

國立臺灣大學生物資源暨農學院農藝學研究所

博士論文

Graduate Institute of Agronomy

College of Bioresources and Agriculture

National Taiwan University

Doctoral Dissertation



Peres 位元亂數產生演算法及其串流版本之分析

Analysis of Peres' algorithm and its streaming versions for
random number generation

林昭京

Zhao Ging Randy Lim

指導教授: 姚怡慶 博士 廖振鐸 博士

Advisor: Yi-Ching Yao Ph.D. Chen-Tuo Liao Ph.D.

中華民國 109 年 12 月

December, 2020

國立臺灣大學博士學位論文

口試委員會審定書

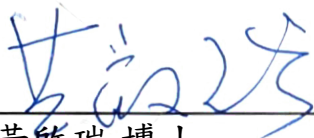


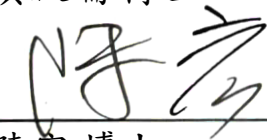
Peres 位元亂數產生演算法及其串流版本之分析

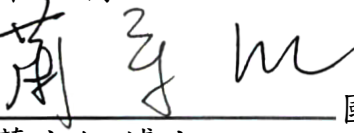
Analysis of Peres' algorithm and its streaming versions for
random number generation

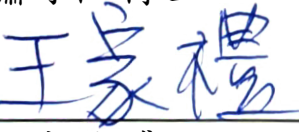
本論文係林昭京君 (D02621201) 在國立臺灣大學農藝學研究所完成之博士學位論文，於民國 109 年 12 月 21 日承下列考試委員審查通過及口試及格，特此證明

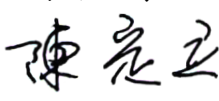
口試委員：

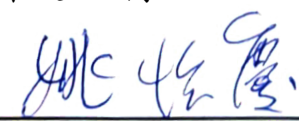

黃啟瑞 博士 中央研究院數學所研究員

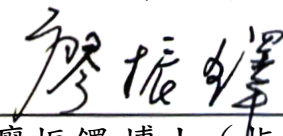

陳宏 博士 國立臺灣大學數學系教授


蕭守仁 博士 國立彰化師範大學數學系教授


王家禮 博士 國立東華大學應用數學系教授


陳定立 博士 中央研究院統計所副研究員


姚怡慶 博士 (指導教授) 中央研究院統計所研究員


廖振鐸 博士 (指導教授) 國立臺灣大學農藝學系教授





致謝

七年博士生涯就這樣過去了，腦海一幕幕的片段，都有著許多讓我感激不盡的人。姚怡慶老師是在學識上教導我最多的人，從初學機率論到課業上的難題，到最後幾年的探索與研究工作，姚老師都指導了我很多，而在最後的年頭開始對論文題材覺得厭倦時，也很慶幸還有姚老師在旁督促，最後還是拿到學位了，儘管如果可以更用心的話應該可以做得再好一些。姚老師也把我寫得亂七八糟脈絡不清的文章重新寫了一遍，把符號系統跟關鍵的性質建構得更容易讓初接觸 Peres 演算法的讀者掌握，我真的很敬佩姚老師的能力。如同我媽說的，人生難得遇見貴人，我無比感謝我的這位貴人姚老師。

我最終沒有沿着當初的計畫走著我的博士路途，但還是感謝在我前幾年忙著修課和滿懷希望申請國外學校時劉仁沛老師和《統計與生活》團隊經濟上的補助。雖然申請的學校都沒有接受我，但我還是感謝幫我寫推薦信的江金倉老師、劉豐哲老師、廖振鐸老師和蘇秀媛老師。我修習不少課程，算是補上做數學研究所該具備的知識，我由衷感謝實分析的劉豐哲老師、複變的王金龍老師、代數導論的李秋坤老師和陳其誠老師、常微分導論的林紹雄老師、偏微分導論的夏俊雄老師、高等微積分的陳金次老師、高等統計推論的江金倉老師、組合學的游森棚老師、機率論的王振男老師和姜祖恕老師，還有這些課的助教們，他們讓我獲益良多。我也感謝系上的蔡欣甫老師，他教導的線性模型鞏固了我的觀念。廖振鐸老師，他一直相信我可以做得很好，除了在經濟上補助我，也指點我方向

還推薦我給姚怡慶老師，我很感激廖老師對我這麼好。老師們的教導讓我認識了自己的方向，日後我也會好好學習，尤其是一些關於組合學（或許 continued fraction）、泛函分析和機率論的結合（或許 percolation），希望可以解決小小的問題（在 stackexchange 上發表也行）來報答老師的教誨。

博士期間我跟我生命中最重要的人房佑嬌結婚了，我們一直相知相惜，謝謝她給了我幸福的生活，跟著我這個長這麼大還沒錢沒房的人住在租來的房子，畢業後的日子我得更加照顧她。我們很少出門遊玩，但是那天在宜蘭睡海邊，我想到論文裡一個非常重要的結果該怎麼解決，原來跟她在一起也多了理性思考無法解釋的幸運。

謝謝你們。



摘要

對於利用投擲 p -偏銅板來產生獨立無偏的位元亂數，馮·諾伊曼（1951）提出了簡單的演算法，雖然這個演算法並非有效率地用上 p -偏銅板所蘊含的資訊熵，然而 Peres（1992）提出可疊代使用上述演算法，並證明了這樣的疊代演算法的亂數產生效率無限趨近於 p -偏銅板的資訊熵上界。確切來說，Peres 證明了 $\frac{1}{n}b(n, p)$ 均勻（於 $p \in (0, 1)$ ）收斂到 $h(p)$ ，其中 $b(n, p)$ 為 Peres 演算法應用於 n 次 p -偏銅板投擲結果 (X_1, \dots, X_n) 所產生的獨立無偏位元亂數個數的期望值，而 $h(p) = -p \log p - (1 - p) \log(1 - p)$ 是單次 p -偏銅板投擲的資訊熵。我們考慮了 $b(n, p)$ 的二階行為，即 $nh(p) - b(n, p)$ 於 n 趨近於無窮大時的漸近行為。在 $p = \frac{1}{2}$ 下，我們得到了 $\lim_{n \rightarrow \infty} \log[n - b(n, \frac{1}{2})] / \log n = \log[\frac{1+\sqrt{5}}{2}]$ 這樣的結果。我們也扼要地討論了當 $nh(p) - b(n, p)$ 在 n 趨近於無窮大時關於其行為的一些待解決問題。Peres 演算法在原來被提出時並非為串流演算法（streaming algorithm），亦即 (X_1, \dots, X_n) 所產生的亂數一般上並不一定會全被排在由 X_{n+1} 所產生的亂數之前。我們介紹了 Peres 演算法的二元樹表示，也進一步介紹了一類由二元樹上節點的排序所決定的 Peres 演算法的串流版本。在一般二元樹節點的排序下，Peres 演算法的串流版本並不會產生無偏的位元亂數列，然而在特定的排序下，我們證明了這樣的串流版本所產生的位元亂數列是無偏的。

關鍵字：資訊熵、演算法分析、Elias 提取器、Peres 提取器、馮·諾伊曼提取器、超可加性、串流演算法、狀態樹





Abstract

von Neumann (1951) introduced a simple algorithm for generating independent and unbiased random bits by tossing a coin of unknown bias p . While his algorithm fails to attain the entropy bound, Peres (1992) showed that the entropy bound can be attained asymptotically by iterating von Neumann's algorithm. Specifically, Peres showed that $\lim_{n \rightarrow \infty} \frac{1}{n} b(n, p) = h(p)$ uniformly in $p \in (0, 1)$, where $b(n, p)$ denotes the expected number of unbiased output bits generated when Peres' algorithm is applied to an input sequence (X_1, \dots, X_n) with X_i being the outcome of the i th coin toss, and $h(p) = -p \log p - (1 - p) \log(1 - p)$ (the Shannon entropy of each X_i). We consider the (second-order) behavior of $nh(p) - b(n, p)$ as $n \rightarrow \infty$. For $p = \frac{1}{2}$, it is shown that $\lim_{n \rightarrow \infty} \log[n - b(n, \frac{1}{2})] / \log n = \log[\frac{1+\sqrt{5}}{2}]$. Some open problems on the asymptotic behavior of $nh(p) - b(n, p)$ are briefly discussed. The original Peres' algorithm is not streaming in the sense that some of the output bits generated from (X_1, \dots, X_n) (the first n coin tosses) may be placed after the output bits induced by X_{n+1} . We introduce a binary tree representation of Peres' algorithm,

based on which we further introduce a class of streaming versions of Peres' algorithm in terms of orderings of the nodes of the binary tree. We show by example that in general a streaming version of Peres' algorithm fails to generate unbiased output bits. However, based on a special node ordering, the corresponding streaming version of Peres' algorithm is shown to be unbiased.

Keywords: entropy, analysis of algorithms, Elias' extractor, Peres' extractor, von Neumann's extractor, superadditivity, streaming algorithm, status tree



Contents

	Page
口試委員審定書	i
致謝	iii
摘要	v
Abstract	vii
Contents	ix
List of Figures	xi
Chapter 1 Introduction	1
1.1 von Neumann's algorithm	1
1.2 Elias' algorithm	2
1.3 Peres' algorithm	4
1.4 Binary tree representation of Peres' algorithm	6
1.5 Brief literature review	8
Chapter 2 Asymptotic Analysis of Peres' algorithm	13
2.1 Main results	13
2.2 Proofs of Propositions 2.1 and 2.2 and (2.1)	15
2.3 Numerical results and discussion	26
Chapter 3 Streaming versions of Peres' algorithm	33
3.1 Introduction and streaming algorithms	33

3.2	Status tree	37
3.3	Unbiasedness of \mathcal{A}_S	46
3.4	Zhou-Bruck's streaming version of Peres' algorithm	55
3.5	Counting status trees	65
Chapter 4	Concluding remarks	71
	References	73





List of Figures

Figure 1.1	Binary tree representation of \mathcal{A}_P	8
Figure 2.1	Plot of $\log g(n)/\log n$ versus n	27
Figure 2.2	Plot of $g(n)/n^\theta$ versus n with $\theta = \log[(1 + \sqrt{5})/2]$	27
Figure 2.3	Plot of $g(2n)/g(n)$ versus n	28
Figure 2.4	Plot of $g(3n)/g(n)$ versus n	28
Figure 2.5	Plot for $\text{Var}_{0.5} \mathcal{A}_P(\mathbf{X}_{2^k}) $	29
Figure 2.6	Plot for $2^k h(0.3) - \mathbb{E}_{0.3} \mathcal{A}_P(\mathbf{X}_{2^k}) $	30
Figure 2.7	Simulated histograms for the standardized distribution of $ \mathcal{A}_P(\mathbf{X}_n) $ for $p = 0.5$ and $n = 2^{10}, 2^{17}, 2^{25}$	31
Figure 3.1	The status tree \mathcal{S} given by $\mathcal{S}(\nu) = \text{O}$ for all nodes ν except $\mathcal{S}(\nu_{1000}) = \text{T}$, $\mathcal{S}(\nu_{1001}) = \text{H}$, $\mathcal{S}(\nu_{110}) = \text{H}$	46
Figure 3.2	An illustration of reconstructing \mathbf{x} given \mathcal{S}_x such that $\mathcal{S}_x(\nu_{(6)}) = \text{H}$, $\mathcal{S}_x(\nu_{(9)}) = \text{H}$, $\mathcal{S}_x(\nu_{(8)}) = \text{T}$, and $\mathcal{S}_x(\nu) = \text{O}$ for all $\nu \neq \nu_{(6)}, \nu_{(8)}, \nu_{(9)}$ and $\mathcal{A}_5(\mathbf{x}) = 00001$	53
Figure 3.3	An example of Zhou-Bruck $\{\text{H}, \text{T}, \text{O}, 1, 0\}$ -labeled status trees $\Sigma_{\mathbf{x}_i}$, $i = 1, \dots, 8$, where $\mathbf{x}_8 = \text{TTHTHHH}$	58





Chapter 1 Introduction

1.1 von Neumann's algorithm

In his seminal work [26], von Neumann introduced a simple algorithm \mathcal{A}_{vN} (also known as an extractor) for generating independent unbiased random bits by tossing a (possibly) biased coin of unknown bias. (A random bit is said to be unbiased if its value is 0 or 1 with equal probability.) Specifically, for $i = 1, 2, \dots$, let $X_i \in \{\text{H}, \text{T}\}$ denote the outcome of the i th toss of the coin, where H and T stand for heads and tails, respectively. Assume that the input sequence $\mathbf{X} = (X_1, X_2, \dots)$ is independent and identically distributed (iid) with $\mathbb{P}(X_i = \text{H}) = p = 1 - \mathbb{P}(X_i = \text{T})$ where the bias $p \in (0, 1)$ is unknown. (The coin is unbiased if $p = 1/2$.) The algorithm \mathcal{A}_{vN} divides \mathbf{X} into pairs $(X_1, X_2), (X_3, X_4), \dots$, discards those pairs of equal values, and then generates an infinite Bernoulli sequence $\mathcal{A}_{\text{vN}}(\mathbf{X})$ whose i th bit is either a 1 or a 0 according as the i th pair of unequal values is HT or TH. It is readily seen that $\mathcal{A}_{\text{vN}}(\mathbf{X})$ consists of iid unbiased bits.

Let \mathcal{A} denote a generic algorithm that generates independent unbiased bits from the sequence $\mathbf{X} = (X_1, X_2, \dots)$. Let $\mathcal{A}(\mathbf{X}_n)$ denote the set of unbiased bits generated by \mathcal{A} applied to $\mathbf{X}_n = (X_1, \dots, X_n)$, the outcomes of the first n tosses. Denote by $|\mathcal{A}(\mathbf{X}_n)|$ the cardinality of $\mathcal{A}(\mathbf{X}_n)$, which is an integer-valued random variable whose distribution

depends on n and p . We say that \mathcal{A} is nested if $\mathcal{A}(\mathbf{X}_{n_1}) \subset \mathcal{A}(\mathbf{X}_{n_2})$ whenever $n_1 < n_2$. We write $X \sim \text{binomial}(n, p)$ if a random variable X has the binomial distribution with parameters n and p . Then given $|\mathcal{A}_{\text{VN}}(\mathbf{X}_n)| = \ell$, the ℓ bits generated by \mathcal{A}_{VN} applied to $\mathbf{X}_n = (X_1, \dots, X_n)$ are (conditionally) independent unbiased. Moreover, $|\mathcal{A}_{\text{VN}}(\mathbf{X}_n)| \sim \text{binomial}(\lfloor \frac{n}{2} \rfloor, 2pq)$, where $q = 1 - p$ and $\lfloor x \rfloor$ denotes the largest integer not exceeding x . When \mathcal{A}_{VN} is applied to \mathbf{X}_n , the expected number of unbiased bits generated per toss equals $\mathbb{E}_p |\mathcal{A}_{\text{VN}}(\mathbf{X}_n)|/n = 2pq \lfloor \frac{n}{2} \rfloor/n$, which converges to pq as $n \rightarrow \infty$, where the subscript p in \mathbb{E}_p refers to the bias of each X_i . Note that pq is less than the entropy bound $h(p) := -p \log p - q \log q$ (the Shannon entropy of each X_i), where $\log = \log_2$ (the logarithm to base 2). This indicates that \mathcal{A}_{VN} does not make efficient use of information contained in X_1, X_2, \dots . It is also worth noting that \mathcal{A}_{VN} is nested.

1.2 Elias' algorithm

To improve the efficiency, Elias [5] presented a more sophisticated algorithm \mathcal{A}_E which generates unbiased bits from $\mathbf{X}_n = (X_1, \dots, X_n)$ by partitioning $\{\text{H}, \text{T}\}^n$ (the set of all possible realizations of \mathbf{X}_n) into disjoint subsets $S_{n,k} = \{\mathbf{x} \in \{\text{H}, \text{T}\}^n : |\mathbf{x}|_{\text{H}} = k\}$, $k = 0, 1, \dots, n$, with $|\mathbf{x}|_{\text{H}}$ and $|\mathbf{x}|_{\text{T}}$ being respectively the number of H's and the number of T's in \mathbf{x} . Write $|S_{n,k}| = \binom{n}{k} = \sum_{\ell=0}^{\lfloor \log \binom{n}{k} \rfloor} c_\ell 2^\ell$ with $c_\ell \in \{0, 1\}$ (binary representation of $\binom{n}{k}$). Then each $S_{n,k}$ is further partitioned as $S_{n,k} = \bigcup_{\{\ell: c_\ell=1\}} S_{n,k,\ell}$, where $|S_{n,k,\ell}| = 2^\ell$ for each ℓ with $c_\ell = 1$. Specify an assignment of 2^ℓ distinct (output) sequences of $\{0, 1\}^\ell$ to the 2^ℓ distinct sequences of $S_{n,k,\ell}$, so that if $\mathbf{X}_n \in S_{n,k,\ell}$, then an output sequence of ℓ bits is generated according to the assignment. While a naive implementation of Elias' algorithm requires an exponential memory size to make a table of assignment of output sequences, Ryabko and Matchikina [22] provided an efficient method to construct an assignment with

much reduced memory size and running time. Note that \mathcal{A}_E is not nested. In fact, when \mathcal{A}_E is applied to $\mathbf{X}_n = (X_1, \dots, X_n)$, all of X_1, \dots, X_n need to be observed before unbiased bits are generated. To show that \mathcal{A}_E attains the entropy bound asymptotically, Elias [5, equation (15)] proved that

$$\sum_{k=0}^n \binom{n}{k} p^k q^{n-k} \log \binom{n}{k} - 3 \leq \mathbb{E}_p |\mathcal{A}_E(\mathbf{X}_n)| \leq \sum_{k=0}^n \binom{n}{k} p^k q^{n-k} \log \binom{n}{k}. \quad (1.1)$$

Letting $\mathcal{H}(Z)$ denote the Shannon entropy of a random variable Z and noting that $|\mathbf{X}|_H \sim \text{binomial}(n, p)$, we have

$$\begin{aligned} nh(p) - \mathcal{H}(|\mathbf{X}|_H) &= -np \log p - nq \log q + \sum_{k=0}^n \binom{n}{k} p^k q^{n-k} \log \left[\binom{n}{k} p^k q^{n-k} \right] \\ &= \sum_{k=0}^n \binom{n}{k} p^k q^{n-k} \log \binom{n}{k}, \end{aligned}$$

from which it follows that (1.1) is equivalent to

$$\mathcal{H}(|\mathbf{X}|_H) \leq nh(p) - \mathbb{E}_p |\mathcal{A}_E(\mathbf{X}_n)| \leq \mathcal{H}(|\mathbf{X}|_H) + 3. \quad (1.2)$$

Since $\mathcal{H}(|\mathbf{X}|_H) = \frac{1}{2} \log n + \frac{1}{2} \log e + \log \sqrt{2\pi pq} + O(\frac{1}{n})$ (cf. [8]), we have

$$nh(p) - \mathbb{E}_p |\mathcal{A}_E(\mathbf{X}_n)| = \frac{1}{2} \log n + O(1). \quad (1.3)$$

Consequently, $\lim_{n \rightarrow \infty} \mathbb{E}_p |\mathcal{A}_E(\mathbf{X}_n)|/n = h(p)$. Later Pae and Loui [19] established the exact optimality of \mathcal{A}_E that for any algorithm \mathcal{A} , $\mathbb{E}_p |\mathcal{A}_E(\mathbf{X}_n)| \geq \mathbb{E}_p |\mathcal{A}(\mathbf{X}_n)|$ for all $p \in (0, 1)$ and $n \geq 1$.



1.3 Peres' algorithm

While \mathcal{A}_{VN} fails to attain the entropy bound asymptotically, Peres [20] pointed out that the entropy bound can be attained asymptotically by iterating \mathcal{A}_{VN} . To describe Peres' ingenious idea, let $\mathbf{X} = (X_1, X_2, \dots)$ be decomposed into three infinite Bernoulli sequences $\mathcal{A}_{\text{VN}}(\mathbf{X})$, $\lambda(\mathbf{X})$, and $\rho(\mathbf{X})$, where the i th bit of $\lambda(\mathbf{X})$ is T or H according as the i th pair (X_{2i-1}, X_{2i}) is of equal values or of unequal values, and the i th bit of $\rho(\mathbf{X})$ is the common value of the i th pair of equal values. As an example, let

$$\mathbf{x} = \text{THHTHHHTTHTTT} \cdots = \text{TH HT HH TT HT TT} \cdots$$

Then

$$\lambda(\mathbf{x}) = \text{HHTTHT} \cdots, \quad \rho(\mathbf{x}) = \text{HTT} \cdots, \quad \text{and} \quad \mathcal{A}_{\text{VN}}(\mathbf{x}) = 011 \cdots.$$

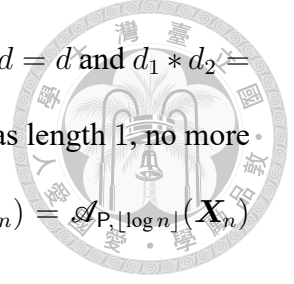
(Here and below, we take the convention that capital \mathbf{X} denotes a random sequence and lower case \mathbf{x} is a realization.) It is readily seen that (i) $\lambda(\mathbf{X})$, $\rho(\mathbf{X})$ and $\mathcal{A}_{\text{VN}}(\mathbf{X})$ are mutually independent, (ii) $\lambda(\mathbf{X})$, $\rho(\mathbf{X})$ and $\mathcal{A}_{\text{VN}}(\mathbf{X})$ are each an iid Bernoulli sequence with respective biases $f_\lambda(p) := 2pq$, $f_\rho(p) := p^2/(p^2 + q^2)$ and $1/2$, (iii) \mathbf{X} can be recovered from $\lambda(\mathbf{X})$, $\rho(\mathbf{X})$ and $\mathcal{A}_{\text{VN}}(\mathbf{X})$, implying that they together contain all information in \mathbf{X} . The first iteration of Peres' algorithm \mathcal{A}_{P} yields $\lambda(\mathbf{X})$, $\rho(\mathbf{X})$ and $\mathcal{A}_{\text{VN}}(\mathbf{X})$. Letting $\psi = \lambda$ or ρ , on the second iteration of \mathcal{A}_{P} , $\psi(\mathbf{X}) (= \lambda(\mathbf{X})$ or $\rho(\mathbf{X}))$ is further decomposed into three iid Bernoulli sequences $\lambda(\psi(\mathbf{X}))$, $\rho(\psi(\mathbf{X}))$ and $\mathcal{A}_{\text{VN}}(\psi(\mathbf{X}))$ with respective biases $f_\lambda(f_\psi(p))$, $f_\rho(f_\psi(p))$ and $1/2$. Thus, after 2 iterations, there are $7 (= 2^3 - 1)$ Bernoulli sequences, $\mathcal{A}_{\text{VN}}(\mathbf{X})$, $\mathcal{A}_{\text{VN}}(\lambda(\mathbf{X}))$, $\mathcal{A}_{\text{VN}}(\rho(\mathbf{X}))$, $\lambda(\lambda(\mathbf{X}))$, $\lambda(\rho(\mathbf{X}))$, $\rho(\lambda(\mathbf{X}))$, and $\rho(\rho(\mathbf{X}))$. The first three have bias $1/2$. More generally, after η

iterations ($\eta = 1, 2, \dots$), there are $2^{\eta+1} - 1$ Bernoulli sequences, $2^\eta - 1$ of which have bias $1/2$. We refer to the $2^\eta - 1$ Bernoulli sequences having bias $1/2$ as *unbiased* Bernoulli sequences, and refer to the other 2^η Bernoulli sequences as *biased* Bernoulli sequences. Note that the $2^{\eta+1} - 1$ Bernoulli sequences are all mutually independent, from which \mathbf{X} can be recovered.

We now consider the finite setting where only the first n terms of the infinite (input) sequence \mathbf{X} are available. Let $\psi = \lambda$ or ρ or \mathcal{A}_{vN} . Then $\mathbf{X}_n = (X_1, \dots, X_n)$, the subsequence of \mathbf{X} consisting of the first n terms, induces a finite sequence $\psi(\mathbf{X}_n)$ consisting of the first n_ψ terms of $\psi(\mathbf{X})$ for some n_ψ . Specifically, $n_\lambda = \lfloor n/2 \rfloor$, i.e. $\lambda(\mathbf{X}_n)$ consists of the first $\lfloor n/2 \rfloor$ terms of $\lambda(\mathbf{X})$. Moreover, $n_\rho = |\lambda(\mathbf{X}_n)|_{\text{T}}$ and $n_{\mathcal{A}_{\text{vN}}} = |\lambda(\mathbf{X}_n)|_{\text{H}}$. We have $n_\lambda = n_\rho + n_{\mathcal{A}_{\text{vN}}} = \lfloor \frac{n}{2} \rfloor$, $n_\rho \sim \text{binomial}(\lfloor \frac{n}{2} \rfloor, p^2 + q^2)$ and $n_{\mathcal{A}_{\text{vN}}} \sim \text{binomial}(\lfloor \frac{n}{2} \rfloor, 2pq)$. While the infinite sequences $\psi(\mathbf{X})$, $\psi = \lambda, \rho, \mathcal{A}_{\text{vN}}$, are mutually independent, the subsequences $\psi(\mathbf{X}_n)$'s are no longer independent. However, it is readily seen that, given the value of $n_{\mathcal{A}_{\text{vN}}}$, the bits in $\mathcal{A}_{\text{vN}}(\mathbf{X}_n)$ are (conditionally) independent unbiased. In fact, given the values of the bits in $\lambda(\mathbf{X}_n)$ and $\rho(\mathbf{X}_n)$, the bits in $\mathcal{A}_{\text{vN}}(\mathbf{X}_n)$ remain (conditionally) independent unbiased. Furthermore, for even n , \mathbf{X}_n can be recovered from $\psi(\mathbf{X}_n)$, $\psi = \lambda, \rho, \mathcal{A}_{\text{vN}}$, but for odd n , the last term of \mathbf{X}_n cannot be recovered, resulting in a loss of information. After η iterations ($\eta = 1, 2, \dots$), \mathbf{X}_n induces a (possibly empty) subsequence of each of the $2^{\eta+1} - 1$ infinite Bernoulli sequences as decomposed from \mathbf{X} .

For $\eta = 1, 2, \dots$, let $\mathcal{A}_{\text{P},\eta}(\mathbf{X}_n)$ denote the total collection of unbiased bits after η iterations. Then $\mathcal{A}_{\text{P},1}(\mathbf{X}_n) = \mathcal{A}_{\text{vN}}(\mathbf{X}_n)$ and $\eta \geq 2$, we have the following recursion

$$\mathcal{A}_{\text{P},\eta}(\mathbf{X}_n) = \mathcal{A}_{\text{vN}}(\mathbf{X}_n) * \mathcal{A}_{\text{P},\eta-1}(\lambda(\mathbf{X}_n)) * \mathcal{A}_{\text{P},\eta-1}(\rho(\mathbf{X}_n)), \quad (1.4)$$



where $*$ is the *concatenation* (binary) operator defined by $d * \emptyset = \emptyset * d = d$ and $d_1 * d_2 = d_1 d_2$. Since after $\lfloor \log n \rfloor$ iterations, the longest biased subsequence has length 1, no more unbiased bits can be produced by further iteration. We have $\mathcal{A}_{P,\eta}(\mathbf{X}_n) = \mathcal{A}_{P,\lfloor \log n \rfloor}(\mathbf{X}_n)$ for $\eta \geq \lfloor \log n \rfloor$. Let $\mathcal{A}_P(\mathbf{X}_n) = \mathcal{A}_{P,\lfloor \log n \rfloor}(\mathbf{X}_n)$, so that

$$\mathcal{A}_P(\mathbf{X}_n) = \mathcal{A}_{vN}(\mathbf{X}_n) * \mathcal{A}_P(\lambda(\mathbf{X}_n)) * \mathcal{A}_P(\rho(\mathbf{X}_n)). \quad (1.5)$$

Consider again the example where $\mathbf{x} = \text{THHTHHTTHTTT} \dots$. For $n = 12$, we have

$$\begin{aligned} \mathcal{A}_{P,1}(\mathbf{x}_{12}) &= 011, & \mathcal{A}_{P,2}(\mathbf{x}_{12}) &= 01111, & \text{and} \\ \mathcal{A}_P(\mathbf{x}_{12}) &= \mathcal{A}_{P,3}(\mathbf{x}_{12}) &= 011111. \end{aligned}$$

It is shown in Peres [20] that (i) for each η , given $|\mathcal{A}_{P,\eta}(\mathbf{X}_n)| = \ell$, the ℓ bits in $\mathcal{A}_{P,\eta}(\mathbf{X}_n)$ are independent unbiased, (ii) the rates $r_\eta(p) := \lim_{n \rightarrow \infty} \mathbb{E}_p |\mathcal{A}_{P,\eta}(\mathbf{X}_n)|/n$ satisfy $r_1(p) = pq$ and the recursion

$$r_\eta(p) = pq + \frac{1}{2} r_{\eta-1}(2pq) + \frac{1}{2} (p^2 + q^2) r_{\eta-1} \left(\frac{p^2}{p^2 + q^2} \right) \quad \text{for } \eta \geq 2, \quad (1.6)$$

and (iii) $r_\eta(p)$ increases as $\eta \rightarrow \infty$ to $h(p)$ uniformly in $p \in (0, 1)$. As a consequence, $\mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|/n \rightarrow h(p)$ as $n \rightarrow \infty$, showing that \mathcal{A}_P attains the entropy bound asymptotically. Moreover, \mathcal{A}_P is nested.

