . According to step 4), we know that $\bar{X}^*$ is constructed from X by replacing $F_{i_{w-k+1}j_{w-k+1}}$ with $F'_{i_{w-k+1}j_{w-k+1}}$. In the following, we prove that $\bar{X}^*$ satisfies the properties in the conclusion.

Let $\bar{x}_\alpha$ be the α^{th} symbol in $\bar{X}$ such that once $\bar{x}_\alpha$ is provided to the algorithm, it outputs $f_E(F_{i_{w-k-1}j_{w-k-1}})$. When the input sequence is $\bar{X}^\alpha$, a subsequence of $\bar{X}$, the output sequence will be

$$f_E(F_{i_1 j_1}) f_E(F_{i_2 j_2}) \dots f_E(F_{i_{w-k-1} j_{w-k-1}})$$

According to Lemma 4, we know that if $F_{i_{w-k+1}j_{w-k+1}}$ is replaced with $F'_{i_{w-k+1}j_{w-k+1}}$, we can get a new sequence U^α with length α such that when the input sequence is U^α , the algorithm will output $f_E(F_{i_1 j_1}), \dots, f_E(F_{i_{w-k} j_{w-k}})$ in some order and output $f_E(F'_{i_{w-k+1}j_{w-k+1}})$ at the end (after reading the last symbol). Note that, after processing U^α , all the values of the variables stored in the memory are the same as those after processing $\bar{X}^\alpha$. If we concatenate U^α with $\bar{X}[\alpha + 1 : N]$ we can get a new sequence $\bar{X}^* = U^\alpha \bar{X}[\alpha + 1 : N]$, which is exactly the $\bar{X}^*$ mentioned above.

For this sequence $\bar{X}^*$, it has the following properties:

- 1) Its output sequence ends with

$$f_E(F'_{i_{w-k+1}j_{w-k+1}}) \bar{F}_{i_{w-k+2}j_{w-k+2}} \dots f_E(\bar{F}_{i_w j_w})$$

- 2) $S^* = S / (i_{w-k+1}, j_{w-k+1})$.

So far, it is not hard to see that

$$f_E(F'_{i_{w-k+1}j_{w-k+1}}) \bar{F}_{i_{w-k+2}j_{w-k+2}} \dots f_E(\bar{F}_{i_w j_w})$$

is a suffix of Y' and $(i_m^*, j_m^*) = (i_m, j_m)$ for all $w - k + 1 \leq m \leq w$. As a result, the conclusion above is true for the $k+1^{th}$ iteration. By induction, we know that the conclusion is true for all $1 \leq k \leq w + 1$. Let $k = w + 1$, we can get that

$$f_E(\bar{F}_{i_1 j_1}) + \dots + f_E(\bar{F}_{i_w j_w}) = Y'$$

i.e., $X' = \bar{X}$ underlies Y' and this X' satisfies all the requirements in the lemma.

Now, we show that there are at most one such sequence X' satisfying all the requirements in the lemma. Assume there are two sequences satisfying all the requirements, such that their outputs are

$$f_E(G_{i_1 j_1}) f_E(G_{i_2 j_2}) \dots f_E(G_{i_w j_w})$$

and

$$f_E(H_{i'_1 j'_1}) f_E(H_{i'_2 j'_2}) \dots f_E(H_{i'_w j'_w})$$

where G_{ij} and H_{ij} are segments to produce outputs for the two input sequences.

By induction, we try to prove that for any $1 \leq k \leq w$, $i_k = i'_k, j_k = j'_k$ and $G_{i_k j_k} = H_{i'_k j'_k}$. When $k = w$, according to Lemma 4, it is not hard to get $i_w = i'_w, j_w = j'_w$. Therefore, $G_{i_w j_w} \equiv H_{i'_w j'_w}$. We also know that $f_E(G_{i_w j_w}) = f_E(H_{i'_w j'_w})$, so $G_{i_k j_k} = H_{i'_k j'_k}$. Finally, the conclusion is true for $k = w$. Assume the conclusion is true for all $l < k \leq w$, using the same argument as above, we can also get that the conclusion is true for l . As a result, the conclusion is true for all $1 \leq k \leq w$. Based on Lemma 2 for uniqueness, we can say that the two sequences are the same, i.e., the sequence X' is unique. ■

Theorem 9 (Algorithm B). *Let the sequence generated by a Markov chain be used as input to algorithm B, then Algorithm B can generate an independent unbiased sequence in expected linear time.*

Proof: In order to prove the theorem above, we can use the same method of the proof for Algorithm A. The only difference is that in the proof we use the mapping obtained in Lemma 8. ■

REFERENCES

- [1] J. von Neumann, "Various techniques used in connection with random digits", Appl. Math. Ser., Notes by G.E. Forstyle, Nat. Bur. Stand., vol. 12, pp. 36-38, 1951.
- [2] W. Hoeffding and G. Simon, "Unbiased coin tossing with a biased coin", Ann. Math. Statist., vol. 41, pp. 341-352, 1970.
- [3] Q. Stout and B. Warren, "Tree algorithms for unbiased coin tossing with a biased coin", Ann. Probab., vol. 12, pp. 212-222, 1984.
- [4] Y. Peres, "Iterating von Neumann's procedure for extracting random bits", Ann. Statist., vol. 20, pp. 590-597, 1992.
- [5] P. Elias, "The efficient construction of an unbiased random sequence", Ann. Math. Statist., vol. 43, pp. 865-870, 1972.
- [6] D. Knuth and A. Yao, "The complexity of nonuniform random number generation", Algorithms and Complexity: New Directions and Recent Results, pp. 357-428, 1976.
- [7] T.S. Han, M. Hoshi, "Interval algorithm for random number generation", IEEE Trans. on Information Theory, vol. 43, No. 2, pp. 599-611, 1997.
- [8] P.A. Samuelson, "Constructing an unbiased random sequence", J. Amer. Statist. Assoc., pp. 1526-1527, 1968.
- [9] M. Blum, "Independent unbiased coin flips from a correlated biased source: - a finite state Markov chain", Combinatorica, vol. 6, pp. 97-108, 1986.
- [10] S. Pae and M. C. Loui, "Optimal random number generation from a biased coin", in Proc. Sixteenth Annu. ACM-SIAM Symp. Discrete Algorithms, pp. 1079C1088, 2005.
- [11] H. Zhou and J. Bruck, "Efficiently Generating Random Bits from Finite State Markov Chains", Tech. Report at <http://paradise.caltech.edu/etr.html>.