1.4 Binary tree representation of Peres' algorithm

It is instructive to describe the iterations of Peres' algorithm via a rooted (infinite) complete binary tree T which is identified with a sequence of nodes $(\nu_1, \nu_{10}, \nu_{11}, \nu_{100}, \nu_{101}, \dots)$.

Here ν_1 is the root node, whose left and right child nodes are denoted by ν_{10} and ν_{11} , respectively. More generally, for $u = b_1 b_2 \dots b_\eta$ (a string of η bits with $b_1 = 1$ and $\eta \geq 1$), the node ν_u has two (left and right) child nodes denoted by ν_{u0} and ν_{u1} . It is also convenient to identify $u = b_1 \dots b_\eta$, with the positive integer $v = \sum_{i=1}^{\eta} b_i 2^{\eta-i}$ and write $\nu_u = \nu_{(v)}$. Thus we have $(\nu_1, \nu_{10}, \nu_{11}, \nu_{100}, \dots) = (\nu_{(1)}, \nu_{(2)}, \nu_{(3)}, \nu_{(4)}, \dots)$. It is readily seen that $\nu_{(v)}$ has two (left and right) child nodes denoted by $\nu_{(2v)}$ and $\nu_{(2v+1)}$, respectively.

We now describe Peres' algorithm as follows. The input (finite or infinite) sequence $\mathbf{X} = (X_1, X_2, \dots)$ arrives at the root node ν_1 , which is decomposed into $\lambda(\mathbf{X})$, $\rho(\mathbf{X})$ and $\mathcal{A}_{\text{vN}}(\mathbf{X})$. The first two sequences $\lambda(\mathbf{X})$ and $\rho(\mathbf{X})$ become the (derived or induced) input sequences at node ν_{10} and ν_{11} , respectively, while $\mathcal{A}_{\text{vN}}(\mathbf{X})$ is the output sequence at ν_1 . On the second iteration of Peres' algorithm, the input sequence $\lambda(\mathbf{X})$ at ν_{10} is decomposed into $\lambda(\lambda(\mathbf{X})) = \lambda^2(\mathbf{X})$, $\rho(\lambda(\mathbf{X})) = \rho\lambda(\mathbf{X})$, and $\mathcal{A}_{\text{vN}}(\lambda(\mathbf{X}))$, which become, respectively, the input sequence at node ν_{100} , the input sequence at node ν_{101} , and the output sequence at node ν_{10} . Similarly, the input sequence $\rho(\mathbf{X})$ at node ν_{11} is decomposed into $\lambda(\rho(\mathbf{X})) = \lambda\rho(\mathbf{X})$, $\rho(\rho(\mathbf{X})) = \rho^2(\mathbf{X})$ and $\mathcal{A}_{\text{vN}}(\rho(\mathbf{X}))$, which become, respectively, the input sequence at node ν_{110} , the input sequence at node ν_{111} , and the output sequence at ν_{11} . More generally, on the η th iteration ($\eta \geq 2$), the input sequence $\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{X})$ at node $\nu_{b_1 b_2 \dots b_\eta}$ where $\psi_{b_i} = \lambda$ or ρ according as $b_i = 0$ or 1 , is decomposed into $\lambda\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{X})$, $\rho\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{X})$ and $\mathcal{A}_{\text{vN}}(\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{X}))$, which become, respectively, the input sequence at node $\nu_{b_1 b_2 \dots b_\eta 0}$, the input sequence at node $\nu_{b_1 b_2 \dots b_\eta 1}$, and the output sequence at node $\nu_{b_1 b_2 \dots b_\eta}$. See Figure 1.1.

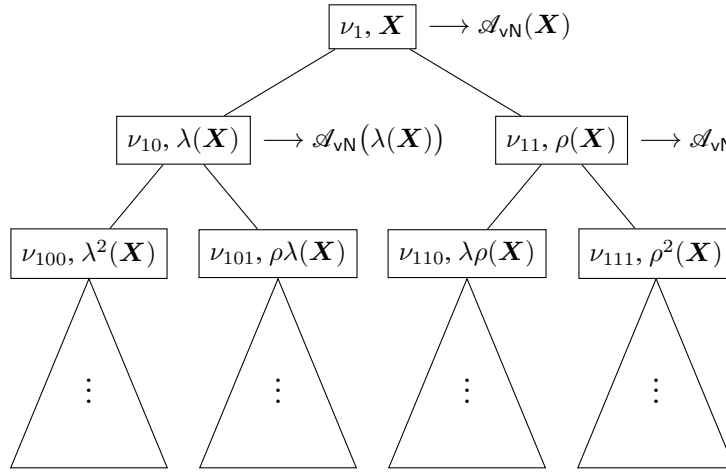


Figure 1.1: Binary tree representation of \mathcal{A}_p .

1.5 Brief literature review

In addition to the papers mentioned earlier, a number of papers in the literature also discuss von Neumann's problem or related problems. Hoeffding and Simons [7] considered the problem of generating an unbiased bit by a stopping time when X_1, X_2, \dots (the outcomes from repeatedly tossing a coin of unknown bias p) are observed sequentially. Note that this problem may be referred to as in the variable-to-fixed length regime as opposed to the fixed-to-variable length regime where the number of input bits is fixed and the number of output bits is random. (In particular, Elias' algorithm is in the fixed-to-variable length regime.) More specifically, their problem is to find an algorithm consisting of a stopping time τ and a function $f : \{H, T\}^* \rightarrow \{0, 1\}$, where $\{H, T\}^* = \bigcup_{n=0}^{\infty} \{H, T\}^n$, such that

$$\mathbb{P}_p(f(X_1, \dots, X_\tau) = 0) = \mathbb{P}_p(f(X_1, \dots, X_\tau) = 1) = \frac{1}{2}, \quad \text{for all } p \in (0, 1).$$

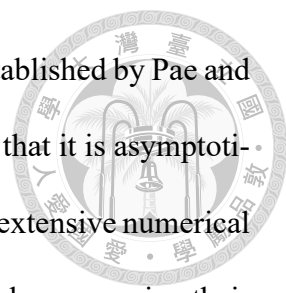
The objective is to choose τ such that $\mathbb{E}_p \tau$ is as small as possible for all p . Note that the stopping time τ_{vN} corresponding to von Neumann's algorithm is $\tau_{vN} = 2 \inf\{n : X_{2n-1} \neq$

X_{2n} } for which $\mathbb{E}_p \tau_{\vee N} = 1/pq \geq 4$. Dwass [4] and Bernard and Letac [1] considered a more general problem of generating an r -valued equiprobable random variable by an algorithm involving a stopping time when X_1, X_2, \dots are iid with an unknown discrete distribution. By representing algorithms in terms of trees, Stout and Warren [25] were able to improve the algorithms in Hoeffding and Simons [7], Dwass [4], and Bernard and Letac [1].

Knuth and Yao [11] considered the problem of generating a random variable with a given target discrete distribution when the input sequence consists of iid unbiased bits. They developed optimal algorithms which minimize the expected number of input bits required to generate a desired random variable. Han and Hoshi [6] generalized their approach to the setting where the common (discrete) distribution of the input sequence is general. See [15] and [27] for recent development.

Samuelson [23] studied the problem of generating an unbiased bit when the input sequence is a Markov chain. By considering transitions out of a specific state, he first constructed an iid sequence (with an unknown common discrete distribution) from which iid unbiased bits can then be generated. Elias [5] and Blum [2] also considered this problem. By generalizing Blum's algorithm together with Elias' method, Zhou and Bruck [29] provided an algorithm that generates unbiased random bits from arbitrary finite Markov chains, operates in expected linear time and attains asymptotically the information-theoretic upper bound on efficiency. In the fixed-to-variable length regime, Seroussi and Weinberger [24] derived a second-order term (a term keeping the expected number of output bits below the entropy of the input) for an optimal algorithm in the Markov setting.

As discussed in Sections 1.2 and 1.3 where the input sequence is iid Bernoulli, Elias



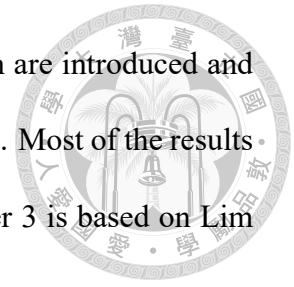
[5] proposed an efficient algorithm \mathcal{A}_E whose exact optimality was established by Pae and Loui [19]. Peres [20] presented an elegant algorithm \mathcal{A}_P and showed that it is asymptotically efficient. Prasitsupparote, Konno and Shikata [21] conducted an extensive numerical study of \mathcal{A}_P and \mathcal{A}_E (with the method of Ryabko and Matchikina [22]) by comparing their memory and running time requirements. Their empirical results suggest that \mathcal{A}_P would be superior to \mathcal{A}_E in practical applications. Pae [17] and Pae [18] generalized Peres' algorithm when the input iid sequence X_1, X_2, \dots has a common (unknown) distribution on $\{0, 1, \dots, r-1\}$ with $r > 1$. Zhou and Bruck [30] proposed a streaming version of Peres' algorithm in the variable-to-fixed length regime where a given number of iid unbiased bits is to be generated by a random number τ of (input) X_i 's with τ being a stopping time. Their streaming algorithm will be discussed in Chapter 3.

Von Neumann's algorithm has been used to remove bias in the output of a true random number generator from a variety of physical devices (see e.g. Wei and Guo [28]). It has applications in cryptography to generate random cryptographic keys for secure data transmission.

The so-called Bernoulli factory refers to the problem of using a p -coin (coin with probability of heads p) to simulate an $f(p)$ -coin (coin with probability of heads $f(p)$) where (unknown) p is known only to belong to a given subset S of $(0, 1)$ and the function $f : S \rightarrow [0, 1]$ is known. The problem considered in this dissertation deals with the special case that $S = (0, 1)$ and $f(p) = 1/2$ for all p . Keane and O'Brien [10] obtained necessary and sufficient conditions on f under which there exists an algorithm to simulate an $f(p)$ -coin using a p -coin. See also Nacu and Peres [14] for related results.

In Chapter 2, we present an asymptotic analysis of Peres' algorithm along with nu-

merical results. In Chapter 3, streaming versions of Peres' algorithm are introduced and their properties are discussed. Chapter 4 contains concluding remarks. Most of the results in Chapter 2 have appeared in Lim, Liao and Yao [12] while Chapter 3 is based on Lim and Yao [13].







Chapter 2 Asymptotic Analysis of Peres' algorithm

2.1 Main results

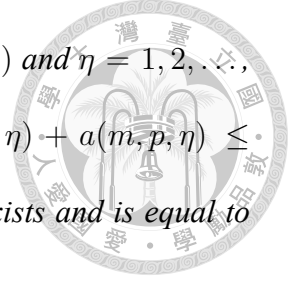
While \mathcal{A}_E and \mathcal{A}_P both attain the entropy bound asymptotically, (1.2) and (1.3) provide a precise (second-order) behavior of $nh(p) - \mathbb{E}_p |\mathcal{A}_E(\mathbf{X}_n)|$. In contrast, there is not much known about the behavior of $nh(p) - \mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|$ for large n . In this regard, Pae [16] gave a formula to compute $\mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|$, which is not convenient for deriving the asymptotic behavior of $nh(p) - \mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|$. Recently, Prasitsupparote *et al.* [21] showed, based on some heuristics, that for $p = 1/2$,

$$nh(p) - \mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)| = n - \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)| \geq n^{\log 3 - 1}. \quad (2.1)$$

To derive (2.1), they assumed, without rigorous justification, that

$$\frac{1}{n} \mathbb{E}_p |\mathcal{A}_{P,\eta}(\mathbf{X}_n)| \leq r_\eta(p) \quad \text{for } p \in (0, 1), n \geq 1, \eta \geq 1. \quad (2.2)$$

In Section 2.2, we prove the following results while numerical results are presented in Section 2.3.



Proposition 2.1. Let $a(n, p, \eta) = \mathbb{E}_p |\mathcal{A}_{P, \eta}(\mathbf{X}_n)|$. Then for $p \in (0, 1)$ and $\eta = 1, 2, \dots$, the sequence $(a(1, p, \eta), a(2, p, \eta), \dots)$ is superadditive, i.e. $a(n, p, \eta) + a(m, p, \eta) \leq a(n + m, p, \eta)$ for $n, m \geq 1$. Consequently, $\lim_{n \rightarrow \infty} a(n, p, \eta)/n$ exists and is equal to $\sup_{n \geq 1} a(n, p, \eta)/n$. That is,

$$r_\eta(p) := \lim_{n \rightarrow \infty} \mathbb{E}_p |\mathcal{A}_{P, \eta}(\mathbf{X}_n)|/n = \sup_{n \geq 1} \mathbb{E}_p |\mathcal{A}_{P, \eta}(\mathbf{X}_n)|/n,$$

which implies (2.2).

Proposition 2.2. For $p = 1/2$, let $b(n) = \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|$. Then

(i) the $b(n)$ satisfy $b(0) = b(1) = 0$ and the recursion

$$b(n) = \lfloor \frac{n}{2} \rfloor / 2 + b\left(\lfloor \frac{n}{2} \rfloor\right) + \mathbb{E} b(B_{\lfloor n/2 \rfloor, 1/2}) \quad \text{for } n = 2, 3, \dots, \quad (2.3)$$

where $B_{n,p}$ denotes a binomial(n, p) random variable;

(ii)

$$\lim_{n \rightarrow \infty} \frac{\log(n - b(n))}{\log n} = \log\left(\frac{1 + \sqrt{5}}{2}\right).$$

Note that with $p = 1/2$, the coin is unbiased and the input sequence consists on n unbiased bits, so that $n - b(n)$ may be referred to as the cost incurred by \mathcal{A}_P when not knowing $p = 1/2$. The next section contains the proofs of Propositions 2.1 and 2.2. In addition, for completeness, a rigorous proof of (2.1) is also given, which is needed for the proof of Proposition 2.2(ii). Most of the results in this chapter have appeared in Lim *et al.* [12].



2.2 Proofs of Propositions 2.1 and 2.2 and (2.1)

Proof of Proposition 2.1. Recall that when \mathcal{A}_P is applied to $\mathbf{X}_n = (X_1, \dots, X_n)$, 3 subsequences $\lambda(\mathbf{X}_n)$, $\rho(\mathbf{X}_n)$ and $\mathcal{A}_{\vee N}(\mathbf{X}_n)$ are induced where $|\lambda(\mathbf{X}_n)| = |\rho(\mathbf{X}_n)| + |\mathcal{A}_{\vee N}(\mathbf{X}_n)| = \lfloor n/2 \rfloor$ and $|\mathcal{A}_{\vee N}(\mathbf{X}_n)| \sim \text{binomial}(\lfloor n/2 \rfloor, 2pq)$. It follows from (1.4) that

$$\begin{aligned} a(n, p, \eta) &= \mathbb{E}_p |\mathcal{A}_{P, \eta}(\mathbf{X}_n)| \\ &= \mathbb{E}_p |\mathcal{A}_{P, \eta-1}(\lambda(\mathbf{X}_n))| + \mathbb{E}_p |\mathcal{A}_{P, \eta-1}(\rho(\mathbf{X}_n))| + 2pq \lfloor n/2 \rfloor. \end{aligned} \quad (2.4)$$

Noting that $\lambda(\mathbf{X}_n)$ is distributed as a sequence \mathbf{U} of $n_\lambda = \lfloor n/2 \rfloor$ iid Bernoulli random variables with bias $f_\lambda(p)$, we have

$$\mathbb{E}_p |\mathcal{A}_{P, \eta-1}(\lambda(\mathbf{X}_n))| = \mathbb{E}_{f_\lambda(p)} |\mathcal{A}_{P, \eta-1}(\mathbf{U})| = a(\lfloor n/2 \rfloor, f_\lambda(p), \eta - 1). \quad (2.5)$$

Similarly, conditioning on $n_\rho = |\lambda(\mathbf{X}_n)|_T$, $\rho(\mathbf{X}_n)$ is a sequence of n_ρ iid Bernoulli random variables with bias $f_\rho(p)$, so that the conditional expectation of $|\mathcal{A}_{P, \eta-1}(\rho(\mathbf{X}_n))|$ given n_ρ equals $a(n_\rho, f_\rho(p), \eta - 1)$. Since $n_\rho \sim \text{binomial}(\lfloor n/2 \rfloor, 1 - 2pq)$, we have

$$\mathbb{E}_p |\mathcal{A}_{P, \eta-1}(\rho(\mathbf{X}_n))| = \mathbb{E} a(B_{\lfloor n/2 \rfloor, 1-2pq}, f_\rho(p), \eta - 1), \quad (2.6)$$

where the expectation operator \mathbb{E} on the right-hand side is on $B_{\lfloor n/2 \rfloor, 1-2pq}$ (a binomial($\lfloor n/2 \rfloor$, $1 - 2pq$) random variable). By (2.4), (2.5) and (2.6),

$$a(n, p, \eta) = a(\lfloor n/2 \rfloor, f_\lambda(p), \eta - 1) + \mathbb{E} a(B_{\lfloor n/2 \rfloor, 1-2pq}, f_\rho(p), \eta - 1) + 2pq \lfloor n/2 \rfloor. \quad (2.7)$$

We now prove by induction on η that

$$a(n, p, \eta) + a(m, p, \eta) \leq a(n + m, p, \eta). \quad (2.8)$$

For $\eta = 1$, $a(n, p, 1) = 2pq \lfloor n/2 \rfloor$. Since $\lfloor n/2 \rfloor + \lfloor m/2 \rfloor \leq \lfloor (n+m)/2 \rfloor$ for $n, m \geq 1$, we have $a(n, p, 1) + a(m, p, 1) \leq a(n+m, p, 1)$, implying that (2.8) holds for $\eta = 1$. Suppose that for an integer $k > 0$, (2.8) holds for all $n, m \geq 1$, all $p \in (0, 1)$, and $\eta = k$. We need to show that (2.8) holds for $n, m \geq 1$, $p \in (0, 1)$ and $\eta = k+1$. By the induction hypothesis,

$$\begin{aligned} a(\lfloor n/2 \rfloor, f_\lambda(p), k) + a(\lfloor m/2 \rfloor, f_\lambda(p), k) &\leq a(\lfloor n/2 \rfloor + \lfloor m/2 \rfloor, f_\lambda(p), k) \\ &\leq a(\lfloor (n+m)/2 \rfloor, f_\lambda(p), k), \end{aligned} \quad (2.9)$$

where the second inequality follows from the fact that $a(n, p, \eta)$ is non-decreasing in n . Let U and V be independent random variables with $U \sim \text{binomial}(\lfloor n/2 \rfloor, 1 - 2pq)$ and $V \sim \text{binomial}(\lfloor m/2 \rfloor, 1 - 2pq)$. Then $U + V \sim \text{binomial}(\lfloor n/2 \rfloor + \lfloor m/2 \rfloor, 1 - 2pq)$. If n and m are both odd, let W be independent of U and V with $W \sim \text{binomial}(1, 1 - 2pq)$. If at least one of n and m is even, let W be identically 0. Then $U + V + W \sim \text{binomial}(\lfloor (n+m)/2 \rfloor, 1 - 2pq)$. We have by the induction hypothesis that

$$\begin{aligned} &\mathbb{E} a(B_{\lfloor n/2 \rfloor, 1-2pq}, f_\rho(p), k) + \mathbb{E} a(B_{\lfloor m/2 \rfloor, 1-2pq}, f_\rho(p), k) \\ &= \mathbb{E} \{a(U, f_\rho(p), k) + a(V, f_\rho(p), k)\} \\ &\leq \mathbb{E} a(U + V, f_\rho(p), k) \\ &\leq \mathbb{E} a(U + V + W, f_\rho(p), k) \\ &= \mathbb{E} a(B_{\lfloor (n+m)/2 \rfloor, 1-2pq}, f_\rho(p), k). \end{aligned} \quad (2.10)$$

Moreover,

$$2pq \lfloor n/2 \rfloor + 2pq \lfloor m/2 \rfloor \leq 2pq \lfloor (n+m)/2 \rfloor. \quad (2.11)$$



By (2.7) and (2.9)–(2.11),

$$a(n, p, k + 1) + a(m, p, k + 1) \leq a(n + m, p, k + 1),$$

showing that (2.8) holds for $n, m \geq 1, p \in (0, 1)$ and $\eta = k + 1$. The proof is complete. \square

Proof of (2.1). The following argument is taken from the proof of Theorem 1 in Prasit-supparote *et al.* [21]. With $p = 1/2$, we have $r_1(1/2) = pq = \frac{1}{4}$ and, by (1.6)

$$r_\eta(1/2) = \frac{1}{4} + \frac{3}{4}r_{\eta-1}(1/2) \quad \text{for } \eta \geq 2,$$

from which it follows that $r_\eta(1/2) = 1 - (\frac{3}{4})^\eta, \eta \geq 1$. By Proposition 2.1,

$$1 - \left(\frac{3}{4}\right)^\eta = r_\eta(1/2) \geq \mathbb{E}_{1/2} |\mathcal{A}_{P,\eta}(\mathbf{X}_n)|/n,$$

so that with $\eta = \lfloor \log n \rfloor$ and $b(n) = \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|$, we have

$$1 - \left(\frac{3}{4}\right)^{\lfloor \log n \rfloor} \geq \mathbb{E}_{1/2} |\mathcal{A}_{P,\lfloor \log n \rfloor}(\mathbf{X}_n)|/n = \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|/n = b(n)/n,$$

implying that

$$n - b(n) \geq n \left(\frac{3}{4}\right)^{\lfloor \log n \rfloor} \geq n \left(\frac{3}{4}\right)^{\log n} = n^{\log 3 - 1},$$

proving (2.1). \square

Proof of Proposition 2.2(i). For $p = 1/2, f_\lambda(1/2) = f_\rho(1/2) = 1/2$, and $B_{\lfloor n/2 \rfloor, 1-2pq} = B_{\lfloor n/2 \rfloor, 1/2}$. Letting $a(n, p, \eta) = \mathbb{E}_p |\mathcal{A}_{P,\eta}(\mathbf{X}_n)|$, we have by (2.7) that

$$a(n, 1/2, \eta) = a(\lfloor n/2 \rfloor, 1/2, \eta - 1) + \mathbb{E} a(B_{\lfloor n/2 \rfloor, 1/2}, 1/2, \eta - 1) + \lfloor n/2 \rfloor / 2. \quad (2.12)$$



Recall that $\mathcal{A}_P(\mathbf{X}_n) = \mathcal{A}_{P,\eta}(\mathbf{X}_n)$ for $\eta \geq \lfloor \log n \rfloor$. By (2.12),

$$\begin{aligned}
 b(n) &= \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)| = \mathbb{E}_{1/2} |\mathcal{A}_{P,\lfloor \log n \rfloor}(\mathbf{X}_n)| \\
 &= a(n, 1/2, \lfloor \log n \rfloor) \\
 &= a(\lfloor n/2 \rfloor, 1/2, \lfloor \log n \rfloor - 1) + \mathbb{E} a(B_{\lfloor n/2 \rfloor, 1/2}, 1/2, \lfloor \log n \rfloor - 1) + \lfloor n/2 \rfloor / 2 \\
 &= b(\lfloor n/2 \rfloor) + \mathbb{E} b(B_{\lfloor n/2 \rfloor, 1/2}) + \lfloor n/2 \rfloor / 2,
 \end{aligned}$$

proving (2.3). □

To prove Proposition 2.2(ii), we need the following lemmas. Proposition 2.2(ii) follows immediately from Lemmas 2.4 and 2.5 below. For the rest of this section, to simplify notation, we write $B_n = B_{n,1/2}$ for a binomial($n, 1/2$) random variable. Let $g(n) = n - b(n) \geq 0$, for $n = 0, 1, \dots$. We have $g(0) = 0$, $g(1) = 1$, and by Proposition 2.2(i), for even $n \geq 0$,

$$\begin{aligned}
 g(n) = n - b(n) &= n - \left[\frac{n}{4} + b\left(\frac{n}{2}\right) + \mathbb{E} b(B_{\frac{n}{2}}) \right] \\
 &= \left[\frac{n}{2} - b\left(\frac{n}{2}\right) \right] + \mathbb{E} [B_{\frac{n}{2}} - b(B_{\frac{n}{2}})] \\
 &= g\left(\frac{n}{2}\right) + \mathbb{E} g(B_{\frac{n}{2}}),
 \end{aligned}$$

and for odd $n \geq 1$,

$$\begin{aligned}
 g(n) = n - b(n) &= n - \left[\frac{n-1}{4} + b\left(\frac{n-1}{2}\right) + \mathbb{E} b(B_{(n-1)/2}) \right] \\
 &= 1 + \left[\frac{n-1}{2} - b\left(\frac{n-1}{2}\right) \right] + \mathbb{E} [B_{(n-1)/2} - b(B_{(n-1)/2})] \\
 &= 1 + g\left(\frac{n-1}{2}\right) + \mathbb{E} g(B_{(n-1)/2}).
 \end{aligned}$$

So, for $n \geq 0$,

$$g(n) = g\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \mathbb{E} g(B_{\lfloor \frac{n}{2} \rfloor}) + \mathbf{1}_{\{n \text{ is odd}\}}, \quad (2.13)$$



where $\mathbf{1}$ denotes the indicator function.

Lemma 2.1. For $\delta \in (0, 1)$, we have

$$\lim_{n \rightarrow \infty} \frac{1}{n} \log \mathbb{P} \left(B_n > \frac{n}{2}(1 + \delta) \right) = -\frac{1}{2} \left[(1 - \delta) \log(1 - \delta) + (1 + \delta) \log(1 + \delta) \right] < 0.$$

Lemma 2.2. If $f(0) \leq f(1) \leq \dots \leq f(n + 1)$, then $\mathbb{E} f(B_{n+1}) \geq \mathbb{E} f(B_n)$.

Lemma 2.1 is a standard result in large deviation theory; see e.g. [9, pages 539–540].

Lemma 2.2 follows from the fact that B_n is stochastically smaller than B_{n+1} . By (2.13),

$$g(0) = 0, \quad g(1) = 1, \quad g(2) = \frac{3}{2}, \quad g(3) = \frac{5}{2}, \quad g(4) = \frac{19}{8} < g(3), \quad (2.14)$$

so that $g(n)$ is not non-decreasing. Lemma 2.3 below constructs two non-decreasing sequences G and H that are closely related to g and satisfy $0 \leq H(n) \leq g(n) \leq G(n) \leq n$.

Lemma 2.3. Let $G(n)$ and $H(n)$, $n = 0, 1, \dots$ be defined by

$$G(n) = g(n) \quad \text{for } n = 0, 1, 2, 3,$$

$$H(n) = g(n) \quad \text{for } n = 0, 1,$$

and recursively

$$G(n) = G\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \mathbb{E} G(B_{\lfloor \frac{n}{2} \rfloor}) + 1 \quad \text{for } n \geq 4, \quad (2.15)$$

$$H(n) = H\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \mathbb{E} H(B_{\lfloor \frac{n}{2} \rfloor}) \quad \text{for } n \geq 2. \quad (2.16)$$

Then (i) G is non-decreasing and $g(n) \leq G(n) \leq n$ for all n , and (ii) H is non-decreasing and $g(n) \geq H(n) \geq 0$ for all n .

Proof. In view of (2.13) and (2.15), it is easily shown by induction that $g(n) \leq G(n)$ for



all n . By (2.14), $G(n) = g(n) \leq n$ and $G(n)$ is non-decreasing for $n \leq 3$. For $n \geq 4$, if $G(\ell) \leq \ell$ for all $\ell < n$, then

$$\begin{aligned} G(n) &= G\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \mathbb{E} G(B_{\lfloor \frac{n}{2} \rfloor}) + 1 \\ &\leq \left\lfloor \frac{n}{2} \right\rfloor + \mathbb{E} B_{\lfloor \frac{n}{2} \rfloor} + 1 \\ &\leq \frac{3}{4}n + 1 \leq n. \end{aligned}$$

It follows by induction that $G(n) \leq n$ for all n . That $G(n)$ is non-decreasing in n also follows by induction and Lemma 2.2. This proves part (i). In view of (2.13) and (2.16), part (ii) can be proved similarly. \square

Lemma 2.4. For each $\delta \in (0, 1)$, there exists an $N \geq 4$ and a non-decreasing sequence $(G'(0), G'(1), \dots)$ such that $G'(n) \geq g(n)$ for all n and

$$G'(n) = G'\left(\left\lfloor \frac{n}{c} \right\rfloor\right) + \frac{4}{c^2} G'\left(\left\lfloor \frac{n}{c^2} \right\rfloor\right), \quad \text{for all } n \geq N,$$

where $c = c(\delta) = 2/\sqrt{1+\delta}$. Moreover,

$$\limsup_{n \rightarrow \infty} \frac{\log g(n)}{\log n} \leq \limsup_{n \rightarrow \infty} \frac{\log G'(n)}{\log n} \leq \frac{1}{\log c} \log \left(\frac{1}{2} + \sqrt{\frac{4}{c^2} + \frac{1}{4}} \right).$$

Consequently, letting $\delta \rightarrow 0$ so that $c = c(\delta) \rightarrow 2$, we have

$$\limsup_{n \rightarrow \infty} \frac{\log g(n)}{\log n} \leq \log \left(\frac{1 + \sqrt{5}}{2} \right).$$

Proof. Let $\delta \in (0, 1)$ be fixed. Let G' be defined as in Lemma 2.3, so that G' is non-

decreasing and $0 \leq g(n) \leq G(n) \leq n$ for all n . We have

$$\begin{aligned} \mathbb{E} G(B_{\lfloor \frac{n}{2} \rfloor}) + 1 &\leq 1 + G\left(\left\lfloor \left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1+\delta}{2}\right) \right\rfloor\right) \mathbb{P}\left(B_{\lfloor \frac{n}{2} \rfloor} \leq \left\lfloor \left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1+\delta}{2}\right) \right\rfloor\right) \\ &\quad + G\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \mathbb{P}\left(B_{\lfloor \frac{n}{2} \rfloor} > \left\lfloor \left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1+\delta}{2}\right) \right\rfloor\right) \\ &\leq 1 + G\left(\left\lfloor \left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1+\delta}{2}\right) \right\rfloor\right) + \left\lfloor \frac{n}{2} \right\rfloor \mathbb{P}\left(B_{\lfloor \frac{n}{2} \rfloor} > \left\lfloor \left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1+\delta}{2}\right) \right\rfloor\right). \end{aligned} \quad (2.17)$$

By (2.1), $G(\lfloor \frac{\lfloor n/2 \rfloor}{2} (1+\delta) \rfloor) \geq g(\lfloor \frac{\lfloor n/2 \rfloor}{2} (1+\delta) \rfloor) \rightarrow \infty$ as $n \rightarrow \infty$, and by Lemma 2.1,

$$\lim_{n \rightarrow \infty} \left\lfloor \frac{n}{2} \right\rfloor \mathbb{P}\left(B_{\lfloor \frac{n}{2} \rfloor} > \left\lfloor \left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1+\delta}{2}\right) \right\rfloor\right) = 0,$$

so that by (2.17), there is a (large) $N \geq 4$ such that

$$\mathbb{E} G(B_{\lfloor \frac{n}{2} \rfloor}) + 1 \leq (1+\delta) G\left(\left\lfloor \left\lfloor \frac{n}{2} \right\rfloor \left(\frac{1+\delta}{2}\right) \right\rfloor\right) \quad \text{for all } n \geq N. \quad (2.18)$$

Letting $c = 2/\sqrt{1+\delta}$, we have by (2.15) and (2.18) that for all $n \geq N(\geq 4)$,

$$\begin{aligned} G(n) &= G\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \mathbb{E} G(B_{\lfloor \frac{n}{2} \rfloor}) + 1 \\ &\leq G\left(\left\lfloor \frac{n}{c} \right\rfloor\right) + \frac{4}{c^2} G\left(\left\lfloor \frac{n}{c^2} \right\rfloor\right). \end{aligned} \quad (2.19)$$

Define $G'(n)$, $n = 0, 1, \dots$ by $G'(n) = G(n)$ for $n < N$ and recursively

$$G'(n) = G'\left(\left\lfloor \frac{n}{c} \right\rfloor\right) + \frac{4}{c^2} G'\left(\left\lfloor \frac{n}{c^2} \right\rfloor\right) \quad \text{for } n \geq N. \quad (2.20)$$

(Note that for $c > \sqrt{2}$ and $n \geq N \geq 4$, $\lfloor n/c^2 \rfloor \leq \lfloor n/c \rfloor \leq n-1$, so G' is well defined.)

Since $G(n)$ is non-decreasing and $G(n) = G'(n)$ for all $n < N$, we have by (2.19), (2.20)

and induction that $G'(n) \geq G(n) (\geq g(n))$ for all n . To show that $G'(n)$ is non-decreasing,

note that $G'(N) \geq G(N) \geq G(N-1) = G'(N-1)$. Since $G'(0) \leq G'(1) \leq \dots \leq$

$G'(N)$, it follows by (2.20) and induction that $G'(n) \leq G'(n+1)$ for all $n \geq N$.



It remains to prove that

$$\limsup_{n \rightarrow \infty} \frac{\log G'(n)}{\log n} \leq \frac{1}{\log c} \log \left(\frac{1}{2} + \sqrt{\frac{4}{c^2} + \frac{1}{4}} \right). \quad (2.21)$$



Let $\ell_k = \lfloor c^k N \rfloor$, $k = 0, 1, \dots$. Let $x_0 = G'(\ell_0)$, $x_1 = G'(\ell_1)$, and

$$x_k = x_{k-1} + \frac{4}{c^2} x_{k-2}, \quad k = 2, 3, \dots \quad (2.22)$$

By (2.20) and monotonicity of G' , we have for $k \geq 2$

$$\begin{aligned} G'(\ell_k) &= G'(\lfloor c^k N \rfloor) = G' \left(\left\lfloor \frac{\lfloor c^k N \rfloor}{c} \right\rfloor \right) + \frac{4}{c^2} G' \left(\left\lfloor \frac{\lfloor c^k N \rfloor}{c^2} \right\rfloor \right) \\ &\leq G'(\lfloor c^{k-1} N \rfloor) + \frac{4}{c^2} G'(\lfloor c^{k-2} N \rfloor) \\ &= G'(\ell_{k-1}) + \frac{4}{c^2} G'(\ell_{k-2}). \end{aligned} \quad (2.23)$$

Since $x_k = G'(\ell_k)$ for $k = 0, 1$, it follows by (2.22), (2.23) and induction that

$$G'(\ell_k) \leq x_k \quad \text{for all } k \geq 0. \quad (2.24)$$

Since x_k satisfies the difference equation (2.22), we have

$$x_k = \alpha_1 \lambda_1^k + \alpha_2 \lambda_2^k, \quad k = 0, 1, \dots$$

where

$$\begin{aligned} \lambda_1 &= \frac{1}{2}(1 + \gamma), & \lambda_2 &= \frac{1}{2}(1 - \gamma) \\ \alpha_1 &= \frac{1}{\gamma} \left(\frac{1}{2}(\gamma - 1)x_0 + x_1 \right), & \alpha_2 &= \frac{1}{\gamma} \left(\frac{1}{2}(\gamma + 1)x_0 - x_1 \right) \end{aligned}$$

and $\gamma = \sqrt{1 + \frac{16}{c^2}}$. Noting that $-1 < \lambda_2 < 0 < 1 < \lambda_1$ (since $\sqrt{2} < c < 2$) and $\alpha_1 > 0$,

it follows that

$$\lim_{k \rightarrow \infty} \frac{\log x_k}{k} = \log \lambda_1 = \log \left(\frac{1}{2} + \sqrt{\frac{4}{c^2} + \frac{1}{4}} \right). \quad (2.25)$$



By (2.24) and (2.25),

$$\limsup_{k \rightarrow \infty} \frac{\log G'(\ell_k)}{\log \ell_k} \leq \limsup_{k \rightarrow \infty} \frac{\log x_k}{\log [c^k N]} = \frac{\log \lambda_1}{\log c}.$$

Since G' is non-decreasing, for each $n \geq 1$, let $k = k(n)$ be such that $\ell_k \leq n < \ell_{k+1}$, so

that

$$\frac{\log G'(n)}{\log n} \leq \frac{\log G'(\ell_{k+1})}{\log \ell_k},$$

implying that

$$\begin{aligned} \limsup_{n \rightarrow \infty} \frac{\log G'(n)}{\log n} &\leq \limsup_{k \rightarrow \infty} \frac{\log G'(\ell_{k+1})}{\log \ell_k} \\ &= \limsup_{k \rightarrow \infty} \frac{\log G'(\ell_{k+1}) \log \ell_{k+1}}{\log \ell_{k+1} \log \ell_k} \\ &\leq \frac{\log \lambda_1}{\log c} = \frac{1}{\log c} \log \left(\frac{1}{2} + \sqrt{\frac{4}{c^2} + \frac{1}{4}} \right), \end{aligned}$$

proving (2.21). The proof is complete. □

Lemma 2.5. For each $\delta \in (0, 1)$, there exists an $N \geq 4$ and a non-decreasing sequence

$(H'(0), H'(1), \dots)$ such that $0 \leq H'(n) \leq g(n)$ for all n and

$$H'(n) = H' \left(\left\lceil \frac{n}{d} \right\rceil \right) + \frac{4}{d^2} H' \left(\left\lceil \frac{n}{d^2} \right\rceil \right), \quad \text{for all } n \geq N,$$

where $d = d(\delta) = 2 + \delta$ and $\lceil x \rceil$ denotes the smallest integer not less than x . Moreover,

$$\liminf_{n \rightarrow \infty} \frac{\log g(n)}{\log n} \geq \liminf_{n \rightarrow \infty} \frac{\log H'(n)}{\log n} \geq \frac{1}{\log d} \log \left(\frac{1}{2} + \sqrt{\frac{4}{d^2} + \frac{1}{4}} \right).$$

Consequently, letting $\delta \rightarrow 0$ so that $d = d(\delta) \rightarrow 2$, we have

$$\liminf_{n \rightarrow \infty} \frac{\log g(n)}{\log n} \geq \log \left(\frac{1 + \sqrt{5}}{2} \right).$$



Proof. The following proof is similar to that of Lemma 2.4. Let $\delta \in (0, 1)$ be fixed. Let H be defined as in Lemma 2.3, so that H is non-decreasing and $0 \leq H(n) \leq g(n)$ for all n . Also $H(0) = g(0) = 0$, $H(1) = g(1) = 1$.

For $d = 2 + \delta > 2$, by the law of large numbers, $\mathbb{P}(B_{\lfloor \frac{n}{2} \rfloor} < \lceil \frac{n}{d^2} \rceil) \rightarrow 0$ as $n \rightarrow \infty$.

So there exists an $N \geq 4$ such that for all $n \geq N$,

$$H\left(\left\lfloor \frac{n}{2} \right\rfloor\right) \geq H\left(\left\lceil \frac{n}{d} \right\rceil\right) \quad \text{and} \quad \mathbb{P}\left(B_{\lfloor \frac{n}{2} \rfloor} \geq \left\lceil \frac{n}{d^2} \right\rceil\right) \geq \frac{4}{d^2}.$$

By (2.16), for $n \geq N \geq 4$,

$$\begin{aligned} H(n) &= H\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + \mathbb{E} H(B_{\lfloor \frac{n}{2} \rfloor}) \\ &\geq H\left(\left\lceil \frac{n}{d} \right\rceil\right) + \mathbb{P}\left(B_{\lfloor \frac{n}{2} \rfloor} \geq \left\lceil \frac{n}{d^2} \right\rceil\right) H\left(\left\lceil \frac{n}{d^2} \right\rceil\right) \\ &\geq H\left(\left\lceil \frac{n}{d} \right\rceil\right) + \frac{4}{d^2} H\left(\left\lceil \frac{n}{d^2} \right\rceil\right). \end{aligned} \tag{2.26}$$

Define $H'(n)$, $n = 0, 1, \dots$ by $H'(0) = 0$, $H'(1) = \dots = H'(N-1) = 1$ and recursively

$$H'(n) = H'\left(\left\lceil \frac{n}{d} \right\rceil\right) + \frac{4}{d^2} H'\left(\left\lceil \frac{n}{d^2} \right\rceil\right) \quad \text{for } n \geq N. \tag{2.27}$$

(Note that $\lceil \frac{n}{d^2} \rceil \leq \lceil \frac{n}{d} \rceil \leq n - 1$ for $n \geq N \geq 4$, so that the recursion is well defined.)

Since by (2.27), $H'(N) = H'(\lceil \frac{N}{d} \rceil) + \frac{4}{d^2} H'(\lceil \frac{N}{d^2} \rceil) = 1 + \frac{4}{d^2} > 1$, we have $H'(0) < H'(1) = \dots = H'(N-1) < H'(N)$. It follows by (2.27) and induction that H' is a non-decreasing sequence. Since $H(n) \geq H'(n)$ for all $n < N$, we have by (2.26), (2.27) and induction that $H'(n) \leq H(n)$ for all n .

It remains to prove that

$$\liminf_{n \rightarrow \infty} \frac{\log H'(n)}{\log n} \geq \frac{1}{\log d} \log \left(\frac{1}{2} + \sqrt{\frac{4}{d^2} + \frac{1}{4}} \right) \quad (2.28)$$



Let $\ell_k = \lceil d^k N \rceil$, $k = 0, 1, \dots$. Let $x_0 = H'(\ell_0)$, $x_1 = H'(\ell_1)$, and

$$x_k = x_{k-1} + \frac{4}{d^2} x_{k-2}, \quad k = 2, 3, \dots \quad (2.29)$$

By (2.27) and monotonicity of H' , we have for $k \geq 2$

$$\begin{aligned} H'(\ell_k) &= H'(\lceil d^k N \rceil) = H' \left(\left\lceil \frac{\lceil d^k N \rceil}{d} \right\rceil \right) + \frac{4}{d^2} H' \left(\left\lceil \frac{\lceil d^k N \rceil}{d^2} \right\rceil \right) \\ &\geq H'(\lceil d^{k-1} N \rceil) + \frac{4}{d^2} H'(\lceil d^{k-2} N \rceil) \\ &= H'(\ell_{k-1}) + \frac{4}{d^2} H'(\ell_{k-2}). \end{aligned} \quad (2.30)$$

Since $x_k = H'(\ell_k)$ for $k = 0, 1$, it follows by (2.29), (2.30) and induction that

$$H'(\ell_k) \geq x_k \quad \text{for all } k \geq 0. \quad (2.31)$$

Note that the difference equation (2.29) is the same as (2.22) with c replaced by d . Solving

(2.29) yields (cf. (2.25))

$$\lim_{k \rightarrow \infty} \frac{\log x_k}{k} = \log \left(\frac{1}{2} + \sqrt{\frac{4}{d^2} + \frac{1}{4}} \right).$$

By (2.31),

$$\liminf_{k \rightarrow \infty} \frac{\log H'(\ell_k)}{\log \ell_k} \geq \liminf_{k \rightarrow \infty} \frac{\log x_k}{\log \lceil d^k N \rceil} = \frac{1}{\log d} \log \left(\frac{1}{2} + \sqrt{\frac{4}{d^2} + \frac{1}{4}} \right).$$

Since H' is non-decreasing, for each $n \geq 1$, let $k = k(n)$ be such that $\ell_k \leq n < \ell_{k+1}$, so



that

$$\frac{\log H'(n)}{\log n} \geq \frac{\log H'(\ell_k)}{\log \ell_{k+1}},$$

implying that

$$\begin{aligned} \liminf_{n \rightarrow \infty} \frac{\log H'(n)}{\log n} &\geq \liminf_{k \rightarrow \infty} \frac{\log H'(\ell_k)}{\log \ell_{k+1}} \\ &= \liminf_{k \rightarrow \infty} \frac{\log H'(\ell_k)}{\log \ell_k} \frac{\log \ell_k}{\log \ell_{k+1}} \\ &\geq \frac{1}{\log d} \log \left(\frac{1}{2} + \sqrt{\frac{4}{d^2} + \frac{1}{4}} \right), \end{aligned}$$

proving (2.28). The proof is complete. □

2.3 Numerical results and discussion

Recall that $g(n) = n - b(n) = n - \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|$. By (2.13), we computed $g(n)$ for all $n \leq 65536$. Figure 2.1 plots $\log g(n)/\log n$ versus n for $n \leq 65536$ where $\theta = \log[(1 + \sqrt{5})/2] \approx 0.694$. It shows that $\log g(n)/\log n$ is slightly greater than θ and appears to converge to θ slowly. Figure 2.2 plots $g(n)/n^\theta$ versus n for $n \leq 65536$. By Proposition 2.2(ii),

$$\lim_{n \rightarrow \infty} \log [g(n)/n^\theta] / \log n = 0.$$

While it is unclear whether $g(n)/n^\theta$ converges to some constant eventually, it appears that $g(n)/n^\theta$ fluctuates less when n becomes larger. Figure 2.3 plots $g(2n)/g(n)$ versus n for $n \leq 32768$. It appears that $g(2n)/g(n)$ is close to 2^θ for large n . Figure 2.4 plots $g(3n)/g(n)$ versus n for $n \leq 21845$, where $g(3n)/g(n)$ oscillates around 3^θ . Our limited numerical results provide weak evidence that $g(3n)/g(n)$ converges to 3^θ eventually.

Figure 2.5 plots $\log \text{Var}_{1/2} |\mathcal{A}_P(\mathbf{X}_{2^k})| / \log 2^k$ for $k = 1, \dots, 25$, where \mathbf{X}_n is a se-

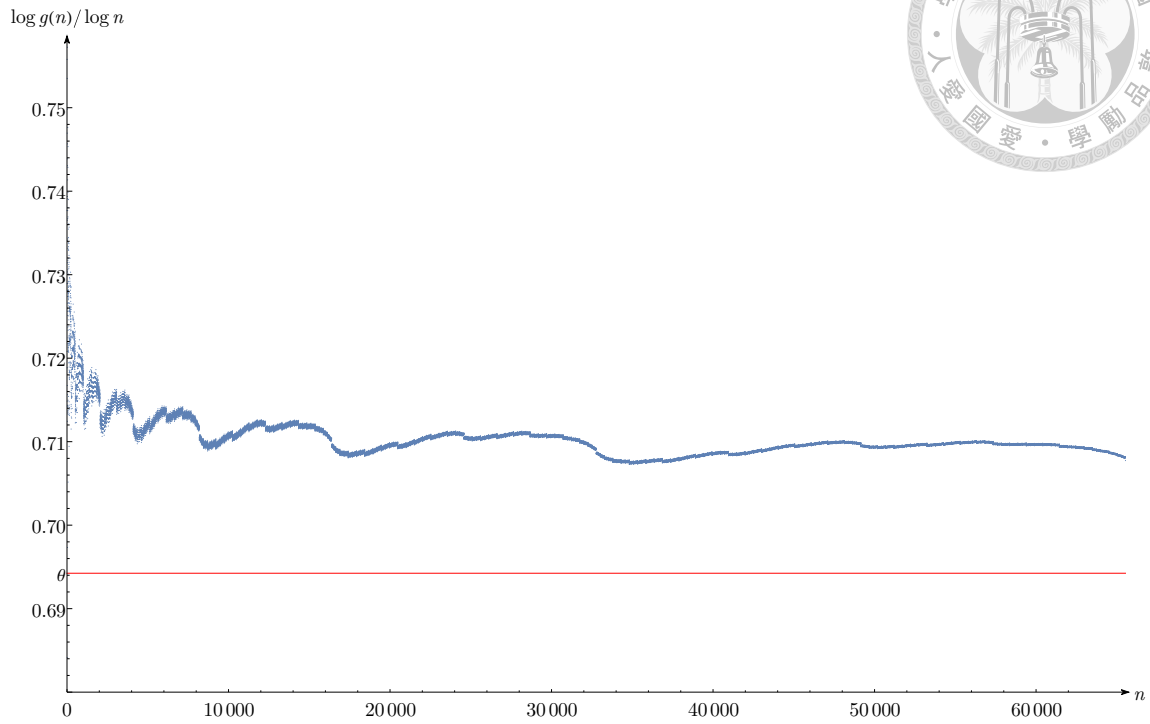


Figure 2.1: Plot of $\log g(n) / \log n$ versus n .

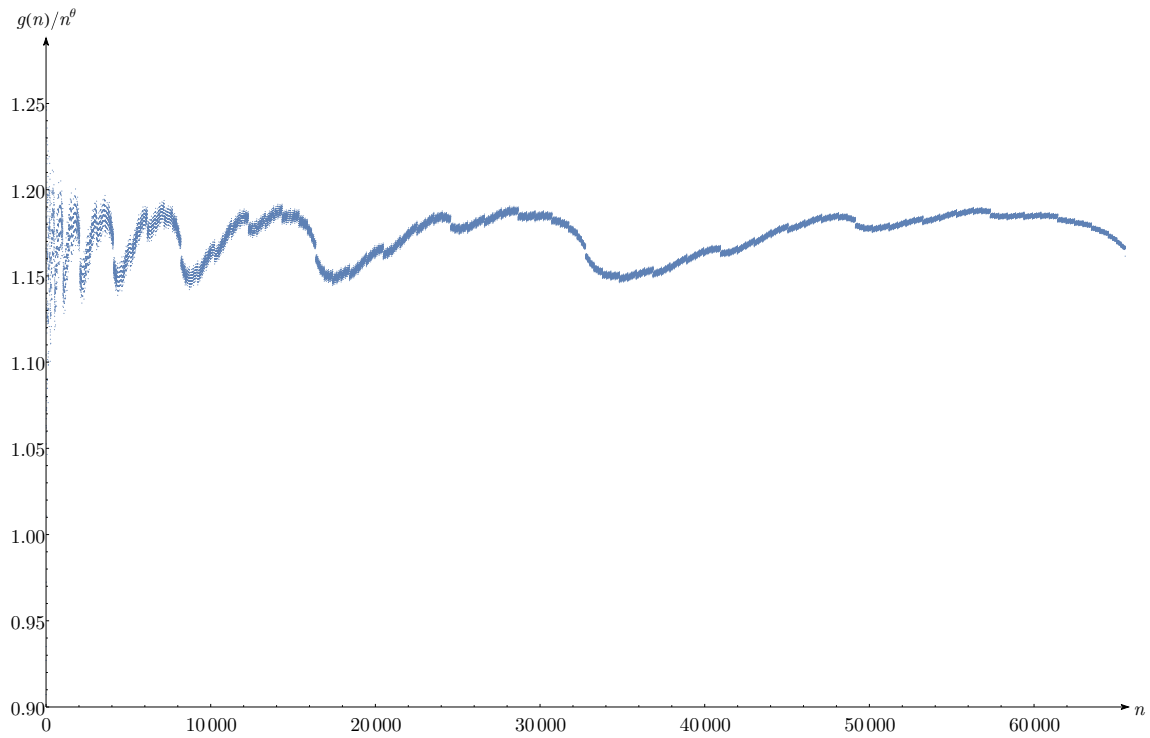


Figure 2.2: Plot of $g(n) / n^\theta$ versus n with $\theta = \log[(1 + \sqrt{5})/2]$.

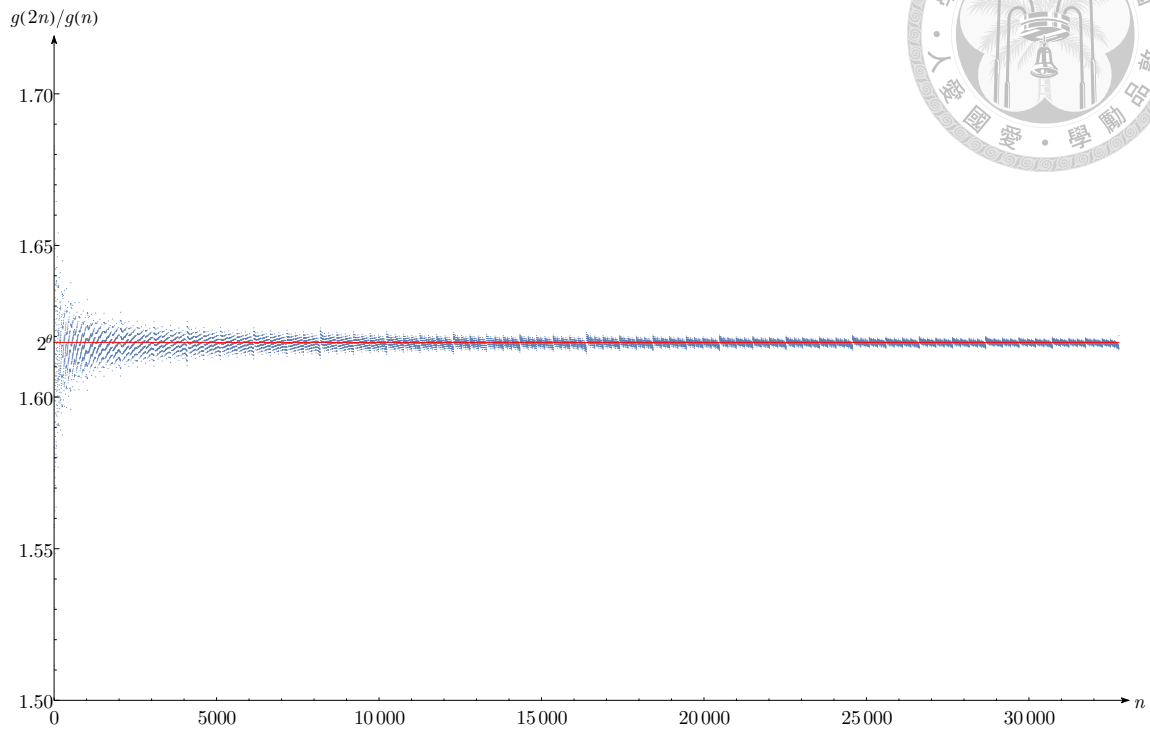


Figure 2.3: Plot of $g(2n)/g(n)$ versus n .

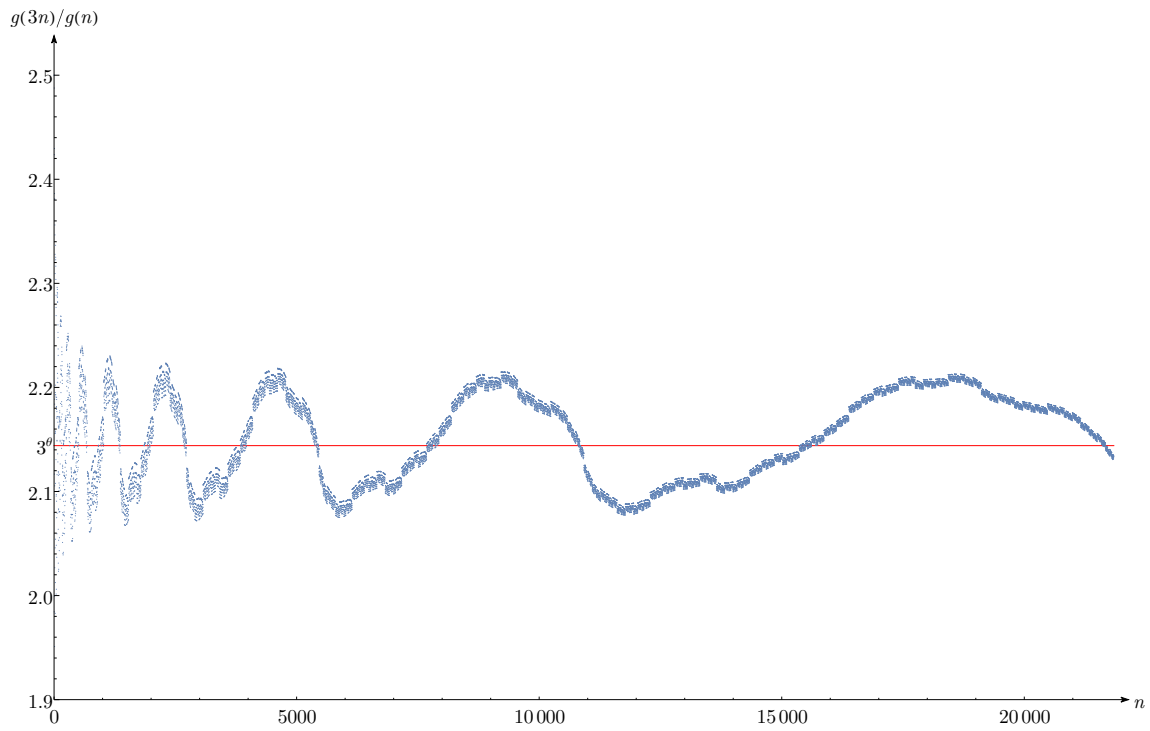


Figure 2.4: Plot of $g(3n)/g(n)$ versus n .

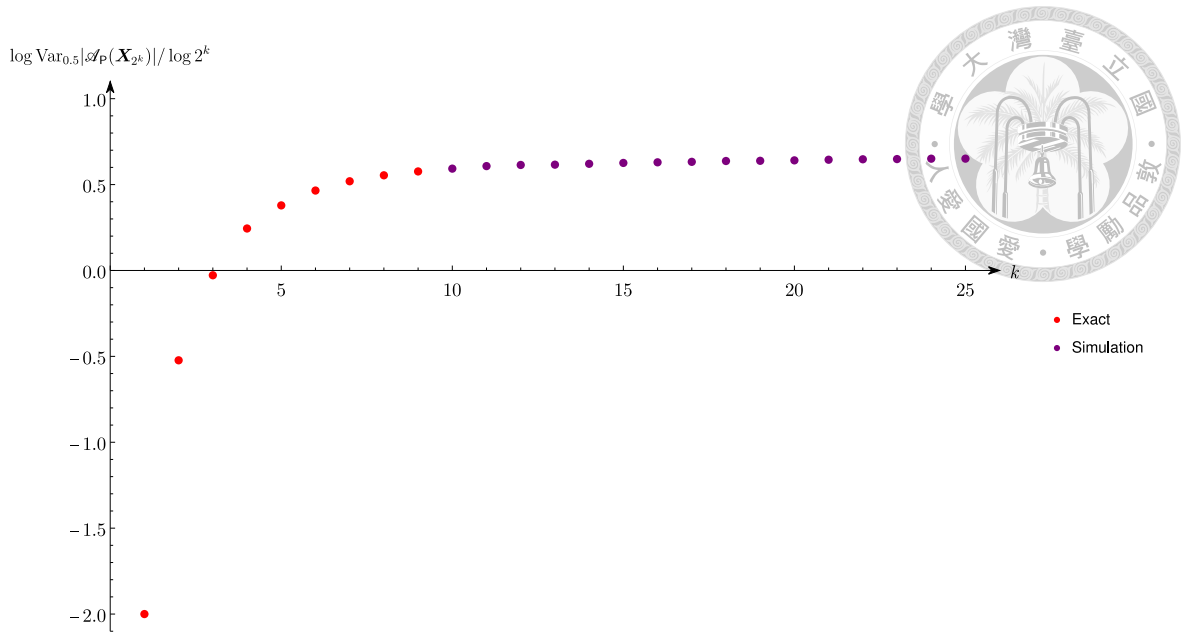


Figure 2.5: Plot for $\text{Var}_{0.5} |\mathcal{A}_P(\mathbf{X}_{2^k})|$.

quence of n unbiased bits. The first 9 points in the plot, with $k = 1, \dots, 9$, are exact values calculated from the distribution of $|\mathcal{A}_P(\mathbf{X}_{2^k})|$ with $p = 1/2$, while the latter 16 points in the plot are obtained from simulation with 10000 replications for each k . The variance $\text{Var}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|$ appears to increase in n . There is very weak evidence that $\log \text{Var}_{1/2}(\mathbf{X}_n) / \log n$ converges to a positive constant. While we have discussed mainly the asymptotic behavior of $\mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|$ for $p = 1/2$, it is also of interest to see how fast $\mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)| / n$ approaches $h(p)$ for $p \neq 1/2$. Figure 2.6 plots $\log(nh(p) - \mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|) / \log n$ for $n = 2^k$, $k = 1, \dots, 26$, $p = 0.3$, suggesting that it might also converge to θ as $n \rightarrow \infty$. In this plot, the first 9 points are exact values, while the latter 17 points are obtained from simulation with 10000 replications for each of $k \in \{10, \dots, 24\}$, 5000 replications for $k = 25$, and 2500 replications for $k = 26$.

While the distribution of $|\mathcal{A}_P(\mathbf{X}_n)|$ is complicated, it seems natural to ask whether $(|\mathcal{A}_P(\mathbf{X}_n)| - \mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|) / \sqrt{\text{Var}_p |\mathcal{A}_P(\mathbf{X}_n)|}$ is approximately standard normal for large n . For $p = 1/2$, we conducted a simulation study with 10000 samples for each of $n = 2^{10}$, 2^{17} , 2^{25} . Let $Q_{n,i}$, $i = 1, \dots, 10000$, denote the 10000 observations of $|\mathcal{A}_P(\mathbf{X}_n)|$. For

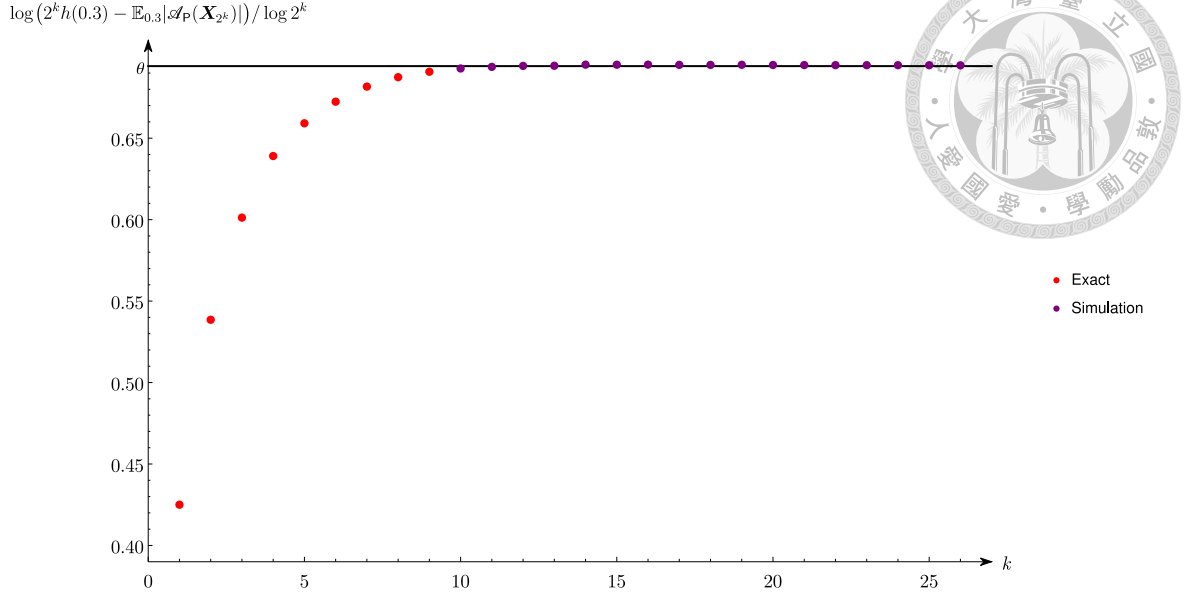


Figure 2.6: Plot for $2^k h(0.3) - \mathbb{E}_{0.3} |\mathcal{A}_P(\mathbf{X}_{2^k})|$.

$n = 2^{10}, 2^{17}, 2^{25}$, Figure 2.7(a)–(c) plots the histograms of the standardized observations $\{(Q_{n,i} - \bar{Q}_n) / \text{s.d.}(Q_n) : i = 1, \dots, 10000\}$ where \bar{Q}_n and $\text{s.d.}(Q_n)$ denote the sample mean and sample standard deviation of $\{Q_{n,i} : i = 1, \dots, 10000\}$. Also included in the plots is the standard normal density function. It appears that the standard normal distribution approximates reasonably well the standardized distribution of $|\mathcal{A}_P(\mathbf{X}_n)|$ for $n = 2^{17}$ and 2^{25} .

Although we have no proof of the asymptotic normality result, the following argument suggests that the asymptotic normality may hold. Choose a sequence ℓ_n such that $\lim_{n \rightarrow \infty} \ell_n = \infty$ and $\lim_{n \rightarrow \infty} \ell_n / n = 0$. Divide the sample $\mathbf{X}_n = (X_1, \dots, X_n)$ into blocks of size ℓ_n ,

$$\mathbf{Y}^{(i)} = (X_{(i-1)\ell_n+1}, \dots, X_{i\ell_n}), \quad i = 1, \dots, \lfloor n/\ell_n \rfloor.$$

Apply \mathcal{A}_p to each $\mathbf{Y}^{(i)}$, yielding output bits $\mathcal{A}_P(\mathbf{Y}^{(i)})$. Note that $|\mathcal{A}_P(\mathbf{Y}^{(i)})|, i = 1, \dots, \lfloor n/\ell_n \rfloor$, are iid, and that $|\mathcal{A}_P(\mathbf{X}_n)| \geq \sum_{i=1}^{\lfloor n/\ell_n \rfloor} |\mathcal{A}_P(\mathbf{Y}^{(i)})|$. If ℓ_n increases to infinity sufficiently fast, the difference of $|\mathcal{A}_P(\mathbf{X}_n)|$ and $\sum_{i=1}^{\lfloor n/\ell_n \rfloor} |\mathcal{A}_P(\mathbf{Y}^{(i)})|$ may be negligible compared to

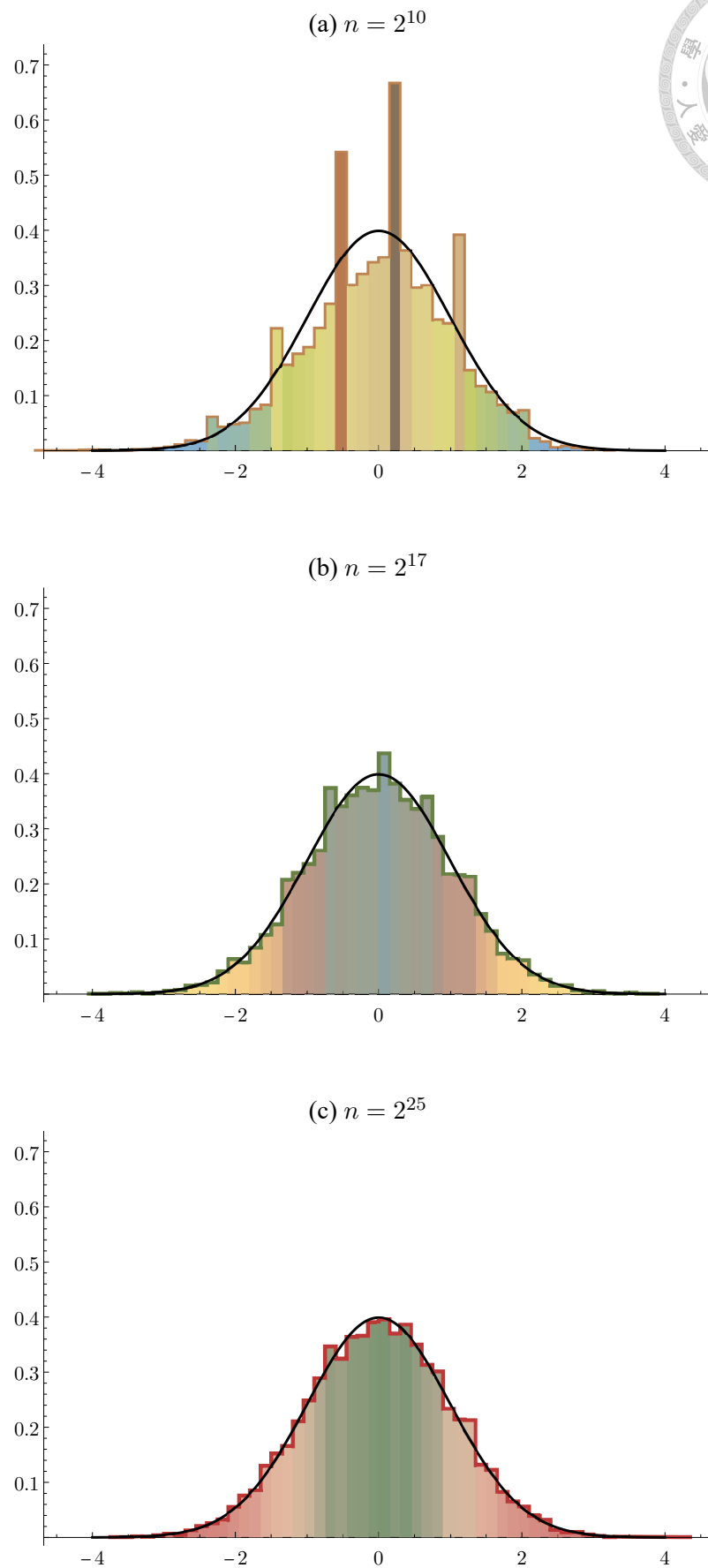
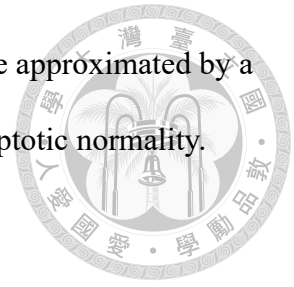


Figure 2.7: Simulated histograms for the standardized distribution of $|\mathcal{A}_P(\mathbf{X}_n)|$ for $p = 0.5$ and $n = 2^{10}, 2^{17}, 2^{25}$.

the standard deviation of $|\mathcal{A}_P(\mathbf{X}_n)|$. In other words, $|\mathcal{A}_P(\mathbf{X}_n)|$ can be approximated by a sum of iid random variables, which suggests the validity of the asymptotic normality.

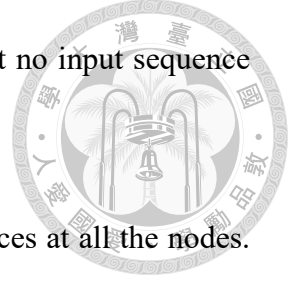




Chapter 3 Streaming versions of Peres' algorithm

3.1 Introduction and streaming algorithms

Let $\mathbf{X} = (X_1, X_2, \dots)$ be the input sequence consisting of the outcomes when a coin of bias p is tossed repeatedly (which is referred to as the source). Let $\mathbf{X}_n = (X_1, \dots, X_n)$. Recall the binary tree representation of \mathcal{A}_P introduced in Section 1.4 where the rooted binary tree \mathbf{T} is identified with the sequence of nodes $(\nu_1, \nu_{10}, \nu_{11}, \nu_{100}, \nu_{101}, \dots)$. At time n , $n = 1, 2, \dots$, the input bit X_n arrives (is received) at the root node ν_1 from the source. For a (fixed) time n , the input sequence \mathbf{X}_n at ν_1 is decomposed into $\lambda(\mathbf{X}_n)$, $\rho(\mathbf{X}_n)$ and $\mathcal{A}_{\text{VN}}(\mathbf{X}_n)$, so that $\lambda(\mathbf{X}_n)$ and $\rho(\mathbf{X}_n)$ become the input sequences at nodes ν_{10} and ν_{11} , respectively, while $\mathcal{A}_{\text{VN}}(\mathbf{X}_n)$ is the output sequence at ν_1 . Furthermore, the input sequence $\lambda(\mathbf{X}_n)$ ($\rho(\mathbf{X}_n)$, resp.) at node ν_{10} (ν_{11} , resp.) is decomposed into $\lambda^2(\mathbf{X}_n)$, $\rho\lambda(\mathbf{X}_n)$, $\mathcal{A}_{\text{VN}}(\lambda(\mathbf{X}_n))$ ($\lambda\rho(\mathbf{X}_n)$, $\rho^2(\mathbf{X}_n)$, $\mathcal{A}_{\text{VN}}(\rho(\mathbf{X}_n))$, resp.). More generally, for a node $\nu_{b_1 \dots b_\eta}$ (of level η), when the input sequence $\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{X}_n)$ is received at node $\nu_{b_1 b_2 \dots b_\eta}$ where $\psi_{b_\ell} = \lambda$ or ρ according as $b_\ell = 0$ or 1 , it is decomposed into three sequences, $\lambda\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{X}_n)$, $\rho\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{X}_n)$, $\mathcal{A}_{\text{VN}}(\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{X}_n))$, so that the first becomes the input sequence at node $\nu_{b_1 \dots b_\eta 0}$, the second becomes the input sequence at node $\nu_{b_1 \dots b_\eta 1}$,



and the third becomes the output sequence at node $\nu_{b_1 \dots b_\eta}$. Note that no input sequence can be received at node $\nu_{b_1 \dots b_\eta}$ of level $\eta \geq \lfloor \log n \rfloor + 2$.

We denote by $\mathcal{A}_P(\mathbf{X}_n)$ the total collection of the output sequences at all the nodes. More precisely, the nodes are ordered lexicographically in the sense that $\nu_{b_1 \dots b_\eta} \prec' \nu_{b'_1 \dots b'_\eta}$ if for some $1 \leq \ell \leq \min\{\eta, \eta'\}$, $b_i = b'_i$ for $i = 1, \dots, \ell$, and either $\ell = \eta < \eta'$ or $b_{\ell+1} < b'_{\ell+1}$. For example, $\nu_{101} \prec' \nu_{11} \prec' \nu_{110} \prec' \nu_{1100} \prec' \nu_{111}$. Then $\mathcal{A}_P(\mathbf{X}_n)$ is the sequence of all output bits that results from arranging the (non-empty) output sequences at all nodes according to the node ordering \prec' .

Given two time points $n < n'$, while $\mathcal{A}_P(\mathbf{X}_n)$ is contained in $\mathcal{A}_P(\mathbf{X}_{n'})$ (due to \mathcal{A}_P being nested), some (output) bits in $\mathcal{A}_P(\mathbf{X}_n)$ generated by time n may be placed after bits in $\mathcal{A}_P(\mathbf{X}_{n'}) \setminus \mathcal{A}_P(\mathbf{X}_n)$. As an example, consider $\mathbf{x} = \text{HTHHHTH} \dots$, we have $\mathcal{A}_P(\mathbf{x}_4) = 11$ and $\mathcal{A}_P(\mathbf{x}_6) = 101$. The second 1 in $\mathcal{A}_P(\mathbf{x}_4)$ is moved to the third place in $\mathcal{A}_P(\mathbf{x}_6)$. To avoid this undesirable property, we introduce the notion of streaming algorithm. For a nested algorithm \mathcal{A} , let $\mathcal{A}(\mathbf{X}_n)$ be the sequence of output bits generated by \mathcal{A} applied to \mathbf{X}_n . The output bits in $\mathcal{A}(\mathbf{X}_n) \setminus \mathcal{A}(\mathbf{X}_{n-1})$ are said to be *induced* by the n th input bit X_n . A (nested) algorithm \mathcal{A} is said to be *streaming* if for every n , the output bits induced by X_n are placed after the bits in $\mathcal{A}_P(\mathbf{X}_{n-1})$. In other words, \mathcal{A} is a streaming algorithm if $\mathcal{A}(\mathbf{X}_n)$ is a prefix of $\mathcal{A}(\mathbf{X}_{n'})$ for $n < n'$. It is easily seen that \mathcal{A}_{ν_N} is a streaming algorithm although it is not efficient. In the previous example with \mathcal{A}_P applied to the sequence $\mathbf{x} = \text{HTHHHTH} \dots$, we have that $x_2 (= T)$ induces an output bit 1 at node ν_1 , and $x_4 (= H)$ induces an output bit 1 at node ν_{10} , and $x_6 (= H)$ induces an output bit 0 at node ν_1 . The output sequence for a streaming version of \mathcal{A}_P applied to $\mathbf{x} = \text{HTHHHTH} \dots$ would be $110 \dots$. Note that for odd n , X_n induces no output bit for any streaming version of \mathcal{A}_P . To define a streaming version of \mathcal{A}_P , when an input bit X_n induces two or more

output bits (necessarily at different nodes), it is required to order the output bits according to a pre-specified rule (e.g. a given ordering of the nodes). For example, consider the lexicographical ordering \prec' . We denote by \mathcal{A}'_S the streaming version of \mathcal{A}_P based on the ordering \prec' . When \mathcal{A}'_S is applied to $\mathbf{x} = \text{TTHTHTHH}$, the output sequence is 10110. In this case, $x_8 (= \text{H})$ induces two output bits 1 and 0 at nodes ν_{10} and ν_{11} , respectively, and 1 is placed ahead of 0 since $\nu_{10} \prec' \nu_{11}$. However, for $n = 8$, given $|\mathcal{A}'_S(\mathbf{X}_8)| = 5$, the conditional distribution of $\mathcal{A}'_S(\mathbf{X}_8)$ is not uniform on $\{0, 1\}^5$. Indeed, we have

$$\begin{aligned}
\mathbb{P}_p(\mathcal{A}'_S(\mathbf{X}_8) = 00000) &= \mathbb{P}_p(\mathbf{X}_8 \in \{\text{THTHTTTH}, \text{TTTHHHTH}, \text{THTHHHTH}\}) \\
&= p^3q^5 + p^4q^4 + p^5q^3, \\
\mathbb{P}_p(\mathcal{A}'_S(\mathbf{X}_8) = 00010) &= \mathbb{P}_p(\mathbf{X}_8 \in \{\text{THTHTHTT}, \text{TTTHHHHT}, \text{TTHTTHHH}, \text{THTHTHHH}\}) \\
&= p^3q^5 + 2p^4q^4 + p^5q^3 \\
&> \mathbb{P}_p(\mathcal{A}'_S(\mathbf{X}_8) = 00000).
\end{aligned}$$

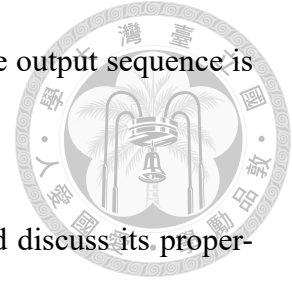
This shows that

$$\mathbb{P}_p(\mathcal{A}'_S(\mathbf{X}_8) = 00000 \mid |\mathcal{A}'_S(\mathbf{X}_8)| = 5) < \mathbb{P}_p(\mathcal{A}'_S(\mathbf{X}_8) = 00010 \mid |\mathcal{A}'_S(\mathbf{X}_8)| = 5).$$

Thus, in general, the output bits generated by \mathcal{A}'_S are not independent unbiased.

Instead of the ordering \prec' , we consider another lexicographical ordering \prec with the roles of 0 and 1 interchanged. Specifically, we write $\nu_{b_1 \dots b_\eta} \prec \nu_{b'_1 \dots b'_\eta}$ if for some $1 \leq \ell \leq \min\{\eta, \eta'\}$, $b_i = b'_i$ for $i = 1, \dots, \ell$, and either $\ell = \eta < \eta'$ or $b_{\ell+1} > b'_{\ell+1}$. For example, $\nu_{11} \prec \nu_{111} \prec \nu_{110} \prec \nu_{1100} \prec \nu_{101}$. We denote by \mathcal{A}_S the streaming version of \mathcal{A}_P based

on the ordering \prec . When \mathcal{A}_S is applied to $x = \text{TTHHTHHTH}\cdots$, the output sequence is $10101\cdots$ as compared to $10110\cdots$ generated by \mathcal{A}'_S .



In the next section, we introduce the notation of status tree and discuss its properties, which plays an important role of establishing unbiasedness of \mathcal{A}_S in Section 3.3. In Section 3.4, we discuss another streaming version of Peres' algorithm which was first introduced by Zhou and Bruck [30]. In Section 3.5, some results on counting status trees are presented, which are useful for computing the distribution of $|\mathcal{A}_P(\mathbf{X}_n)|$.

Remark 3.1. An algorithm \mathcal{A} is said to be unbiased if for all n, p and ℓ , given $|\mathcal{A}(\mathbf{X}_n)| = \ell$, the conditional distribution of $\mathcal{A}(\mathbf{X}_n)$ is uniform on $\{0, 1\}^\ell$. An unbiased algorithm is not necessarily a streaming algorithm. In general, for an unbiased algorithm \mathcal{A} , given $|\mathcal{A}(\mathbf{X}_n)| = \ell$ and $|\mathcal{A}(\mathbf{X}_{n'})| = \ell'$ for some $n < n'$ and $\ell \leq \ell'$, the conditional distribution of $\mathcal{A}(\mathbf{X}_{n'})$ is not uniform on $\{0, 1\}^{\ell'}$. As an example, let \mathcal{A}^* denote either \mathcal{A}_P or a streaming version of \mathcal{A}_P . We have

$$\begin{aligned} & \mathbb{P}_p(\mathcal{A}^*(\mathbf{X}_4) = 00 \mid |\mathcal{A}^*(\mathbf{X}_2)| = 1, |\mathcal{A}^*(\mathbf{X}_4)| = 2) \\ &= \mathbb{P}_p(\mathcal{A}^*(\mathbf{X}_4) = 10 \mid |\mathcal{A}^*(\mathbf{X}_2)| = 1, |\mathcal{A}^*(\mathbf{X}_4)| = 2) \\ &= \frac{1}{2}pq \leq \frac{1}{8} < \frac{1}{4}. \end{aligned}$$

On the other hand, von Neumann's algorithm \mathcal{A}_{VN} enjoys the unbiasedness property in the strongest sense. For any $p, r \geq 1, n_1 < n_2 < \cdots < n_r$ and $\ell_1 \leq \ell_2 \leq \cdots \leq \ell_r$, given $|\mathcal{A}_{\text{VN}}(\mathbf{X}_{n_i})| = \ell_i, i = 1, \dots, r$, the conditional distribution of $\mathcal{A}_{\text{VN}}(\mathbf{X}_{n_r})$ is uniform on $\{0, 1\}^{\ell_r}$.



3.2 Status tree

To establish the unbiasedness property of \mathcal{A}_S , we need to introduce the notion of status tree. Recall that an (input) sequence \mathbf{x}_n from the source can be decomposed into $\lambda(\mathbf{x}_n)$, $\rho(\mathbf{x}_n)$ and $\mathcal{A}_{\text{VN}}(\mathbf{x}_n)$. If n is even, \mathbf{x}_n can be recovered from $\lambda(\mathbf{x}_n)$, $\rho(\mathbf{x}_n)$ and $\mathcal{A}_{\text{VN}}(\mathbf{x}_n)$. If n is odd, the last bit x_n of \mathbf{x}_n is *lost*. It is for this reason that we define

$$\sigma(\mathbf{x}_n) = \begin{cases} \text{O}, & \text{if } n \text{ is even} \\ x_n, & \text{if } n \text{ is odd,} \end{cases}$$

where O stands for “void”. Thus $\sigma(\mathbf{x}_n) \in \{\text{H}, \text{T}, \text{O}\}$. (By convention, $\sigma(\mathbf{x}_n) = \text{O}$ if $n = 0$.) Then \mathbf{x}_n can be recovered from $\lambda(\mathbf{x}_n)$, $\rho(\mathbf{x}_n)$, $\mathcal{A}_{\text{VN}}(\mathbf{x}_n)$ and $\sigma(\mathbf{x}_n)$. Also it is readily seen that

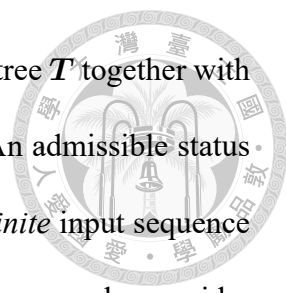
$$|\mathcal{A}_{\text{VN}}(\mathbf{x}_n)| = |\lambda(\mathbf{x}_n)|_{\text{H}}, \quad |\rho(\mathbf{x}_n)| = |\lambda(\mathbf{x}_n)|_{\text{T}}, \quad (3.1)$$

$$|\mathbf{x}_n|_{\text{H}} = |\lambda(\mathbf{x}_n)|_{\text{H}} + 2|\rho(\mathbf{x}_n)|_{\text{H}} + \mathbf{1}(\sigma(\mathbf{x}_n) = \text{H}), \quad (3.2)$$

$$|\mathbf{x}_n|_{\text{T}} = |\lambda(\mathbf{x}_n)|_{\text{H}} + 2|\rho(\mathbf{x}_n)|_{\text{T}} + \mathbf{1}(\sigma(\mathbf{x}_n) = \text{T}), \quad (3.3)$$

where $\mathbf{1}(\cdot)$ denotes the indicator function.

For the (infinite complete) binary tree \mathbf{T} with nodes $\nu_1, \nu_{10}, \nu_{11}, \dots$, given an input sequence \mathbf{x}_n at the root node ν_1 from the source, we can derive an input sequence at each of its descendant nodes. Specifically, the input sequence at node $\nu_{b_1 b_2 \dots b_\eta}$ derived (or induced) from \mathbf{x}_n is $\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{x}_n)$ where $\psi_{b_i} = \lambda$ or ρ according as $b_i = 0$ or 1. For notational simplicity, we write $\mathbf{I}_{\mathbf{x}_n}(\nu_{b_1 \dots b_\eta}) = \psi_{b_\eta} \dots \psi_{b_2}(\mathbf{x}_n)$. In particular, $\mathbf{I}_{\mathbf{x}_n}(\nu_1) = \mathbf{x}_n$. (Here \mathbf{I} stands for “input”.) We call $\sigma(\mathbf{I}_{\mathbf{x}_n}(\nu_{b_1 \dots b_\eta})) = \sigma(\psi_{b_\eta} \dots \psi_{b_2}(\mathbf{x}_n))$ the *status* (or *label*) of



node $\nu_{b_1 \dots b_n}$ at time n (derived from \mathbf{x}_n). A status tree \mathcal{S} is the binary tree T together with the node status (label) $\mathcal{S}(\nu)$ for every node $\nu \in \{\nu_1, \nu_{10}, \nu_{11}, \dots\}$. An admissible status tree is a status tree whose status of each node is derived from some *finite* input sequence at the root node ν_1 . Note that not every status tree is admissible. As an example, consider the status tree \mathcal{S} with $\mathcal{S}(\nu_1) = O$, $\mathcal{S}(\nu_{10}) = H$, $\mathcal{S}(\nu_{11}) = T$, and $\mathcal{S}(\nu) = O$ for all nodes ν of level ≥ 3 . Clearly, this status tree cannot be derived from any input sequence. From now on, we drop the word “admissible”, so that a status tree always refers to an admissible status tree. Note also that for an (admissible) status tree, all but finitely many nodes have status O. A status tree may be derived from more than one input sequence. For example, the status tree \mathcal{S} with $\mathcal{S}(\nu_{10}) = H$ and $\mathcal{S}(\nu) = O$ for all nodes $\nu \neq \nu_{10}$ is derived from the two input sequences HT and TH.

For notational convenience, we denote by $\lambda\nu$ and $\rho\nu$ the left and right child nodes of ν , respectively. Thus $\rho^2\lambda^3\rho\nu_{10} = \nu_{10100011}$. By convention, $\lambda^0\nu = \rho^0\nu = \nu$. For a status tree \mathcal{S} , define the depth $\delta_{\mathcal{S}}$ of \mathcal{S} by

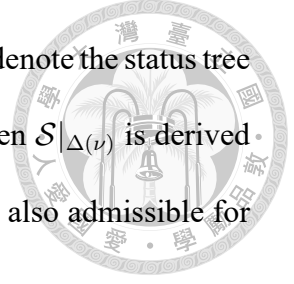
$$\delta_{\mathcal{S}} = \begin{cases} 0, & \text{if } \mathcal{S}(\nu) = O \text{ for all nodes } \nu, \\ 1 + \max\{m \geq 0 : \mathcal{S}(\lambda^m\nu_1) \neq O\}, & \text{otherwise.} \end{cases}$$

Note that if \mathcal{S} is derived from a sequence \mathbf{x} of length $n \geq 1$, then we have $\delta_{\mathcal{S}} = \lfloor \log n \rfloor + 1$, and $\mathcal{S}(\lambda^{\delta_{\mathcal{S}}-1}\nu_1) \neq O$, and $\mathcal{S}(\nu) = O$ for all nodes ν of level $> \delta_{\mathcal{S}}$. Define $\{H, T\}^* = \bigcup_{n=0}^{\infty} \{H, T\}^n$, where $\{H, T\}^0 = \emptyset$. Given a status tree \mathcal{S} , define

$$\mathcal{X}_{\mathcal{S}} = \{\mathbf{x} \in \{H, T\}^* : \mathcal{S}_{\mathbf{x}} = \mathcal{S}\},$$

where $\mathcal{S}_{\mathbf{x}}$ denotes the status tree derived from \mathbf{x} . For a node ν , let $\Delta(\nu)$ denote the subtree consisting of ν and all its descendant nodes. Note that $\Delta(\nu)$ is an infinite complete binary

tree rooted at ν (and is isomorphic to \mathbf{T}). For a status tree \mathcal{S} , let $\mathcal{S}|_{\Delta(\nu)}$ denote the status tree \mathcal{S} restricted to the subtree $\Delta(\nu)$. Note that if \mathcal{S} is derived from \mathbf{x} , then $\mathcal{S}|_{\Delta(\nu)}$ is derived from $\mathbf{I}_x(\nu)$. Consequently, if \mathcal{S} is an admissible status tree, $\mathcal{S}|_{\Delta(\nu)}$ is also admissible for all nodes ν .



Proposition 3.1. *Let \mathcal{S} be a status tree. For $\mathbf{x} \in \mathcal{X}_S$, the quantities $|\mathcal{A}_P(\mathbf{x})|$, $|\mathbf{I}_x(\nu)|$, $|\mathbf{I}_x(\nu)|_H$, $|\mathbf{I}_x(\nu)|_T$ and $|\mathcal{A}_{vN}(\mathbf{I}_x(\nu))|$ for all nodes ν are all \mathcal{S} -properties (i.e. independent of the individual input sequence \mathbf{x}) where $\mathbf{I}_x(\nu)$ denotes the input sequence at node ν derived from \mathbf{x} . In particular, for $\mathbf{x} \in \mathcal{X}_S$, $|\mathbf{x}|$, $|\mathbf{x}|_H$ and $|\mathbf{x}|_T$ are \mathcal{S} -properties (since $\mathbf{x} = \mathbf{I}_x(\nu_1)$).*

Proof. Note that $|\mathbf{I}_x(\nu)| = |\mathbf{I}_x(\nu)|_H + |\mathbf{I}_x(\nu)|_T$ and $|\mathcal{A}_P(\mathbf{x})| = \sum_{\nu} |\mathcal{A}_{vN}(\mathbf{I}_x(\nu))|$. So it suffices to show that for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}_S$,

$$|\mathbf{I}_x(\nu)|_H = |\mathbf{I}_{x'}(\nu)|_H, \quad |\mathbf{I}_x(\nu)|_T = |\mathbf{I}_{x'}(\nu)|_T, \quad |\mathcal{A}_{vN}(\mathbf{I}_x(\nu))| = |\mathcal{A}_{vN}(\mathbf{I}_{x'}(\nu))|. \quad (3.4)$$

If $\delta_S = 0$ (i.e. $\mathcal{S}(\nu) = \mathbf{O}$ for all nodes ν), then necessarily $\mathbf{x} = \mathbf{x}' = \emptyset$, so that (3.4) holds trivially for \mathcal{S} with $\delta_S = 0$. If $\delta_S = 1$, then necessarily \mathcal{X}_S consists only of one element and $\mathbf{x} = \mathbf{x}' = \mathcal{S}(\nu_1)$, so that (3.4) also holds trivially for \mathcal{S} with $\delta_S = 1$. We now proceed by induction on δ_S . Suppose that for $m \geq 1$, (3.4) holds for all \mathcal{S} with $\delta_S \leq m$. Consider a status tree \mathcal{S} with $\delta_S = m + 1$, and two sequences $\mathbf{x}, \mathbf{x}' \in \mathcal{X}_S$. Then $\mathcal{S}|_{\Delta(\nu_{10})}$ and $\mathcal{S}|_{\Delta(\nu_{11})}$ are both status trees of depth less than or equal to m . For $i = 0, 1$, if a node ν of level η is such that $\nu \in \Delta(\nu_{1i})$, then the level of ν is $\eta - 1$ with respect to $\Delta(\nu_{1i})$. Moreover, for $i = 0, 1$, the status tree $\mathcal{S}|_{\Delta(\nu_{1i})}$ is derived from each of the two sequences $\mathbf{I}_x(\nu_{1i})$ and $\mathbf{I}_{x'}(\nu_{1i})$. For $\nu \in \Delta(\nu_{1i})$, the input sequence $\mathbf{I}_x(\nu)$ ($\mathbf{I}_{x'}(\nu)$, resp.) at ν derived from \mathbf{x} (\mathbf{x}' , resp.) is exactly the input sequence at ν derived from $\mathbf{I}_x(\nu_{1i})$ ($\mathbf{I}_{x'}(\nu_{1i})$, resp.) with

respect to $\Delta(\nu_{1i})$. By the induction hypothesis, we have for nodes ν of level ≥ 2 ,

$$|\mathbf{I}_x(\nu)|_{\mathbf{H}} = |\mathbf{I}_{x'}(\nu)|_{\mathbf{H}}, \quad |\mathbf{I}_x(\nu)|_{\mathbf{T}} = |\mathbf{I}_{x'}(\nu)|_{\mathbf{T}}, \quad |\mathcal{A}_{\mathbf{VN}}(\mathbf{I}_x(\nu))| = |\mathcal{A}_{\mathbf{VN}}(\mathbf{I}_{x'}(\nu))|. \quad (3.5)$$

By (3.2) and (3.5),

$$\begin{aligned} |\mathbf{I}_x(\nu_1)|_{\mathbf{H}} &= |\mathbf{x}|_{\mathbf{H}} = |\lambda(\mathbf{x})|_{\mathbf{H}} + 2|\rho(\mathbf{x})|_{\mathbf{H}} + \mathbf{1}(\sigma(\mathbf{x}) = \mathbf{H}) \\ &= |\mathbf{I}_x(\nu_{10})|_{\mathbf{H}} + 2|\mathbf{I}_x(\nu_{11})|_{\mathbf{H}} + \mathbf{1}\{\mathcal{S}_x(\nu_1) = \mathbf{H}\} \\ &= |\mathbf{I}_{x'}(\nu_{10})|_{\mathbf{H}} + 2|\mathbf{I}_{x'}(\nu_{11})|_{\mathbf{H}} + \mathbf{1}\{\mathcal{S}_{x'}(\nu_1) = \mathbf{H}\} \\ &= |\mathbf{x}'|_{\mathbf{H}} = |\mathbf{I}_{x'}(\nu_1)|_{\mathbf{H}} \end{aligned}$$

and similarly, by (3.3) and (3.5), $|\mathbf{I}_x(\nu_1)|_{\mathbf{T}} = |\mathbf{I}_{x'}(\nu_1)|_{\mathbf{T}}$. Moreover, by (3.1) and (3.5),

$$\begin{aligned} |\mathcal{A}_{\mathbf{VN}}(\mathbf{I}_x(\nu_1))| &= |\mathcal{A}_{\mathbf{VN}}(\mathbf{x})| = |\lambda(\mathbf{x})|_{\mathbf{H}} \\ &= |\mathbf{I}_x(\nu_{10})|_{\mathbf{H}} \\ &= |\mathbf{I}_{x'}(\nu_{10})|_{\mathbf{H}} = |\lambda(\mathbf{x}')|_{\mathbf{H}} \\ &= |\mathcal{A}_{\mathbf{VN}}(\mathbf{I}_{x'}(\nu_1))|. \end{aligned}$$

This shows that (3.4) holds for \mathcal{S} with $\delta_{\mathcal{S}} = m + 1$. The proof is complete. \square

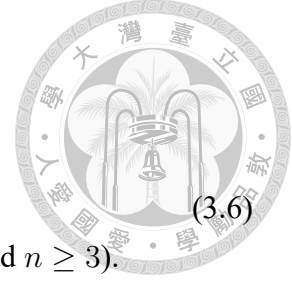
Remark 3.2. For a status tree \mathcal{S} and the corresponding set of sequences $\mathcal{X}_{\mathcal{S}} = \{\mathbf{x} : \mathcal{S}_x = \mathcal{S}\}$, let $n_{\mathcal{S}}(\nu)$, $k_{\mathcal{S}}(\nu)$, $\ell_{\mathcal{S}}$, $\ell_{\mathcal{S}}(\nu)$ be non-negative integers such that $n_{\mathcal{S}}(\nu) = |\mathbf{I}_x(\nu)|$, $k_{\mathcal{S}}(\nu) = |\mathbf{I}_x(\nu)|_{\mathbf{H}}$, $n_{\mathcal{S}}(\nu) - k_{\mathcal{S}}(\nu) = |\mathbf{I}_x(\nu)|_{\mathbf{T}}$, $\ell_{\mathcal{S}} = |\mathcal{A}_{\mathbf{P}}(\mathbf{x})|$, $\ell_{\mathcal{S}}(\nu) = |\mathcal{A}_{\mathbf{VN}}(\mathbf{I}_x(\nu))|$ for all $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$. Thus, for a given status tree \mathcal{S} , the above quantities are known. We write $n_{\mathcal{S}} = n_{\mathcal{S}}(\nu_1)$ and $k_{\mathcal{S}} = k_{\mathcal{S}}(\nu_1)$. Furthermore, for all $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$, $\mathbb{P}_p(\mathbf{X} = \mathbf{x}) = p^{k_{\mathcal{S}}} q^{n_{\mathcal{S}} - k_{\mathcal{S}}}$, independent of \mathbf{x} . In other words, this probability is also an \mathcal{S} -property.

Remark 3.3. A status tree \mathcal{S} of depth $\delta_{\mathcal{S}} \geq 1$ with $\ell_{\mathcal{S}} = 0$ is referred to as *trivial*. A

sequence \mathbf{x} of length n for which $\mathcal{S}_{\mathbf{x}}$ is trivial must be

$$\text{either } \mathbf{x} = \text{H} \cdots \text{H}, \text{ or } \mathbf{x} = \text{T} \cdots \text{T},$$

$$\text{or } \mathbf{x} = \text{H} \cdots \text{HT (for odd } n \geq 3), \text{ or } \mathbf{x} = \text{T} \cdots \text{TH (for odd } n \geq 3).$$



(3.6)

Furthermore, by Lemma 3.2 below, we have $|\mathcal{X}_{\mathcal{S}}| = 1$ for a trivial status tree \mathcal{S} . Indeed, if \mathbf{x} satisfies (3.6), then $\mathcal{X}_{\mathcal{S}_{\mathbf{x}}} = \{\mathbf{x}\}$. (The status tree \mathcal{S} of depth $\delta_{\mathcal{S}} = 0$ is also trivial for which $\mathcal{X}_{\mathcal{S}} = \{\emptyset\}$.)

Lemma 3.1. *For a status tree \mathcal{S} , the \mathcal{S} -properties $n_{\mathcal{S}}(\nu)$, $k_{\mathcal{S}}(\nu)$, $\ell_{\mathcal{S}}(\nu)$ satisfy the following conditions. For each node ν ,*

$$\ell_{\mathcal{S}}(\nu) = k_{\mathcal{S}}(\lambda\nu), \quad n_{\mathcal{S}}(\rho\nu) = n_{\mathcal{S}}(\lambda\nu) - k_{\mathcal{S}}(\lambda\nu),$$

$$k_{\mathcal{S}}(\nu) = k_{\mathcal{S}}(\lambda\nu) + 2k_{\mathcal{S}}(\rho\nu) + \mathbf{1}(\mathcal{S}(\nu) = \text{H}),$$

$$n_{\mathcal{S}}(\nu) - k_{\mathcal{S}}(\nu) = k_{\mathcal{S}}(\lambda\nu) + 2(n_{\mathcal{S}}(\rho\nu) - k_{\mathcal{S}}(\rho\nu)) + \mathbf{1}(\mathcal{S}(\nu) = \text{T}).$$

Proof. For $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$, we have by (3.1)–(3.3),

$$|\mathcal{A}_{\text{vN}}(\mathbf{I}_{\mathbf{x}}(\nu))| = |\mathbf{I}_{\mathbf{x}}(\lambda\nu)|_{\text{H}}, \quad |\mathbf{I}_{\mathbf{x}}(\rho\nu)| = |\mathbf{I}_{\mathbf{x}}(\lambda\nu)|_{\text{T}}$$

$$|\mathbf{I}_{\mathbf{x}}(\nu)|_{\text{H}} = |\mathbf{I}_{\mathbf{x}}(\lambda\nu)|_{\text{H}} + 2|\mathbf{I}_{\mathbf{x}}(\rho\nu)|_{\text{H}} + \mathbf{1}(\mathcal{S}(\nu) = \text{H}),$$

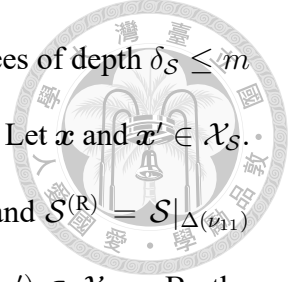
$$|\mathbf{I}_{\mathbf{x}}(\nu)|_{\text{T}} = |\mathbf{I}_{\mathbf{x}}(\lambda\nu)|_{\text{H}} + 2|\mathbf{I}_{\mathbf{x}}(\rho\nu)|_{\text{T}} + \mathbf{1}(\mathcal{S}(\nu) = \text{T}),$$

from which the lemma follows. □

Lemma 3.2. *If \mathcal{S} is a trivial status tree (i.e. $\ell_{\mathcal{S}} = 0$), then $|\mathcal{X}_{\mathcal{S}}| = 1$.*

Proof. For \mathcal{S} with $\delta_{\mathcal{S}} = 0$, we have $\mathcal{S}(\nu) = \text{O}$ for all nodes ν , implying that $\mathcal{X}_{\mathcal{S}} = \{\emptyset\}$.

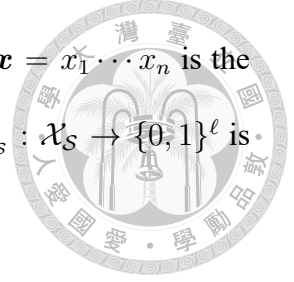
So $|\mathcal{X}_{\mathcal{S}}| = 1$. For \mathcal{S} with $\delta_{\mathcal{S}} = 1$, we have $\mathcal{S}(\nu_1) = \text{H}$ or T and $\mathcal{S}(\nu) = \text{O}$ for all nodes ν of level ≥ 2 . Clearly, we have $\mathcal{X}_{\mathcal{S}} = \{\text{H}\}$ or $\{\text{T}\}$, implying that $|\mathcal{X}_{\mathcal{S}}| = 1$. We now



proceed by induction on $\delta_{\mathcal{S}}$. Suppose $|\mathcal{X}_{\mathcal{S}}| = 1$ for all trivial status trees of depth $\delta_{\mathcal{S}} \leq m$ for some $m \geq 1$. Consider a trivial status tree \mathcal{S} of depth $\delta_{\mathcal{S}} = m + 1$. Let \mathbf{x} and $\mathbf{x}' \in \mathcal{X}_{\mathcal{S}}$. We need to show $\mathbf{x} = \mathbf{x}'$. Since \mathcal{S} is trivial, both $\mathcal{S}^{(L)} = \mathcal{S}|_{\Delta(\nu_{10})}$ and $\mathcal{S}^{(R)} = \mathcal{S}|_{\Delta(\nu_{11})}$ are trivial of depth $\leq m$. We have $\lambda(\mathbf{x}), \lambda(\mathbf{x}') \in \mathcal{X}_{\mathcal{S}^{(L)}}$ and $\rho(\mathbf{x}), \rho(\mathbf{x}') \in \mathcal{X}_{\mathcal{S}^{(R)}}$. By the induction hypothesis, $|\mathcal{X}_{\mathcal{S}^{(L)}}| = |\mathcal{X}_{\mathcal{S}^{(R)}}| = 1$, so that $\lambda(\mathbf{x}) = \lambda(\mathbf{x}')$ and $\rho(\mathbf{x}) = \rho(\mathbf{x}')$. Since $\mathcal{A}_{\vee N}(\mathbf{x}) = \mathcal{A}_{\vee N}(\mathbf{x}') = \emptyset$ and $\sigma(\mathbf{x}) = \sigma(\mathbf{x}') = \mathcal{S}(\nu_1)$, we have $\mathbf{x} = \mathbf{x}'$. (Recall that $\lambda(\mathbf{x}), \rho(\mathbf{x}), \mathcal{A}_{\vee N}(\mathbf{x})$ and $\sigma(\mathbf{x})$ together determine \mathbf{x} .) The proof is complete. \square

Proposition 3.2. *Let \mathcal{S} be a status tree and $\mathcal{X}_{\mathcal{S}} = \{\mathbf{x} : \mathcal{S}_{\mathbf{x}} = \mathcal{S}\}$. Let $n(\nu) = n_{\mathcal{S}}(\nu)$, $k(\nu) = k_{\mathcal{S}}(\nu)$, $\ell = \ell_{\mathcal{S}}$, $\ell(\nu) = \ell_{\mathcal{S}}(\nu)$ be the \mathcal{S} -properties such that $|\mathbf{I}_{\mathbf{x}}(\nu)| = n(\nu)$, $|\mathbf{I}_{\mathbf{x}}(\nu)|_{\text{H}} = k(\nu)$, $|\mathcal{A}_{\text{P}}(\mathbf{x})| = \ell$, $|\mathcal{A}_{\vee N}(\mathbf{I}_{\mathbf{x}}(\nu))| = \ell(\nu)$ for all nodes ν and all $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$. Assume $\ell \geq 1$. Then for $\mathbf{y} \in \{0, 1\}^{\ell}$, there is a unique $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$ such that $\mathcal{A}_{\text{P}}(\mathbf{x}) = \mathbf{y}$. That is, $\mathcal{A}_{\vee N}(\mathbf{I}_{\mathbf{x}}(\nu_1))$ consists of the first $\ell(\nu_1)$ bits of \mathbf{y} , $\mathcal{A}_{\vee N}(\mathbf{I}_{\mathbf{x}}(\nu_{10}))$ consists of the next $\ell(\nu_{10})$ bits of \mathbf{y} , and so on, according to the node ordering \prec' . In other words, $\mathcal{A}_{\text{P}}|_{\mathcal{X}_{\mathcal{S}}} : \mathcal{X}_{\mathcal{S}} \rightarrow \{0, 1\}^{\ell}$ is 1-1 and onto (i.e. bijective). In particular, $|\mathcal{X}_{\mathcal{S}}| = 2^{\ell}$.*

Proof. Although the proposition only considers \mathcal{S} with $\ell_{\mathcal{S}} \geq 1$, we will say that the proposition holds for \mathcal{S} if either $\ell = \ell_{\mathcal{S}} = 0$ (i.e. \mathcal{S} is trivial) or $\ell = \ell_{\mathcal{S}} \geq 1$ and $\mathcal{A}_{\text{P}}|_{\mathcal{X}_{\mathcal{S}}} : \mathcal{X}_{\mathcal{S}} \rightarrow \{0, 1\}^{\ell}$ is 1-1 and onto. Thus, the proposition holds trivially for \mathcal{S} with $\ell_{\mathcal{S}} = 0$. Since a status tree \mathcal{S} of depth $\delta_{\mathcal{S}} \leq 1$ has $\ell_{\mathcal{S}} = 0$, the proposition holds for \mathcal{S} with $\delta_{\mathcal{S}} \leq 1$. We proceed by induction on $\delta_{\mathcal{S}}$. Suppose the proposition holds for all \mathcal{S} with $\delta_{\mathcal{S}} \leq m$ for some $m \geq 1$. Consider a status tree \mathcal{S} with $\delta_{\mathcal{S}} = m + 1$ and $\ell = \ell_{\mathcal{S}} \geq 1$. Let $\mathcal{S}^{(L)} = \mathcal{S}|_{\Delta(\nu_{10})}$ and $\mathcal{S}^{(R)} = \mathcal{S}|_{\Delta(\nu_{11})}$. Then $\delta_{\mathcal{S}^{(L)}} = m$ and $\delta_{\mathcal{S}^{(R)}} \leq m$. If $\delta_{\mathcal{S}^{(R)}} = 0$, then $\mathcal{S}^{(R)}$ is the trivial status tree for which the status of each node is O. It follows that for $\mathbf{x} = x_1 x_2 \cdots x_n \in \mathcal{X}_{\mathcal{S}}$ where $n = n_{\mathcal{S}}(\nu_1)$, we have $x_1 \neq x_2, x_3 \neq x_4, \dots, x_{2\lfloor \frac{n}{2} \rfloor - 1} \neq x_{2\lfloor \frac{n}{2} \rfloor}$. Given $\mathbf{y} = y_1 \cdots y_{\ell}$, let $x_{2i-1} x_{2i} = \text{HT}$ or TH according as $y_i = 1$



or 0, $i = 1, \dots, \lfloor \frac{n}{2} \rfloor$. If n is odd, let $x_n = \mathcal{S}(\nu_1)$. It is readily seen $\mathbf{x} = x_1 \cdots x_n$ is the unique sequence in $\mathcal{X}_{\mathcal{S}}$ such that $\mathcal{A}_{\mathcal{P}}(\mathbf{x}) = \mathbf{y}$. This shows that $\mathcal{A}_{\mathcal{P}}|_{\mathcal{X}_{\mathcal{S}}} : \mathcal{X}_{\mathcal{S}} \rightarrow \{0, 1\}^{\ell}$ is 1-1 and onto.

We now assume $1 \leq \delta_{\mathcal{S}^{(R)}} \leq m$. Letting $\ell^{(L)} = \sum_{\nu \in \Delta(\nu_{10})} \ell(\nu)$ and $\ell^{(R)} = \sum_{\nu \in \Delta(\nu_{11})} \ell(\nu)$, for $\mathbf{y} \in \{0, 1\}^{\ell}$, write $\mathbf{y} = \mathbf{y}'\mathbf{y}^{(L)}\mathbf{y}^{(R)}$ where \mathbf{y}' consists of the first $\ell(\nu_1)$ bits of \mathbf{y} , $\mathbf{y}^{(L)}$ consists of the next $\ell^{(L)}$ bits of \mathbf{y} , and $\mathbf{y}^{(R)}$ consists of the remaining $\ell^{(R)}$ bits of \mathbf{y} . By convention, $\mathbf{y}' = \emptyset$ for $\ell(\nu_1) = 0$, $\mathbf{y}^{(L)} = \emptyset$ for $\ell^{(L)} = 0$, and $\mathbf{y}^{(R)} = \emptyset$ for $\ell^{(R)} = 0$. If $\ell^{(L)} = 0$, then $\mathcal{S}^{(L)}$ is trivial and $|\mathcal{X}_{\mathcal{S}^{(L)}}| = 1$ by Lemma 3.2, in which case we write $\mathcal{X}_{\mathcal{S}^{(L)}} = \{\mathbf{x}^{(L)}\}$ (i.e. $\mathbf{x}^{(L)}$ denotes the only sequence in $\mathcal{X}_{\mathcal{S}^{(L)}}$). If $\ell^{(L)} > 0$, by the induction hypothesis applied to $\mathcal{S}^{(L)}$ and $\mathbf{y}^{(L)}$, there is a unique $\mathbf{x}^{(L)} \in \mathcal{X}_{\mathcal{S}^{(L)}}$ such that $\mathcal{A}_{\mathcal{P}}(\mathbf{x}^{(L)}) = \mathbf{y}^{(L)}$. Similarly, if $\ell^{(R)} = 0$, then $\mathcal{S}^{(R)}$ is trivial and let $\mathbf{x}^{(R)}$ denote the only sequence in $\mathcal{X}_{\mathcal{S}^{(R)}}$. If $\ell^{(R)} > 0$, by the induction hypothesis applied to $\mathcal{S}^{(R)}$ and $\mathbf{y}^{(R)}$, there is a unique $\mathbf{x}^{(R)} \in \mathcal{X}_{\mathcal{S}^{(R)}}$ such that $\mathcal{A}_{\mathcal{P}}(\mathbf{x}^{(R)}) = \mathbf{y}^{(R)}$. Note by Lemma 3.1 that

$$|\mathbf{x}^{(L)}| = n(\nu_{10}), \quad |\mathbf{x}^{(L)}|_{\text{H}} = k(\nu_{10}) = \ell(\nu_1) = |\mathbf{y}'|,$$

$$|\mathbf{x}^{(R)}| = n(\nu_{11}) = n(\nu_{10}) - k(\nu_{10}) = |\mathbf{x}^{(L)}| - |\mathbf{x}^{(L)}|_{\text{H}}.$$

It follows that there is a unique $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$ such that

$$\lambda(\mathbf{x}) = \mathbf{x}^{(L)}, \quad \rho(\mathbf{x}) = \mathbf{x}^{(R)}, \quad \text{and} \quad \mathcal{A}_{\text{VN}}(\mathbf{x}) = \mathbf{y}'.$$

(Recall that $\lambda(\mathbf{x})$, $\rho(\mathbf{x})$, $\mathcal{A}_{\text{VN}}(\mathbf{x})$ and $\sigma(\mathbf{x})$ together determine \mathbf{x} .) We have shown that there is a unique $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$ such that

$$\mathcal{A}_{\mathcal{P}}(\mathbf{x}) = \mathcal{A}_{\text{VN}}(\mathbf{x}) \mathcal{A}_{\mathcal{P}}(\lambda(\mathbf{x})) \mathcal{A}_{\mathcal{P}}(\rho(\mathbf{x})) = \mathcal{A}_{\text{VN}}(\mathbf{x}) \mathcal{A}_{\mathcal{P}}(\mathbf{x}^{(L)}) \mathcal{A}_{\mathcal{P}}(\mathbf{x}^{(R)}) = \mathbf{y}'\mathbf{y}^{(L)}\mathbf{y}^{(R)} = \mathbf{y}.$$

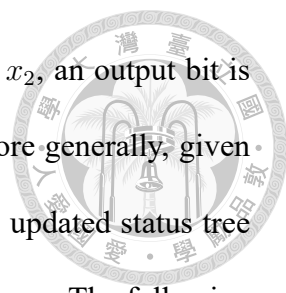


The proof is complete.

Remark 3.4. Peres [20] proved that given $|\mathcal{A}_P(\mathbf{X}_n)| = \ell \geq 1$, the conditional distribution of $\mathcal{A}_P(\mathbf{X}_n)$ is uniform on $\{0, 1\}^\ell$. By Proposition 3.2, if we further condition on the status tree $\mathcal{S}_X = \mathcal{S}$ (with $\ell_S = \ell$), the conditional distribution of $\mathcal{A}_P(\mathbf{X})$ remains uniform on $\{0, 1\}^\ell$. Actually, Peres [20] proved that if \mathbf{X} is an exchangeable sequence (which is more general than an iid sequence), given $|\mathcal{A}_P(\mathbf{X})| = \ell$, the conditional distribution of $\mathcal{A}_P(\mathbf{X})$ is uniform on $\{0, 1\}^\ell$. Since for an exchangeable sequence \mathbf{X} , the probability of $\mathbf{X} = \mathbf{x}$ is the same for all $\mathbf{x} \in \mathcal{X}_S$, it follows that given $\mathcal{S}_X = \mathcal{S}$ with $\ell = \ell_S \geq 1$, the conditional distribution of $\mathcal{A}_P(\mathbf{X})$ remains uniform on $\{0, 1\}^\ell$ when \mathbf{X} is an exchangeable sequence.

Remark 3.5. While $\mathcal{A}_P(\mathbf{x})$ is the total collection of the (non-empty) output sequences at all nodes arranged according to the node ordering \prec' , \prec' may be replaced by any other node ordering without losing the unbiasedness property. For example, let \prec'' be the ordering such that $\nu_{(1)} \prec'' \nu_{(2)} \prec'' \nu_{(3)} \prec'' \dots$, where $\nu_{(v)} = \nu_{b_1 \dots b_\eta}$ with $v = \sum_{i=1}^{\eta} 2^{n-i} b_i$. Let $\mathcal{A}'_P(\mathbf{x}) = \mathcal{A}_{vN}(\mathbf{I}_x(\nu_{(1)})) \mathcal{A}_{vN}(\mathbf{I}_x(\nu_{(2)})) \mathcal{A}_{vN}(\mathbf{I}_x(\nu_{(3)})) \dots$, which is the collection of the output sequence at all nodes arranged according to the node ordering \prec'' . Then for a status tree \mathcal{S} with $\ell = \ell_S \geq 1$, given $\mathcal{S}_X = \mathcal{S}$, the conditional distribution of $\mathcal{A}'_P(\mathbf{X})$ is uniform on $\{0, 1\}^\ell$. This is due to the fact that the number of output bits at each node is an \mathcal{S} -property so that for $\mathbf{x} \in \mathcal{X}_S$, there is a 1-1 correspondence between $\mathcal{A}_P(\mathbf{x})$ and $\mathcal{A}'_P(\mathbf{x})$.

Remark 3.6. It is instructive to describe how the status tree evolves (changes) as input bits arrive at ν_1 one after another. Initially, at time 0, we have the trivial status tree $\tilde{\mathcal{S}}$ with $\tilde{\mathcal{S}}(\nu) = \mathbf{O}$ for all nodes ν . We write $\mathcal{S}_0 = \tilde{\mathcal{S}}$. At time 1 when an input bit x_1 arrives at ν_1 , the status tree becomes \mathcal{S}_1 with $\mathcal{S}_1(\nu_1) = x_1$ and $\mathcal{S}_1(\nu) = \mathbf{O}$ for $\nu \neq \nu_1$. At time 2 when a second input bit x_2 arrives at ν_1 , the status tree becomes \mathcal{S}_2 with (i) $\mathcal{S}_2(\nu) = \mathbf{O}$ for $\nu = \nu_1$ and for all nodes ν of level ≥ 3 , (ii) $\mathcal{S}_2(\nu_{10}) = \mathbf{H}$ and $\mathcal{S}_2(\nu_{11}) = \mathbf{O}$ if $x_1 \neq x_2$,



(iii) $\mathcal{S}_2(\nu_{10}) = T$ and $\mathcal{S}_2(\nu_{11}) = x_1$ if $x_1 = x_2$. Moreover, if $x_1 \neq x_2$, an output bit is generated at ν_1 , which is 0 or 1 according as $x_1x_2 = TH$ or HT . More generally, given the status tree \mathcal{S}_n at time n , when an input bit x_{n+1} arrives at ν_1 , the updated status tree \mathcal{S}_{n+1} and the output bits induced by x_{n+1} are determined by \mathcal{S}_n and x_{n+1} . The following procedure describes how the status tree is updated when input bits arrive at ν_1 one after another. Initially, all nodes are labeled as O. When an input symbol (H or T) arrives at ν_1 , each node ν may receive a symbol (H or T) from its parent node $\pi\nu$ ($\pi\nu$ is referred to as the source if $\nu = \nu_1$), and may send a symbol (H or T) to $\lambda\nu$ or $\rho\nu$. Meanwhile, an output bit (0 or 1) may be generated at ν . Specifically, let ν be a node with a label $s \in \{H, T, O\}$ and it receives a symbol $\iota \in \{H, T\}$ from its parent node $\pi\nu$ (or from the source if $\nu = \nu_1$). We do the following operations on ν .

- (i) When $s = O$, set (update) $s = \iota$, and send no symbol to $\lambda\nu$ or $\rho\nu$.
- (ii) If $s\iota = HT$, set $s = O$, output a bit 1, and send a symbol H to $\lambda\nu$.
- (iii) If $s\iota = TH$, set $s = O$, output a bit 0, and send a symbol H to $\lambda\nu$.
- (iv) If $s\iota = HH$, set $s = O$ and send a symbol T to $\lambda\nu$ and a symbol H to $\rho\nu$.
- (v) If $s\iota = TT$, set $s = O$ and send a symbol T to $\lambda\nu$ and a symbol T to $\rho\nu$.

If a node receives no symbol from its parent node, then its label (status) is not updated. Note that if a node receives no symbol from its parent node, then none of its descendant nodes receives a symbol, so that their labels are not updated. When the label of a node ν is updated from H or T to O, it must have received a symbol from $\pi\nu$ and must send a symbol to $\lambda\nu$ and may also send a symbol to $\rho\nu$ if the current label and the received symbol are the same.



3.3 Unbiasedness of \mathcal{A}_S

For a finite random sequence \mathbf{X} , the status tree \mathcal{S}_X generated from \mathbf{X} is random. In this section, we prove that conditioning on $\mathcal{S}_X = \mathcal{S}$ for a given (fixed) status tree \mathcal{S} , the sequence of ℓ_S output bits generated by \mathcal{A}_S are (conditionally) iid unbiased. For notational simplicity, we write $n = n_S$, $k = k_S$, $\ell = \ell_S$, $\ell(\nu) = \ell_S(\nu)$. Note that $\ell = \sum_{\nu} \ell(\nu)$. Assume $\ell \geq 1$. By Proposition 3.2, $\mathcal{X}_S = \{\mathbf{x} : \mathcal{S}_x = \mathcal{S}\}$ consists of 2^ℓ sequences of length n each of which has k H's and $n - k$ T's. Each sequence of \mathcal{X}_S yields an output sequence of $\{0, 1\}^\ell$. We will show that $\mathcal{A}_S|_{\mathcal{X}_S} : \mathcal{X}_S \rightarrow \{0, 1\}^\ell$ is 1-1 (and hence onto).

It is worth noting that if \mathcal{A}'_S is used instead of \mathcal{A}_S , there may be two different \mathbf{x} and $\mathbf{x}' \in \mathcal{X}_S$ such that $\mathcal{A}'_S(\mathbf{x}) = \mathcal{A}'_S(\mathbf{x}')$. Specifically, consider the status tree \mathcal{S} given by $\mathcal{S}(\nu) = \text{O}$ for all nodes ν except $\mathcal{S}(\nu_{1000}) = \text{T}$, $\mathcal{S}(\nu_{1001}) = \text{H}$, $\mathcal{S}(\nu_{110}) = \text{H}$. (See Figure 3.1 in which those nodes not explicitly shown have status of O.) It is readily seen that

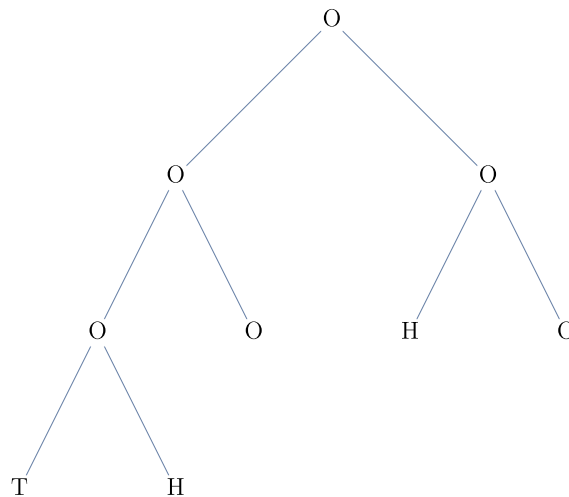
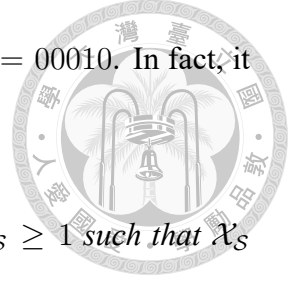


Figure 3.1: The status tree \mathcal{S} given by $\mathcal{S}(\nu) = \text{O}$ for all nodes ν except $\mathcal{S}(\nu_{1000}) = \text{T}$, $\mathcal{S}(\nu_{1001}) = \text{H}$, $\mathcal{S}(\nu_{110}) = \text{H}$.

$n = n_S = 8$, $k = k_S = 4$, $\ell = \ell_S = 5$, $\ell(\nu_1) = \ell_S(\nu_1) = 2$, $\ell(\nu_{10}) = \ell_S(\nu_{10}) = 2$, $\ell(\nu_{11}) = \ell_S(\nu_{11}) = 1$. Moreover, $\mathcal{A}'_S(\mathbf{x}) = \mathcal{A}'_S(\mathbf{x}') = 00010$ for $\mathbf{x} = \text{TTTHTHHH}$ and

$\mathbf{x}' = \text{TTTHHHHT}$. On the other hand, $\mathcal{A}_S(\mathbf{x}) = 00001$ and $\mathcal{A}_S(\mathbf{x}') = 00010$. In fact, it can be shown that $\mathcal{A}_S|_{\mathcal{X}_S} : \mathcal{X}_S \rightarrow \{0, 1\}^5$ is 1-1 and onto.

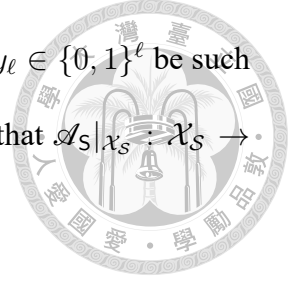


Proposition 3.3. *For a given status tree \mathcal{S} with $n = n_S$ and $\ell = \ell_S \geq 1$ such that \mathcal{X}_S consists of 2^ℓ sequences of length n , we have that $\mathcal{A}_S|_{\mathcal{X}_S} : \mathcal{X}_S \rightarrow \{0, 1\}^\ell$ is a 1-1 (and hence onto). Consequently, given $\mathcal{S}_X = \mathcal{S}$, the conditional distribution of $\mathcal{A}_S(\mathbf{X})$ is uniform on $\{0, 1\}^\ell$, which implies that*

$$\mathbb{P}_p(\mathcal{A}_S(\mathbf{X}) = \beta_1 \cdots \beta_\ell \mid |\mathcal{A}_S(\mathbf{X})| = \ell) = \frac{1}{2^\ell}, \quad \text{for all } \beta_i \in \{0, 1\}, i = 1, \dots, \ell.$$

Proof. Although the proposition only considers \mathcal{S} with $\ell_S \geq 1$, we will say that the proposition holds for \mathcal{S} if either $\ell = \ell_S = 0$ (i.e. \mathcal{S} is trivial) or $\ell = \ell_S \geq 1$ and $\mathcal{A}_S|_{\mathcal{X}_S} : \mathcal{X}_S \rightarrow \{0, 1\}^\ell$ is 1-1. Thus, the proposition holds trivially for \mathcal{S} with $\ell_S = 0$. For \mathcal{S} with $n_S = 1$, \mathcal{X}_S must be either $\{\text{H}\}$ or $\{\text{T}\}$, so that $\ell_S = 0$. This shows that the proposition holds trivially for \mathcal{S} with $n_S = 1$. We now proceed by induction on n_S .

Suppose that for some $m \geq 2$, the proposition holds for all status trees \mathcal{S} with $n_S \leq m - 1$. Consider a status tree \mathcal{S} with $n_S = m$. Let $\ell = \ell_S \geq 1$ and $\ell(\nu) = \ell_S(\nu)$ for all nodes ν . Since $m = n_S$ and $\ell = \ell_S$, \mathcal{X}_S consists of 2^ℓ sequences of length m . For $\mathbf{x} = x_1 \cdots x_m \in \mathcal{X}_S$, we write $\mathbf{x}_{-m} = x_1 \cdots x_{m-1}$ (which is \mathbf{x} with the last bit x_m deleted). A node ν is said to be *affected* by x_m (or more precisely, by the last bit x_m of \mathbf{x}) if $\mathbf{I}_\mathbf{x}(\nu) \neq \mathbf{I}_{\mathbf{x}_{-m}}(\nu)$. It is readily seen that ν is affected by x_m if and only if $\mathcal{S}_\mathbf{x}(\nu) \neq \mathcal{S}_{\mathbf{x}_{-m}}(\nu)$. In other words, ν is affected by x_m if and only if the status of ν changes at time m when x_m joins the input sequence at ν_1 . Note that if a node ν is not affected by x_m , then none of its descendant nodes are affected by x_m . Note also that $\mathbf{I}_\mathbf{x}(\nu) = \emptyset$ for all nodes ν of level $> \delta_S$, implying that none of the nodes of level $> \delta_S$ are affected by the last bit x_m of $\mathbf{x} \in \mathcal{X}_S$.



Let $\mathbf{x}' = x'_1 \cdots x'_m \in \mathcal{X}_S$, $\mathbf{x}'' = x''_1 \cdots x''_m \in \mathcal{X}_S$ and $\mathbf{y} = y_1 \cdots y_\ell \in \{0, 1\}^\ell$ be such that $\mathcal{A}_S(\mathbf{x}') = \mathcal{A}_S(\mathbf{x}'') = \mathbf{y}$. We want to show $\mathbf{x}' = \mathbf{x}''$ (implying that $\mathcal{A}_S|_{\mathcal{X}_S : \mathcal{X}_S \rightarrow \{0, 1\}^\ell}$ is 1-1). We make the following claim.

Claim (A):

We have $x'_m = x''_m$ and $\mathcal{S}_{\mathbf{x}'_{-m}} = \mathcal{S}_{\mathbf{x}''_{-m}}$ (i.e. the status trees derived from \mathbf{x}'_{-m} and \mathbf{x}''_{-m} are identical).

Assume for now that claim (A) holds. In view of $\mathcal{S}_{\mathbf{x}'_{-m}} = \mathcal{S}_{\mathbf{x}''_{-m}}$ and $x'_m = x''_m$, for any node ν , x'_m induces an output bit at ν if and only if x''_m induces the same output bit at ν (cf. Remark 3.6). Since $\mathcal{A}_S(\mathbf{x}') = \mathcal{A}_S(\mathbf{x}'') (= \mathbf{y})$, we must have $\mathcal{A}_S(\mathbf{x}'_{-m}) = \mathcal{A}_S(\mathbf{x}''_{-m})$. Letting $\mathcal{S}^* = \mathcal{S}_{\mathbf{x}'_{-m}} (= \mathcal{S}_{\mathbf{x}''_{-m}})$, if $\ell_{\mathcal{S}^*} = 0$, then $\mathbf{x}'_{-m} = \mathbf{x}''_{-m}$ since $|\mathcal{X}_{\mathcal{S}^*}| = 1$ by Lemma 3.2. If $\ell_{\mathcal{S}^*} > 0$, since $\mathbf{x}'_{-m}, \mathbf{x}''_{-m} \in \mathcal{X}_{\mathcal{S}^*}$ and since $\mathcal{A}_S(\mathbf{x}'_{-m}) = \mathcal{A}_S(\mathbf{x}''_{-m})$, it follows from the induction hypothesis that $\mathbf{x}'_{-m} = \mathbf{x}''_{-m}$. In either case, we have $\mathbf{x}'_{-m} = \mathbf{x}''_{-m}$, which together with $x'_m = x''_m$ implies that $\mathbf{x}' = \mathbf{x}''$.

It remains to establish claim (A). If $\mathcal{S}(\nu_1) = \text{H or T}$ (i.e. m is odd), then $x'_m = x''_m = \mathcal{S}(\nu_1)$, and ν_1 is the only node affected by x'_m (and x''_m). So $\mathcal{S}_{\mathbf{x}'_{-m}}(\nu_1) = \mathcal{S}_{\mathbf{x}''_{-m}}(\nu_1) = \text{O}$ and $\mathcal{S}_{\mathbf{x}'_{-m}}(\nu) = \mathcal{S}_{\mathbf{x}''_{-m}}(\nu) = \mathcal{S}(\nu)$ for $\nu \neq \nu_1$. This proves claim (A) for the case $\mathcal{S}(\nu_1) = \text{H or T}$. Next suppose $\mathcal{S}(\nu_1) = \text{O}$ (i.e. m is even). For any $\mathbf{x} = x_1 \cdots x_m \in \mathcal{X}_S$, we must have that $\mathcal{S}_{\mathbf{x}_{-m}}(\nu_1) \neq \text{O}$ and ν_{10} is affected by x_m . If $\mathcal{S}(\nu_{10}) = \text{O}$, then $\mathcal{S}_{\mathbf{x}_{-m}}(\nu_{10}) \neq \text{O}$ and ν_{100} is affected by x_m . More generally, $\lambda^i \nu_1, i = 0, 1, \dots, r$ are affected by x_m where $r = \min\{i \geq 1 : \mathcal{S}(\lambda^i \nu_1) \neq \text{O}\} \geq 1$. Clearly, $\mathcal{S}_{\mathbf{x}_{-m}}(\lambda^i \nu_1) \neq \text{O}$ for $i = 0, \dots, r-1$, and $\mathcal{S}_{\mathbf{x}_{-m}}(\lambda^r \nu_1) = \text{O}$. Moreover, all descendant nodes of $\lambda^r \nu_1$ are not affected by x_m . So $\mathcal{S}_{\mathbf{x}_{-m}}(\nu) = \mathcal{S}(\nu)$ for all descendant nodes ν of $\lambda^r \nu_1$.

A node ν is said to be *good* if

$$\mathbf{I}_{\mathbf{x}'}(\nu) \setminus \mathbf{I}_{\mathbf{x}'_{-m}}(\nu) = \mathbf{I}_{\mathbf{x}''}(\nu) \setminus \mathbf{I}_{\mathbf{x}''_{-m}}(\nu) \quad \text{and} \quad \mathcal{S}_{\mathbf{x}'_{-m}}(\nu) = \mathcal{S}_{\mathbf{x}''_{-m}}(\nu).$$



The first equality implies that one of the two cases occurs:

- (i) node ν is neither affected by x'_m nor by x''_m ;
- (ii) ν is affected by x'_m and by x''_m and the input bit at ν derived from x'_m is the same as that derived from x''_m .

A good node ν is said to be *great* if

$$\mathbf{I}_{\mathbf{x}'}(\nu) \setminus \mathbf{I}_{\mathbf{x}'_{-m}}(\nu) = \mathbf{I}_{\mathbf{x}''}(\nu) \setminus \mathbf{I}_{\mathbf{x}''_{-m}}(\nu) \neq \emptyset.$$

We have shown that for all $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$, $\mathcal{S}_{\mathbf{x}_{-m}}(\lambda^r \nu_1) = \mathbf{O}$ and $\mathcal{S}_{\mathbf{x}}(\lambda^r \nu_1) = \mathcal{S}(\lambda^r \nu_1) \neq \mathbf{O}$ and all descendant nodes of $\lambda^r \nu_1$ are not affected by x_m , where $r = \min\{i : \mathcal{S}(\lambda^i \nu_1) \neq \mathbf{O}\} \geq 1$. In particular, $\mathcal{S}_{\mathbf{x}'_{-m}}(\lambda^r \nu_1) = \mathcal{S}_{\mathbf{x}''_{-m}}(\lambda^r \nu_1) = \mathbf{O}$ and $\mathcal{S}_{\mathbf{x}'}(\lambda^r \nu_1) = \mathcal{S}_{\mathbf{x}''}(\lambda^r \nu_1) \neq \mathbf{O}$, which implies that the node $\lambda^r \nu_1$ is great. Moreover, all the descendant nodes of $\lambda^r \nu_1$ are good since they are not affected by x'_m or x''_m . Note also that a node $\nu \succ \lambda^r \nu_1$ is necessarily a descendant node of $\lambda^r \nu_1$. So all nodes $\nu \succ \lambda^r \nu_1$ are good. We make the following claim.

Claim (B):

If a node $\nu' \neq \nu_1$ is great and all nodes $\nu \succ \nu'$ are good,

then there is a node $\nu'' \prec \nu'$ such that ν'' is great and all nodes $\nu \succ \nu''$ are good.

If claim (B) holds, then starting with $\nu' = \lambda^r \nu_1$, there is a node $\nu'' \prec \lambda^r \nu_1$ such that ν'' is great and all nodes $\nu \succ \nu''$ are good. If $\nu'' \neq \nu_1$, by claim (B) applied to ν'' , there is



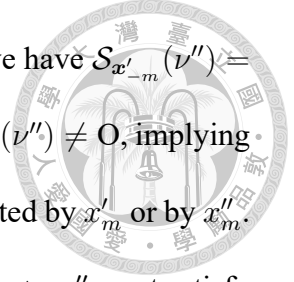
a node $\nu''' \prec \nu''$ such that ν''' is great and all nodes $\nu \succ \nu'''$ are good. Since there are only finitely many nodes that are great, eventually ν_1 will be reached. That is, ν_1 is great and all nodes $\nu \succ \nu_1$ are good, which implies claim (A).

It remains to establish claim (B). Suppose $\nu' \neq \nu_1$ is great and all nodes $\nu \succ \nu'$ are good. We need to show that there is a node $\nu'' \prec \nu'$ such that ν'' is great and all nodes $\nu \succ \nu''$ are good. Consider the following four cases separately.

Case (1): ν' is the left child node of its parent node (denoted $\pi\nu'$) and $\mathbf{I}_{\mathbf{x}'}(\nu') \setminus \mathbf{I}_{\mathbf{x}'_{-m}}(\nu') = \mathbf{I}_{\mathbf{x}''}(\nu') \setminus \mathbf{I}_{\mathbf{x}''_{-m}}(\nu') = \mathbf{H}$. Clearly, x'_m induces an output bit at $\pi\nu'$ and x''_m induces an output bit at $\pi\nu'$. Let $S' = \{\nu \succ \nu' : x'_m \text{ induces an output bit at } \nu\}$ and $S'' = \{\nu \succ \nu' : x''_m \text{ induces an output bit at } \nu\}$. Since all nodes $\nu \succ \nu'$ are good, we have $S' = S''$. Since $\mathcal{A}_S(\mathbf{x}') = \mathcal{A}_S(\mathbf{x}'') = \mathbf{y} = y_1 \cdots y_\ell$, the output bits at the nodes in S' ($= S''$) induced by x'_m (by x''_m) must be $y_{\ell-\gamma+1} \cdots y_\ell$ or \emptyset if $\gamma = 0$, where $\gamma = |S'| = |S''|$. Thus, the output bit at ν' induced by x'_m (and by x''_m) must be $y_{\ell-\gamma}$. If $y_{\ell-\gamma} = 0$, then $\mathcal{S}_{\mathbf{x}'_{-m}}(\pi\nu') = \mathcal{S}_{\mathbf{x}''_{-m}}(\pi\nu') = \mathbf{T}$, and $\mathbf{I}_{\mathbf{x}'}(\pi\nu') \setminus \mathbf{I}_{\mathbf{x}'_{-m}}(\pi\nu') = \mathbf{I}_{\mathbf{x}''}(\pi\nu') \setminus \mathbf{I}_{\mathbf{x}''_{-m}}(\pi\nu') = \mathbf{H}$. If $y_{\ell-\gamma} = 1$, then $\mathcal{S}_{\mathbf{x}'_{-m}}(\pi\nu') = \mathcal{S}_{\mathbf{x}''_{-m}}(\pi\nu') = \mathbf{H}$, and $\mathbf{I}_{\mathbf{x}'}(\pi\nu') \setminus \mathbf{I}_{\mathbf{x}'_{-m}}(\pi\nu') = \mathbf{I}_{\mathbf{x}''}(\pi\nu') \setminus \mathbf{I}_{\mathbf{x}''_{-m}}(\pi\nu') = \mathbf{T}$. So $\pi\nu'$ is great. Moreover, $\rho\pi\nu'$ and its descendant nodes are not affected by x'_m or by x''_m . So $\rho\pi\nu'$ and its descendant nodes are good. Furthermore, a node $\nu \succ \pi\nu'$ must satisfy one of the following conditions: $\nu \succ \nu'$; $\nu = \nu'$; $\nu = \rho\pi\nu'$; ν is a descendant node of $\rho\pi\nu'$. It follows that all nodes $\nu \succ \pi\nu'$ are good. So $\nu'' = \pi\nu'$ satisfies the requirement in claim (B).

Case (2): ν' is the left child node of $\pi\nu'$ and $\mathbf{I}_{\mathbf{x}'}(\nu') \setminus \mathbf{I}_{\mathbf{x}'_{-m}}(\nu') = \mathbf{I}_{\mathbf{x}''}(\nu') \setminus \mathbf{I}_{\mathbf{x}''_{-m}}(\nu') = \mathbf{T}$.

It follows that $\rho\pi\nu'$ (the sibling node of ν') is affected by x'_m and by x''_m . Let $r' = \min\{i : \mathcal{S}(\lambda^i \rho\pi\nu') \neq \mathbf{O}\} \geq 0$. It is readily seen that the nodes $\lambda^i \rho\pi\nu'$, $i = 0, \dots, r'$



are affected by x'_m and by x''_m . Moreover, letting $\nu'' = \lambda^{r'} \rho \pi \nu'$, we have $\mathcal{S}_{x'_m}(\nu'') = \mathcal{S}_{x''_m}(\nu'') = \mathbf{O}$ and $\mathbf{I}_{x'}(\nu'') \setminus \mathbf{I}_{x'_m}(\nu'') = \mathbf{I}_{x''}(\nu'') \setminus \mathbf{I}_{x''_m}(\nu'') = \mathcal{S}(\nu'') \neq \mathbf{O}$, implying that ν'' is great. Furthermore, no descendant nodes of ν'' are affected by x'_m or by x''_m . Thus, all descendant nodes of ν'' are good. Note that every node $\nu \succ \nu''$ must satisfy one of the following conditions: $\nu \succ \nu'$; $\nu = \nu'$; ν is a descendant node of ν'' . So ν'' satisfies the requirement in claim (B).

Case (3): ν' is the right child node of $\pi \nu'$ and $\mathbf{I}_{x'}(\nu') \setminus \mathbf{I}_{x'_m}(\nu') = \mathbf{I}_{x''}(\nu') \setminus \mathbf{I}_{x''_m}(\nu') = \mathbf{H}$.

Let $\nu'' = \pi \nu'$. Necessarily, we have $\mathcal{S}(\nu'') = \mathbf{O}$, and $\mathcal{S}_{x'_m}(\nu'') = \mathcal{S}_{x''_m}(\nu'') = \mathbf{H}$, and $\mathbf{I}_{x'}(\nu'') \setminus \mathbf{I}_{x''_m}(\nu'') = \mathbf{I}_{x''}(\nu'') \setminus \mathbf{I}_{x''_m}(\nu'') = \mathbf{H}$. Thus, ν'' is great. Also, every node $\nu \succ \nu''$ is good since either $\nu = \nu'$ or $\nu \succ \nu'$. So ν'' satisfies the requirement in claim (B).

Case (4): ν' is the right child node of $\pi \nu'$ and $\mathbf{I}_{x'}(\nu') \setminus \mathbf{I}_{x'_m}(\nu') = \mathbf{I}_{x''}(\nu') \setminus \mathbf{I}_{x''_m}(\nu') = \mathbf{T}$.

This case is similar to Case (3). It can be shown that $\nu'' = \pi \nu'$ satisfies the requirement in claim (B).

The proof is complete. □

The proof of Proposition 3.3 contains (implicitly) a procedure to reconstruct \mathbf{x} from $\mathcal{S}_{\mathbf{x}}$ and $\mathcal{A}_{\mathcal{S}}(\mathbf{x})$. As an illustration, consider a status tree \mathcal{S} given in Figure 3.2(a) and $\mathcal{A}_{\mathcal{S}}(\mathbf{x}) = 00001$. It is readily seen that $n = |\mathbf{x}| = 8$ for all $\mathbf{x} \in \mathcal{X}_{\mathcal{S}}$. We write $\mathbf{x}_n = \mathbf{x}_8$ for the “unknown” \mathbf{x} satisfying $\mathcal{A}_{\mathcal{S}}(\mathbf{x}_n) = 00001$. For ν with $\mathbf{I}_{x_n}(\nu) = \emptyset$, let $r_n(\nu) = \min\{i \geq 0 : \mathcal{S}_{x_n}(\lambda^i \nu) \neq \mathbf{O}\}$ and $\nu_n(\nu) = \lambda^{r_n(\nu)} \nu$. Recall that ν is said to be affected by the last term x_n of \mathbf{x}_n if $\mathbf{I}_{x_n}(\nu) \setminus \mathbf{I}_{x_{n-1}}(\nu) \neq \emptyset$. To find the last term x_8 of $\mathbf{x}_8 = x_1 \cdots x_8$, we need to identify all the nodes affected by x_8 backwards (with respect to the ordering

→). By definition, ν_1 is always affected by an input symbol.



(i) The last affected node is $\nu_8(\nu_1) = \nu_{(8)}$.

Since $\mathcal{S}_{x_8}(\nu_{(8)}) = \text{T}$, we have $\mathbf{I}_{x_8}(\nu_{(8)}) \setminus \mathbf{I}_{x_7}(\nu_{(8)}) = \text{T}$ and $\mathcal{S}_{x_7}(\nu_{(8)}) = \text{O}$. As $\mathbf{I}_{x_8}(\nu_{(8)}) \setminus \mathbf{I}_{x_7}(\nu_{(8)}) = \text{T}$,

(ii) the second-to-last affected node is $\nu_8(\rho\nu_{(8)}) = \nu_{(9)}$.

Since $\mathcal{S}_{x_8}(\nu_{(9)}) = \text{H}$ and $\nu_{(9)}$ is a right child node, we have $\mathcal{S}_{x_7}(\nu_{(9)}) = \text{O}$ and

(iii) the third-to-last affected node is $\pi\nu_{(9)} = \nu_{(4)}$,

and $\mathbf{I}_{x_8}(\nu_{(4)}) \setminus \mathbf{I}_{x_7}(\nu_{(4)}) = \text{H}$ and $\mathcal{S}_{x_7}(\nu_{(4)}) = \text{H}$. Since $\mathbf{I}_{x_8}(\nu_{(4)}) \setminus \mathbf{I}_{x_7}(\nu_{(4)}) = \text{H}$ and $\nu_{(4)}$ is a left child node, we have

(iv) the fourth-to-last affected node is $\pi\nu_{(4)} = \nu_{(2)}$ and

an output bit (indeed the last output bit) is induced at $\nu_{(2)}$ by x_8 , which is “1” according to $\mathcal{A}_5(x_8) = 00001$. So $\mathbf{I}_{x_8}(\nu_{(2)}) \setminus \mathbf{I}_{x_7}(\nu_{(2)}) = \text{T}$ and $\mathcal{S}_{x_7}(\nu_{(2)}) = \text{H}$. As $\mathbf{I}_{x_8}(\nu_{(2)}) \setminus \mathbf{I}_{x_7}(\nu_{(2)}) = \text{T}$,

(v) the fifth-to-last affected node is $\nu_8(\rho\nu_{(2)}) = \nu_{(6)}$.

Since $\mathcal{S}_{x_8}(\nu_{(6)}) = \text{H}$, we have $\mathcal{S}_{x_7}(\nu_{(6)}) = \text{O}$ and $\mathbf{I}_{x_8}(\nu_{(6)}) \setminus \mathbf{I}_{x_7}(\nu_{(6)}) = \text{H}$, so that

(vi) the sixth-to-last affected node is $\pi\nu_{(6)} = \nu_{(3)}$

and an output bit (second-to-last output bit) is induced at $\nu_{(3)}$ by x_8 , which is "0" according to $\mathcal{A}_S(x_8) = 00001$. As the output bit is a 0, we have $\mathcal{S}_{x_7}(\nu_{(3)}) = \text{T}$ and $\mathcal{I}_{x_8}(\nu_{(3)}) \setminus \mathcal{I}_{x_7}(\nu_{(3)}) = \text{H}$, implying that $x_7 x_8 = \text{HH}$. Figure 3.2(a)–(h) provides a step-by-step description of identifying the affected nodes along with their labels at time 7.

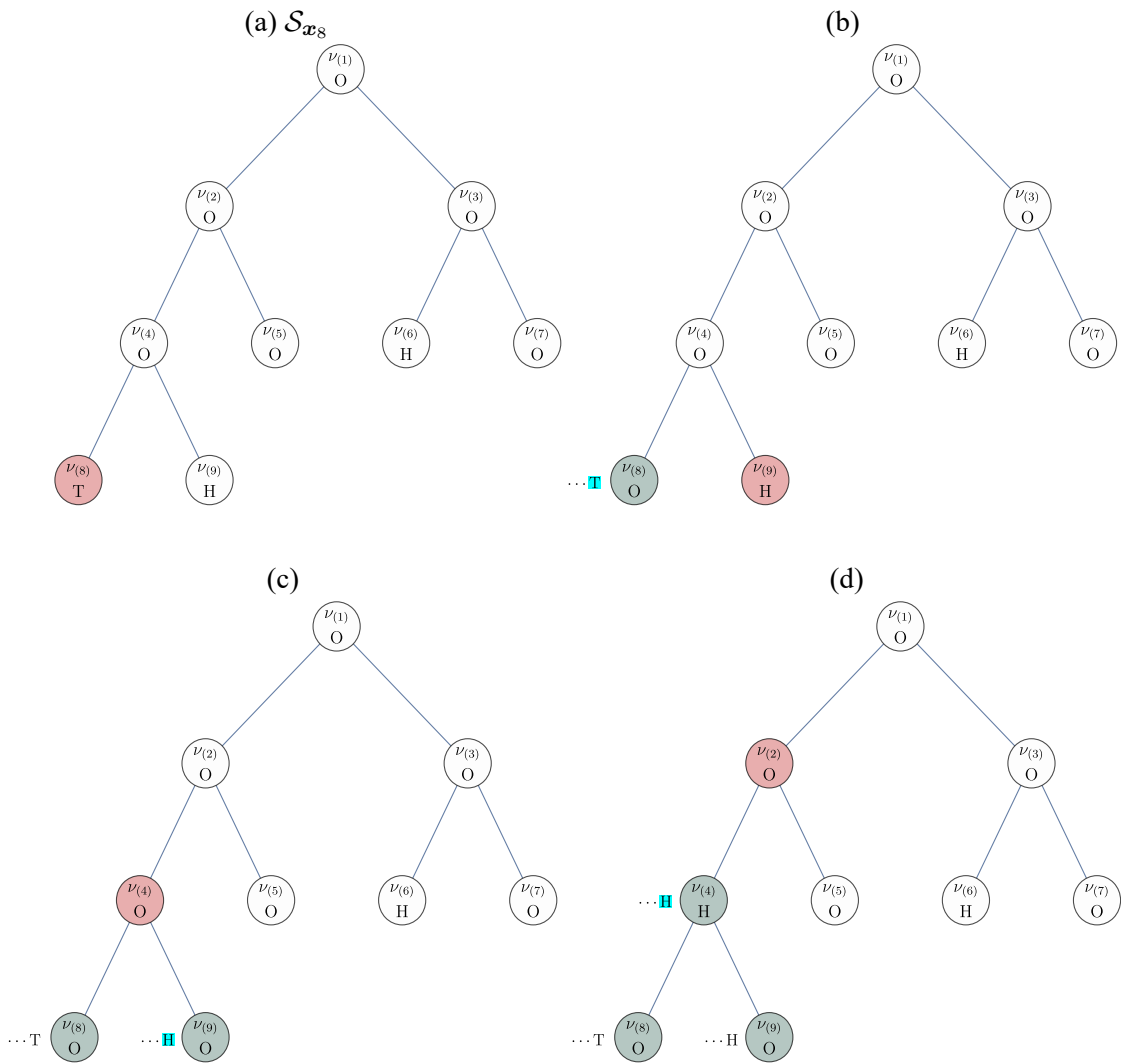


Figure 3.2: An illustration of reconstructing x given \mathcal{S}_x and $\mathcal{A}_S(x) = 00001$.

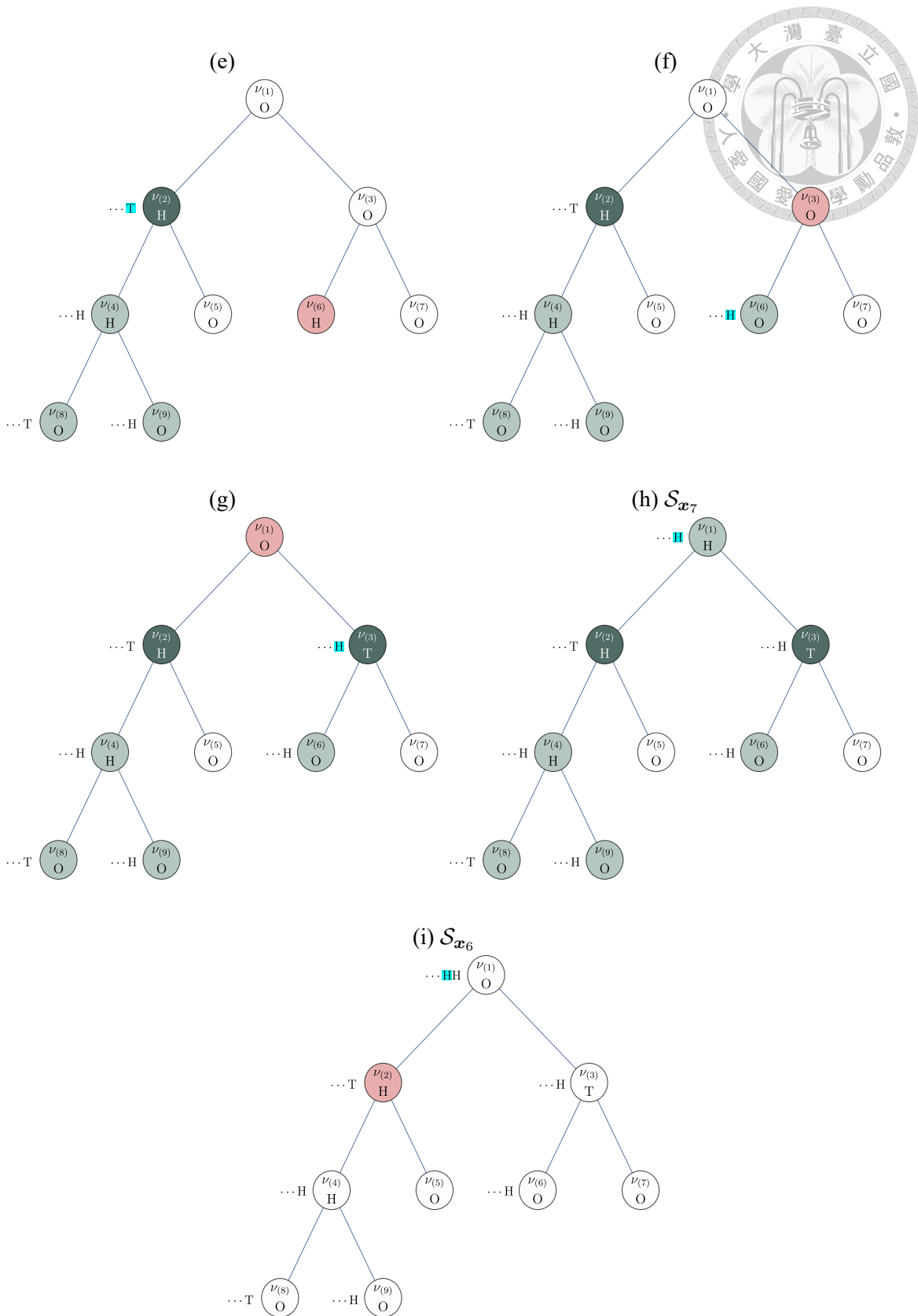


Figure 3.2: (contd.) An illustration of reconstructing x given \mathcal{S}_x and $\mathcal{A}_5(x) = 00001$.

We have found $x_8 = H$ and $x_7 = H$ as well as \mathcal{S}_{x_7} and \mathcal{S}_{x_6} , which is given in Figure

3.2(i). Given \mathcal{S}_{x_6} and $\mathcal{A}_S(x_6) = 000$, we can further identify x_6 and x_5 . In the end, we arrive at $x_8 = \text{TTHTHHH}$.



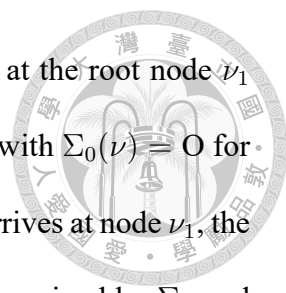
3.4 Zhou-Bruck's streaming version of Peres' algorithm

Peres [20] described and analyzed his algorithm in the fixed-to-variable length regime where the length of the input sequence is fixed. In order to generate a pre-specified number k of (unbiased) output bits, Zhou and Bruck [30] introduced a streaming version of Peres' algorithm (denoted by \mathcal{A}_{ZB}). Letting $\tau = \inf\{n : |\mathcal{A}_{\text{ZB}}(\mathbf{X}_n)| \geq k\}$, they showed that the first k bits in $\mathcal{A}_{\text{ZB}}(\mathbf{X}_\tau)$ are independent unbiased. While we do not consider the variable-to-fixed length regime in this dissertation, we will show that given $|\mathcal{A}_{\text{ZB}}(\mathbf{X}_n)| = \ell$, the conditional distribution of $\mathcal{A}_{\text{ZB}}(\mathbf{X}_n)$ is uniform on $\{0, 1\}^\ell$.

We first describe the $\{\text{H}, \text{T}, \text{O}, 0, 1\}$ -labeled status tree introduced in [30], which is different from our $\{\text{H}, \text{T}, \text{O}\}$ -labeled status tree discussed in Section 3.2. For $x_n = (x_1, \dots, x_n) \in \{\text{H}, \text{T}\}^n$, let

$$\sigma'(x_n) = \begin{cases} x_n, & \text{if } n \text{ is odd} \\ \text{O}, & \text{if } n = 0 \text{ or } n > 0 \text{ is even and } x_{n-1} = x_n \\ 0, & \text{if } n > 0 \text{ is even and } x_{n-1}x_n = \text{TH} \\ 1, & \text{if } n > 0 \text{ is even and } x_{n-1}x_n = \text{HT}. \end{cases}$$

Given an input sequence $x \in \{\text{H}, \text{T}\}^*$ (from the source), let $\mathbf{I}_x(\nu)$ be the input sequence at node ν derived from x . Then Zhou-Bruck's status tree Σ_x derived from x is the (infinite complete) binary tree T with the label $\Sigma(\nu) = \sigma'(\mathbf{I}_x(\nu))$ at each node ν . Note that \mathcal{S}_x can be derived from Σ_x by converting a label of 0 or 1 to O. We may describe dynamically



how Zhou-Bruck's status tree evolves as input bits (symbols) arrive at the root node ν_1 from the source. Initially at time 0, we have the trivial status tree Σ_0 with $\Sigma_0(\nu) = O$ for all ν . Given the status tree Σ_n at time n , when an input symbol x_{n+1} arrives at node ν_1 , the updated status tree Σ_{n+1} and the output bits induced by x_{n+1} are determined by Σ_n and x_{n+1} . The following procedure describes how the label of each node is updated. When an input symbol (H or T) arrives at ν_1 , each node ν may receive a symbol (H or T) from its parent node $\pi\nu$ ($\pi\nu$ is referred to as the source if $\nu = \nu_1$), and may send a symbol (H or T) to $\lambda\nu$ or $\rho\nu$. Meanwhile, an output bit (0 or 1) may be induced at ν . Specifically, let ν be a node with a label $s \in \{H, T, O, 0, 1\}$. Suppose it receives a symbol $\iota \in \{H, T\}$ from its parent node $\pi\nu$. We do the following operations on ν .

- (i) When $s = O$, set (update) $s = \iota$ (and send no symbol to $\lambda\nu$ or $\rho\nu$).
- (ii) When $s = 0$ or 1 , output s and set $s = \iota$ (and send no symbol to $\lambda\nu$ or $\rho\nu$).
- (iii) If $s\iota = HT$, set $s = 1$ and send a symbol H to $\lambda\nu$ (and send no symbol to $\rho\nu$).
- (iv) If $s\iota = TH$, set $s = 0$ and send a symbol H to $\lambda\nu$ (and send no symbol to $\rho\nu$).
- (v) If $s\iota = HH$, set $s = O$ and send a symbol T to $\lambda\nu$ and a symbol H to $\rho\nu$.
- (vi) If $s\iota = TT$, set $s = O$ and send a symbol T to $\lambda\nu$ and a symbol T to $\rho\nu$.

If a node receives no symbol from its parent node, then its label is unchanged and it sends no symbol to its child nodes. Thus, if a node receives no symbol from its parent node, then none of its descendant nodes receives a symbol, so that their labels remain unchanged. When the label of a node ν is updated from H or T to O or 0 or 1, it must have received a symbol from $\pi\nu$ and must send a symbol to $\lambda\nu$ and may also send a symbol to $\rho\nu$ if the current label and the received symbol are the same. As an example, let $x_8 = TTTHTHHH$

be the input sequence. Then Σ_{x_i} , $i = 1, \dots, 8$ are presented in Figure 3.3 where those nodes not shown have label O. Note that at time 4, we have $\Sigma_{x_4}(\nu_1) = \Sigma_{x_4}(\nu_{10}) = 0$, and these two 0's have yet to join the output sequence. At time 5, the label 0 at node ν_1 joins the output sequence. At time 6, the label 0 at node ν_{10} joins the output sequence, while $\Sigma_{x_6}(\nu_1) = 0$. The second 0 at ν_1 joins the output sequence at time 7. At time 8, we have $\Sigma_{x_8}(\nu_{10}) = 1$ and $\Sigma_{x_8}(\nu_{11}) = 0$, and these labels 1 and 0 have yet to join the output sequence. So we have $\mathcal{A}_{ZB}(x_8) = 000$.

Remark 3.7. To implement \mathcal{A}_{ZB} , a node ordering needs to be specified when two or more bits are simultaneously ready to join the output sequence. However, unlike \mathcal{A}_S which requires a particular node ordering to guarantee unbiasedness, we will show that any node ordering in implementation of \mathcal{A}_{ZB} yields independent and unbiased output bits. We may refer to \mathcal{A}_{ZB} as a *delayed* version of \mathcal{A}_S as a label of 0 or 1 at a node ν has to wait to join the output sequence until the node ν receives a symbol from its parent node. Zhou and Bruck [30] wrote that “the timing is crucial that we output a node’s label (when it is 1 or 0) only after it receives the next symbol from its parent node or from the source.” But they did not explain what may go wrong with “no delay”. Their approach may be related to an important observation of Blum [2] where in a more general setting, “no delay” results in biased output bits.

Remark 3.8. Before we establish the unbiasedness property of \mathcal{A}_{ZB} , we need to discuss the properties of Zhou-Bruck’s status tree. While a status tree Σ is the binary tree T with a node label $\Sigma(\nu) \in \{H, T, O, 0, 1\}$ for each node ν , not all such status trees are derived from an input sequence $x \in \{H, T\}^*$. A status tree is said to be admissible if it is derived from some input sequence $x \in \{H, T\}^*$. In what follows, we drop the word “admissible” so that a status tree always refers to an admissible status tree. The depth δ_Σ of a status tree

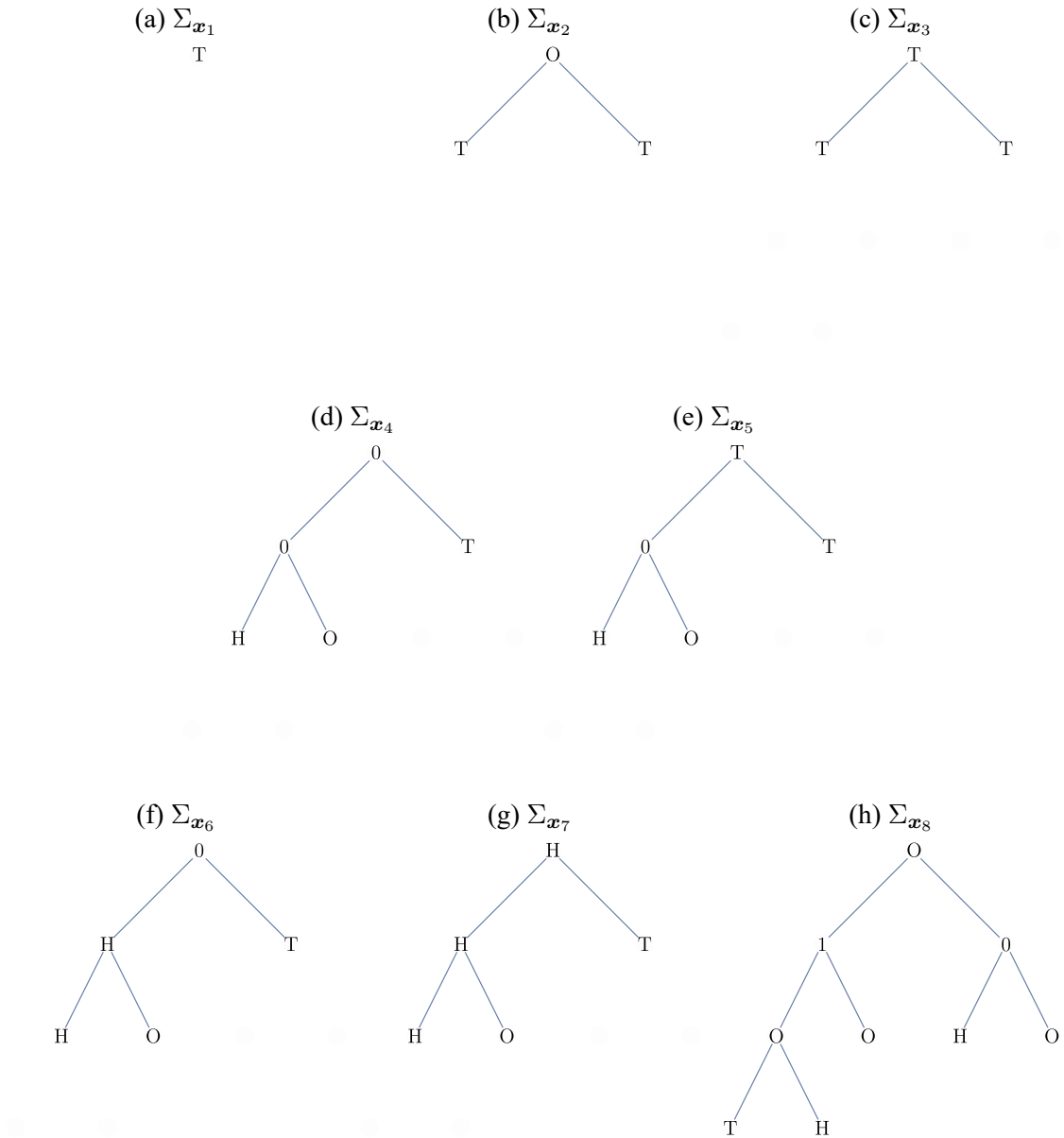


Figure 3.3: An example of Zhou-Bruck $\{H, T, O, 1, 0\}$ -labeled status trees Σ_{x_i} , $i = 1, \dots, 8$, where $x_8 = \text{TTHTTHHH}$.



Σ is defined by

$$\delta_{\Sigma} = \begin{cases} 0, & \text{if } \Sigma(\nu) = \text{O for all nodes } \nu, \\ 1 + \max\{m \geq 0 : \Sigma(\lambda^m \nu_1) = \text{H or T}\}, & \text{otherwise.} \end{cases}$$

Note that $\delta_{\Sigma_x} = \delta_{S_x}$ for $x \in \{\text{H}, \text{T}\}^*$.

Lemma 3.3. For an (admissible) status tree Σ , let $\mathcal{X}_{\Sigma} = \{x \in \{\text{H}, \text{T}\}^* : \Sigma_x = \Sigma\} \neq \emptyset$.

Then for $x, x' \in \mathcal{X}_{\Sigma}$, we have

$$|\mathbf{I}_x(\nu)|_{\text{H}} = |\mathbf{I}_{x'}(\nu)|_{\text{H}}, \quad |\mathbf{I}_x(\nu)|_{\text{T}} = |\mathbf{I}_{x'}(\nu)|_{\text{T}}, \quad |\mathcal{A}_{\nu\text{N}}(\mathbf{I}_x(\nu))| = |\mathcal{A}_{\nu\text{N}}(\mathbf{I}_{x'}(\nu))|,$$

for all nodes ν . Consequently, there are non-negative integers $n_{\Sigma}(\nu)$, $k_{\Sigma}(\nu)$, $\ell_{\Sigma}(\nu)$ such that for all $x \in \mathcal{X}_{\Sigma}$ and for all nodes ν ,

$$\begin{aligned} n_{\Sigma}(\nu) &= |\mathbf{I}_x(\nu)|, & k_{\Sigma} &= |\mathbf{I}_x(\nu)|_{\text{H}}, & n_{\Sigma}(\nu) - k_{\Sigma}(\nu) &= |\mathbf{I}_x(\nu)|_{\text{T}} \\ \ell_{\Sigma}(\nu) &= |\mathcal{A}_{\nu\text{N}}(\mathbf{I}_x(\nu))| - \mathbf{1}(\Sigma(\nu) = 0 \text{ or } 1) \end{aligned}$$

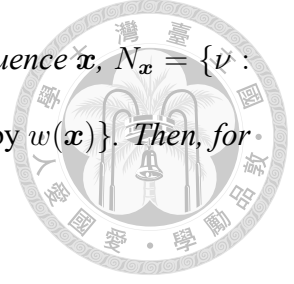
In particular, for $x \in \mathcal{X}_{\Sigma}$, $|x|$, $|x|_{\text{H}}$, $|x|_{\text{T}}$ and $\ell_{\Sigma} = \sum_{\nu} \ell_{\Sigma}(\nu) = \sum_{\nu} |\mathcal{A}_{\nu\text{N}}(\mathbf{I}_x(\nu))| - |\{\nu : \Sigma(\nu) = 0 \text{ or } 1\}|$ are Σ -properties.

Proof. Let S be a $\{\text{H}, \text{T}, \text{O}\}$ -labeled status tree given by

$$S(\nu) = \begin{cases} \Sigma(\nu) & \text{if } \Sigma(\nu) = \text{H, T or O,} \\ \text{O} & \text{if } \Sigma(\nu) = 0 \text{ or } 1. \end{cases} \quad (3.7)$$

Then we have $\mathcal{X}_{\Sigma} \subset \mathcal{X}_S$, which together with Proposition 3.1 implies Lemma 3.3. \square

Lemma 3.4. Let Σ be a status tree with the corresponding set $\mathcal{X}_{\Sigma} = \{x \in \{\text{H}, \text{T}\}^* : \Sigma_x =$



$\Sigma\} \neq \emptyset$. For $\mathbf{x} \in \mathcal{X}_\Sigma$, let $w_{\mathbf{x}} = w(\mathbf{x})$ denote the last term of the sequence \mathbf{x} , $N_{\mathbf{x}} = \{\nu : \nu \text{ is affected by } w(\mathbf{x})\}$, and $\tilde{N}_{\mathbf{x}} = \{\nu : \text{an output bit is induced at } \nu \text{ by } w(\mathbf{x})\}$. Then, for $\mathbf{x}', \mathbf{x}'' \in \mathcal{X}_\Sigma$, we have for all nodes ν ,

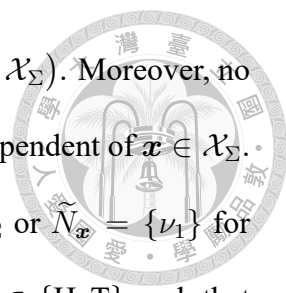
$$w(\mathbf{I}_{\mathbf{x}'}(\nu)) = w(\mathbf{I}_{\mathbf{x}''}(\nu)), \quad N_{\mathbf{x}'} = N_{\mathbf{x}''} \quad \text{and} \quad \tilde{N}_{\mathbf{x}'} = \tilde{N}_{\mathbf{x}''}.$$

In other words, there are $\theta(\nu) \in \{\mathbf{H}, \mathbf{T}\}$ for all nodes ν and two sets of nodes S and \tilde{S} such that for all $\mathbf{x} \in \mathcal{X}_\Sigma$ and for all nodes ν , $w(\mathbf{I}_{\mathbf{x}}(\nu)) = \theta(\nu)$, $N_{\mathbf{x}} = S$ and $\tilde{N}_{\mathbf{x}} = \tilde{S}$.

Remark 3.9. Let $n = |\mathbf{x}|$. By $w(\mathbf{x})$ inducing an output bit at ν , we mean that the $\{0, 1\}$ -valued label of ν joins the output sequence at time n . More precisely, with respect to \mathbf{x} , the label of ν at time $n - 1$ is 0 or 1. When $w(\mathbf{x})$ arrives at the root node ν_1 , ν receives a $\{\mathbf{H}, \mathbf{T}\}$ -valued symbol from its parent node which triggers the $\{0, 1\}$ -valued label of ν to join the output sequence at time n .

Proof of Lemma 3.4. The lemma holds trivially for Σ with $\delta_\Sigma = 0$ or 1 since $|\mathcal{X}_\Sigma| = 1$. We proceed by induction on δ_Σ . Suppose the lemma holds for Σ with $\delta_\Sigma \leq m - 1$ for some $m \geq 2$. Consider a status tree Σ with $\delta_\Sigma = m$. By the induction hypothesis applied to $\Sigma^{(\text{L})} := \Sigma|_{\Delta(\nu_{10})}$ and $\Sigma^{(\text{R})} := \Sigma|_{\Delta(\nu_{11})}$ (both of depth $\leq m - 1$), we have $w(\mathbf{I}_{\mathbf{x}}(\nu))$ independent of $\mathbf{x} \in \mathcal{X}_\Sigma$ for all $\nu \neq \nu_1$. (More precisely, for a descendant node ν of ν_{10} , the input sequence $\mathbf{I}_{\mathbf{x}}(\nu)$ at ν derived from the input sequence $\mathbf{x} \in \mathcal{X}_\Sigma$ at ν_1 may also be referred to as the input sequence at ν derived from the input sequence $\mathbf{I}_{\mathbf{x}}(\nu_{10})$ at the ‘‘root’’ node ν_{10} (with respect to $\Delta(\nu_{10})$). Then by the induction hypotheses applied to $\Delta(\nu_{10})$, we have $w(\mathbf{I}_{\mathbf{x}}(\nu))$ independent of $\mathbf{x} \in \mathcal{X}_\Sigma$.) It remains to show that $w(\mathbf{x}) = w(\mathbf{I}_{\mathbf{x}}(\nu_1))$, $N_{\mathbf{x}}$ and $\tilde{N}_{\mathbf{x}}$ are independent of $\mathbf{x} \in \mathcal{X}_\Sigma$.

By Lemma 3.3, let $n = n_\Sigma(\nu_1)$ be such that $|\mathbf{x}| = |\mathbf{I}_{\mathbf{x}}(\nu_1)| = n$ for all $\mathbf{x} \in \mathcal{X}_\Sigma$. If n is



odd, then $w(\mathbf{x}) = \Sigma(\nu_1)$ for $\mathbf{x} \in \mathcal{X}_\Sigma$ (i.e. $w(\mathbf{x})$ is independent of $\mathbf{x} \in \mathcal{X}_\Sigma$). Moreover, no node of level ≥ 2 is affected by $w(\mathbf{x})$, implying that $N_{\mathbf{x}} = \{\nu_1\}$, independent of $\mathbf{x} \in \mathcal{X}_\Sigma$. We have $\tilde{N}_{\mathbf{x}} \subset N_{\mathbf{x}} = \{\nu_1\}$. To show that $\tilde{N}_{\mathbf{x}} = \emptyset$ for all $\mathbf{x} \in \mathcal{X}_\Sigma$ or $\tilde{N}_{\mathbf{x}} = \{\nu_1\}$ for all $\mathbf{x} \in \mathcal{X}_\Sigma$, by the induction hypothesis applied to $\Sigma^{(L)}$, there is a $z \in \{H, T\}$ such that $z = w(\mathbf{x}^{(L)})$ for all $\mathbf{x}^{(L)} \in \mathcal{X}_{\Sigma^{(L)}}$. Noting that $\mathbf{I}_{\mathbf{x}}(\nu_{10}) \in \mathcal{X}_{\Sigma^{(L)}}$ for $\mathbf{x} \in \mathcal{X}_\Sigma$, we have $w(\mathbf{I}_{\mathbf{x}}(\nu_{10})) = z$ for $\mathbf{x} \in \mathcal{X}_\Sigma$. For $\mathbf{x} \in \mathcal{X}_\Sigma$, let \mathbf{x}_{-n} be \mathbf{x} with the last term x_n deleted. If $z = H$, then $w(\mathbf{I}_{\mathbf{x}}(\nu_{10})) = w(\mathbf{I}_{\mathbf{x}_{-n}}(\nu_{10})) = H$, implying that $\Sigma_{\mathbf{x}_{-n}}(\nu_1) = 0$ or 1 for $\mathbf{x} \in \mathcal{X}_\Sigma$, so $\tilde{N}_{\mathbf{x}} = \{\nu_1\}$, independent of $\mathbf{x} \in \mathcal{X}_\Sigma$. If $z = T$, then $w(\mathbf{I}_{\mathbf{x}}(\nu_{10})) = w(\mathbf{I}_{\mathbf{x}_{-n}}(\nu_{10})) = T$, implying that $\Sigma_{\mathbf{x}_{-n}}(\nu_1) = 0$ for $\mathbf{x} \in \mathcal{X}_\Sigma$, so $\tilde{N}_{\mathbf{x}} = \emptyset$, independent of $\mathbf{x} \in \mathcal{X}_\Sigma$.

Next suppose n is even. If $\Sigma(\nu_1) = 0$, then we have $w(\mathbf{x}) = H$, independent of $\mathbf{x} \in \mathcal{X}_\Sigma$. Moreover, for $\mathbf{x} \in \mathcal{X}_\Sigma$, when the symbol $w(\mathbf{x}) = H$ arrives at ν_1 , ν_1 sends a symbol H to ν_{10} and sends no symbol to ν_{11} , implying that $N_{\mathbf{x}}$ contains no nodes in $\Delta(\nu_{11})$ for all $\mathbf{x} \in \mathcal{X}_\Sigma$. (For notational convenience, we write $N_{\mathbf{x}} \cap \Delta(\nu_{11}) = \emptyset$ for $\mathbf{x} \in \mathcal{X}_\Sigma$.) By the induction hypothesis applied to $\Sigma^{(L)}$, $N_{\mathbf{x}} \cap \Delta(\nu_{10})$ and $\tilde{N}_{\mathbf{x}} \cap \Delta(\nu_{10})$ are independent of $\mathbf{x} \in \mathcal{X}_\Sigma$, where $N_{\mathbf{x}} \cap \Delta(\nu_{10})$ is the set of nodes in $\Delta(\nu_{10})$ that are affected by the input symbol H at ν_{10} , and $\tilde{N}_{\mathbf{x}} \cap \Delta(\nu_{10})$ is the set of nodes in $\Delta(\nu_{10})$ where an output bit is induced by the input symbol H at ν_{10} . Since $N_{\mathbf{x}} \cap \Delta(\nu_{11}) = \emptyset$ for $\mathbf{x} \in \mathcal{X}_\Sigma$ (implying that $\tilde{N}_{\mathbf{x}} \cap \Delta(\nu_{11}) = \emptyset$), we have $N_{\mathbf{x}} = \{\nu_1\} \cup (N_{\mathbf{x}} \cap \Delta(\nu_{10}))$ and $\tilde{N}_{\mathbf{x}} = \tilde{N}_{\mathbf{x}} \cap \Delta(\nu_{10})$ independent of $\mathbf{x} \in \mathcal{X}_\Sigma$. Similarly, if $\Sigma(\nu_1) = 1$, we have $w(\mathbf{x}) = T$, $N_{\mathbf{x}}$ and $\tilde{N}_{\mathbf{x}}$ independent of $\mathbf{x} \in \mathcal{X}_\Sigma$.

Finally, suppose $\Sigma(\nu_1) = 0$. By the induction hypothesis applied to $\Sigma^{(L)}$ and $\Sigma^{(R)}$, we have $z^{(L)} \in \{H, T\}$ and $z^{(R)} \in \{H, T\}$ such that $w(\mathbf{I}_{\mathbf{x}}(\nu_{10})) = z^{(L)}$ and $w(\mathbf{I}_{\mathbf{x}}(\nu_{11})) = z^{(R)}$ for $\mathbf{x} \in \mathcal{X}_\Sigma$. The fact that $\Sigma(\nu_1) = 0$ implies that $z^{(L)} = T$. Let $N^{(L)}$ ($N^{(R)}$, resp.)

be the set of nodes in $\Delta(\nu_{10})$ ($\Delta(\nu_{11})$, resp.) that are affected by $w(\mathbf{I}_x(\nu_{10})) = z^{(L)}$ ($w(\mathbf{I}_x(\nu_{11})) = z^{(R)}$, resp.). Let $\tilde{N}^{(L)}$ ($\tilde{N}^{(R)}$, resp.) be the set of nodes in $\Delta(\nu_{10})$ ($\Delta(\nu_{11})$, resp.) where an output bit is induced by $w(\mathbf{I}_x(\nu_{10})) = z^{(L)}$ ($w(\mathbf{I}_x(\nu_{11})) = z^{(R)}$, resp.). Note that by the induction hypothesis, the sets $N^{(L)}$, $N^{(R)}$, $\tilde{N}^{(L)}$ and $\tilde{N}^{(R)}$ are independent of $\mathbf{x} \in \mathcal{X}_\Sigma$. It follows that $N_x = \{\nu_1\} \cup N^{(L)} \cup N^{(R)}$ and $\tilde{N}_x = \tilde{N}^{(L)} \cup \tilde{N}^{(R)}$ are independent of $\mathbf{x} \in \mathcal{X}_\Sigma$. If $z^{(R)} = \text{H}$, then $w(\mathbf{x}) = \text{H}$, independent of $\mathbf{x} \in \mathcal{X}_\Sigma$. If $z^{(R)} = \text{T}$, then $w(\mathbf{x}) = \text{T}$, independent of $\mathbf{x} \in \mathcal{X}_\Sigma$. The proof is complete. \square

Lemma 3.5. *For a status tree Σ , the Σ -properties $n_\Sigma(\nu)$, $k_\Sigma(\nu)$, $\ell_\Sigma(\nu)$ satisfy the following conditions. For each node ν ,*

$$\ell_\Sigma(\nu) + \mathbf{1}(\Sigma(\nu) = 0 \text{ or } 1) = k_\Sigma(\lambda\nu), \quad n_\Sigma(\rho\nu) = n_\Sigma(\lambda\nu) - k_\Sigma(\lambda\nu)$$

$$k_\Sigma(\nu) = k_\Sigma(\lambda\nu) + 2k_\Sigma(\rho\nu) + \mathbf{1}(\Sigma(\nu) = \text{H}),$$

$$n_\Sigma(\nu) - k_\Sigma(\nu) = k_\Sigma(\lambda\nu) + 2(n_\Sigma(\rho\nu) - k_\Sigma(\rho\nu)) + \mathbf{1}(\Sigma(\nu) = \text{T}).$$

Proof. Let \mathcal{S} be a $\{\text{H}, \text{T}, \text{O}\}$ -labeled status tree as defined in (3.7). Noting that $\mathcal{X}_\Sigma \subset \mathcal{X}_{\mathcal{S}}$, Lemma 3.5 follows from Lemma 3.1. \square

Lemma 3.6. *Let Σ be a status tree and $\mathcal{X}_\Sigma = \{\mathbf{x} : \Sigma_{\mathbf{x}} = \Sigma\} \neq \emptyset$. Let $n(\nu) = n_\Sigma(\nu)$, $k(\nu) = k_\Sigma(\nu)$, $\ell = \ell_\Sigma$, $\ell(\nu) = \ell_\Sigma(\nu)$ be the Σ -properties such that $|\mathbf{I}_x(\nu)| = n(\nu)$, $|\mathbf{I}_x(\nu)|_{\text{H}} = k(\nu)$, $\ell(\nu) = |\mathcal{A}_{\text{vN}}(\mathbf{I}_x(\nu))| - \mathbf{1}(\Sigma(\nu) = 0 \text{ or } 1)$, and $\ell = \sum_{\nu} \ell(\nu) = |\mathcal{A}_{\text{P}}(\mathbf{x})| - |\{\nu : \Sigma(\nu) = 0 \text{ or } 1\}|$ for $\mathbf{x} \in \mathcal{X}_\Sigma$. Then $|\mathcal{X}_\Sigma| = 2^\ell$.*

Proof. The lemma holds trivially for Σ with $\delta_\Sigma = 0$ or 1 since $\ell = 0$ and $|\mathcal{X}_\Sigma| = 1$. We proceed by induction on δ_Σ . Suppose the lemma holds for Σ with $\delta_\Sigma \leq m$ for some $m \geq 1$. Consider a status tree Σ with $\delta_\Sigma = m + 1$. Let $\Sigma^{(L)} = \Sigma|_{\Delta(\nu_{10})}$ and $\Sigma^{(R)} = \Sigma|_{\Delta(\nu_{11})}$. Let $\ell^{(L)} = \sum_{\nu \in \Delta(\nu_{10})} \ell(\nu)$ and $\ell^{(R)} = \sum_{\nu \in \Delta(\nu_{11})} \ell(\nu)$. By the induction hypothesis applied to



$\Sigma^{(L)}$ and $\Sigma^{(R)}$, we have $|\mathcal{X}_{\Sigma^{(L)}}| = 2^{\ell^{(L)}}$ and $|\mathcal{X}_{\Sigma^{(R)}}| = 2^{\ell^{(R)}}$. By Lemma 3.5, for $\mathbf{x}^{(L)} \in \mathcal{X}_{\Sigma^{(L)}}$ and $\mathbf{x}^{(R)} \in \mathcal{X}_{\Sigma^{(R)}}$,

$$|\mathbf{x}^{(L)}|_H = k(\nu_{10}) = \ell(\nu_1) + \mathbf{1}(\Sigma(\nu_1) = 0 \text{ or } 1), \quad |\mathbf{x}^{(L)}|_T = n(\nu_{10}) - k(\nu_{10}) = n(\nu_{11}) = |\mathbf{x}^{(R)}|.$$

If $\ell(\nu_1) \geq 1$, then for each $\mathbf{y} \in \{0, 1\}^{\ell(\nu_1)}$, $(\mathbf{x}^{(L)}, \mathbf{x}^{(R)}, \mathbf{y})$ determines a unique $\mathbf{x} \in \mathcal{X}_{\Sigma}$ such that $\lambda(\mathbf{x}) = \mathbf{x}^{(L)}$, $\rho(\mathbf{x}) = \mathbf{x}^{(R)}$ and $\mathcal{A}_{\text{VN}}(\mathbf{x}) = \mathbf{y}$ if $\Sigma(\nu_1) \neq 0$ or 1 or $\mathcal{A}_{\text{VN}}(\mathbf{x}) = \mathbf{y} * \Sigma(\nu_1)$ if $\Sigma(\nu_1) = 0$ or 1 , where $*$ is concatenation. If $\ell(\nu_1) = 0$, then $(\mathbf{x}^{(L)}, \mathbf{x}^{(R)})$ determines a unique $\mathbf{x} \in \mathcal{X}_{\Sigma}$ such that $\lambda(\mathbf{x}) = \mathbf{x}^{(L)}$, $\rho(\mathbf{x}) = \mathbf{x}^{(R)}$ and $\mathcal{A}_{\text{VN}}(\mathbf{x}) = \emptyset$ if $\Sigma(\nu_1) \neq 0$ or 1 or $\mathcal{A}_{\text{VN}}(\mathbf{x}) = \Sigma(\nu_1)$ if $\Sigma(\nu_1) = 0$ or 1 . Indeed, the mapping $f : \mathcal{X}_{\Sigma} \rightarrow \mathcal{X}_{\Sigma^{(L)}} \times \mathcal{X}_{\Sigma^{(R)}} \times \{0, 1\}^{\ell(\nu_1)}$ with $f(\mathbf{x}) = (\lambda(\mathbf{x}), \rho(\mathbf{x}), \mathcal{A}_{\text{VN}}^*(\mathbf{x}))$ is 1-1 and onto where $\mathcal{A}_{\text{VN}}^*(\mathbf{x})$ is $\mathcal{A}_{\text{VN}}(\mathbf{x})$ if $\Sigma(\nu_1) \neq 0$ or 1 or $\mathcal{A}_{\text{VN}}(\mathbf{x})$ with the last bit deleted if $\Sigma(\nu_1) = 0$ or 1 .

This shows that

$$\begin{aligned} |\mathcal{X}_{\Sigma}| &= |\mathcal{X}_{\Sigma^{(L)}}| \times |\mathcal{X}_{\Sigma^{(R)}}| \times 2^{\ell(\nu_1)} \\ &= 2^{\ell^{(L)}} \times 2^{\ell^{(R)}} \times 2^{\ell(\nu_1)} = 2^{\ell}. \end{aligned}$$

The proof is complete. □

Proposition 3.4. *Let $\mathcal{A}_{\text{ZB}}(\mathbf{x})$ denote the output sequence generated by \mathcal{A}_{ZB} applied to \mathbf{x} according to any given node ordering. Let Σ be a status tree with $\ell = \ell_{\Sigma} \geq 1$. Then given $\Sigma_{\mathbf{X}} = \Sigma$, the conditional distribution of $\mathcal{A}_{\text{ZB}}(\mathbf{X})$ is uniform on $\{0, 1\}^{\ell}$. Consequently, given $|\mathcal{A}_{\text{ZB}}(\mathbf{X})| = \ell$, the conditional distribution of $\mathcal{A}_{\text{ZB}}(\mathbf{X})$ is uniform on $\{0, 1\}^{\ell}$.*

Proof. To establish that $\mathcal{A}_{\text{ZB}}|_{\mathcal{X}_{\Sigma}} : \mathcal{X}_{\Sigma} \rightarrow \{0, 1\}^{\ell}$ is 1-1 and onto, by Lemma 3.6, it suffices



to show that

$$\text{if } \mathcal{A}_{\text{ZB}}(\mathbf{x}') = \mathcal{A}_{\text{ZB}}(\mathbf{x}'') \text{ for } \mathbf{x}', \mathbf{x}'' \in \mathcal{X}_{\Sigma}, \text{ then } \mathbf{x}' = \mathbf{x}''. \quad (3.8)$$

Let n_{Σ} be such that $|\mathbf{x}| = n_{\Sigma}$ for $\mathbf{x} \in \mathcal{X}_{\Sigma}$. If $n_{\Sigma} = 1$, (3.8) holds trivially since $|\mathcal{X}_{\Sigma}| = 1$.

We proceed by induction on n_{Σ} .

Suppose (3.8) holds for Σ with $n_{\Sigma} \leq m - 1$ for some $m \geq 2$. Consider a status tree Σ with $n_{\Sigma} = m$ and $\ell = \ell_{\Sigma} \geq 1$. Let $\mathbf{x}', \mathbf{x}'' \in \mathcal{X}_{\Sigma}$ be such that $\mathcal{A}_{\text{ZB}}(\mathbf{x}') = \mathcal{A}_{\text{ZB}}(\mathbf{x}'')$. We need to show that $\mathbf{x}' = \mathbf{x}''$. By Lemma 3.4, $w(\mathbf{x}') = w(\mathbf{x}'')$, $N_{\mathbf{x}'} = N_{\mathbf{x}''}$ and $\tilde{N}_{\mathbf{x}'} = \tilde{N}_{\mathbf{x}''}$. Let $\mathbf{y} = y_1 \cdots y_{\ell} = \mathcal{A}_{\text{ZB}}(\mathbf{x}') = \mathcal{A}_{\text{ZB}}(\mathbf{x}'') \in \{0, 1\}^{\ell}$. Let $\gamma = |\tilde{N}_{\mathbf{x}'}| (= |\tilde{N}_{\mathbf{x}''}|)$. Since $\tilde{N}_{\mathbf{x}'} = \tilde{N}_{\mathbf{x}''}$, we have $\mathcal{A}_{\text{ZB}}(\mathbf{x}'_{-m}) = \mathcal{A}_{\text{ZB}}(\mathbf{x}''_{-m}) = y_1 \cdots y_{\ell-\gamma}$, where \mathbf{x}'_{-m} and \mathbf{x}''_{-m} are, respectively, \mathbf{x}' and \mathbf{x}'' with the last term deleted. We claim that $\Sigma_{\mathbf{x}'_{-m}} = \Sigma_{\mathbf{x}''_{-m}}$. To prove the claim, we have $\Sigma_{\mathbf{x}'_{-m}}(\nu) = \Sigma_{\mathbf{x}''_{-m}}(\nu) = \Sigma(\nu)$ for $\nu \notin N_{\mathbf{x}'} (= N_{\mathbf{x}''})$. For $\nu \in \tilde{N}_{\mathbf{x}'} (= \tilde{N}_{\mathbf{x}''})$, $\Sigma_{\mathbf{x}'_{-m}}(\nu)$ and $\Sigma_{\mathbf{x}''_{-m}}(\nu)$ are the (same) output bit at ν (induced by $w(\mathbf{x}')$ and $w(\mathbf{x}'')$). For $\nu \in N_{\mathbf{x}'} \setminus \tilde{N}_{\mathbf{x}'} (= N_{\mathbf{x}''} \setminus \tilde{N}_{\mathbf{x}''})$, we have

$$\Sigma_{\mathbf{x}'_{-m}}(\nu) = \Sigma_{\mathbf{x}''_{-m}}(\nu) = \begin{cases} \text{O} & \text{if } \Sigma(\nu) = \text{H or T}, \\ w(\mathbf{I}_{\mathbf{x}}(\rho\nu)) & \text{if } \Sigma(\nu) = \text{O}, \\ \text{H} & \text{if } \Sigma(\nu) = 1, \\ \text{T} & \text{if } \Sigma(\nu) = 0. \end{cases}$$

(Note that for $\nu \in N_{\mathbf{x}'} \setminus \tilde{N}_{\mathbf{x}'}$, $\Sigma_{\mathbf{x}'_{-m}}(\nu) \neq 0$ or 1 . Note also that $w(\mathbf{I}_{\mathbf{x}}(\rho\nu))$ is independent of $\mathbf{x} \in \mathcal{X}_{\Sigma}$ by Lemma 3.4.) This established the claim that $\Sigma_{\mathbf{x}'_{-m}} = \Sigma_{\mathbf{x}''_{-m}}$.

By the induction hypothesis applied to the status tree of $\Sigma_{\mathbf{x}'_{-m}} = \Sigma_{\mathbf{x}''_{-m}}$ together with $\mathcal{A}_{\text{ZB}}(\mathbf{x}'_{-m}) = \mathcal{A}_{\text{ZB}}(\mathbf{x}''_{-m})$, we have $\mathbf{x}'_{-m} = \mathbf{x}''_{-m}$, implying that $\mathbf{x}' = \mathbf{x}''$ (since

$w(\mathbf{x}') = w(\mathbf{x}'')$ by Lemma 3.4). The proof is complete.



3.5 Counting status trees

For $\mathbf{X}_n = (X_1, \dots, X_n) \in \{H, T\}^n$, an iid sequence with $\mathbb{P}(X_i = H) = p$, to compute $\mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|$, it is useful to find $a_\ell^{(n,k)} = |\{\mathcal{S} : n_S = n, k_S = k, \ell_S = \ell\}|$ for integers $k \leq n, \ell \leq n - 1$, where n_S, k_S and ℓ_S are the \mathcal{S} -properties such that $|\mathbf{x}| = n_S, |\mathbf{x}|_H = k_S$ and $|\mathcal{A}_P(\mathbf{x})| = \ell_S$ for $\mathbf{x} \in \mathcal{X}_S$. By Proposition 3.2, we have $|\mathcal{X}_S| = 2^{\ell_S}$. We write $\mathbf{x} \sim \mathbf{x}'$ if $\mathcal{S}_\mathbf{x} = \mathcal{S}_{\mathbf{x}'}$, which is an equivalence relation. The set $S_{n,k} = \{\mathbf{x} \in \{H, T\}^n : |\mathbf{x}|_H = k, |\mathbf{x}|_T = n - k\}$ is then partitioned into equivalence classes each of which corresponds to a status tree. The equivalence class corresponding to a status trees \mathcal{S} is \mathcal{X}_S whose cardinality is 2^{ℓ_S} . Given $a_\ell^{(n,k)} = |\{\mathcal{S} : n_S = n, k_S = k, \ell_S = \ell\}|$, the set $S_{n,k}$ is partitioned into $a_\ell^{(n,k)}$ classes of cardinality $2^\ell, \ell = 0, 1, \dots$. Consequently,

$$\binom{n}{k} = \sum_{\ell \geq 0} a_\ell^{(n,k)} 2^\ell.$$

As an example, for $n = 10, k = 4$, we have

$$a_1^{(10,4)} = 1, \quad a_2^{(10,4)} = 2, \quad a_3^{(10,4)} = 1, \quad a_4^{(10,4)} = 4, \quad a_5^{(10,4)} = 2, \quad a_6^{(10,4)} = 1.$$

and $a_\ell^{(10,4)} = 0$ for $\ell \notin \{1, \dots, 6\}$. It is easily verified that

$$\sum_{\ell \geq 0} a_\ell^{(10,4)} 2^\ell = 210 = \binom{10}{4}.$$



Given $a_\ell^{(n,k)}$'s, $\mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|$ can be calculated by

$$\mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)| = \sum_{k=0}^n \sum_{\ell \geq 0} \ell a_\ell^{(n,k)} 2^\ell p^k q^{n-k}.$$

To derive a recursive formula for $a_\ell^{(n,k)}$'s, we need the following lemma.

Lemma 3.7. *For a status tree \mathcal{S} , we have*

$$|\mathbf{x}| - |\mathcal{A}_P(\mathbf{x})| = |\{\nu : \mathcal{S}(\nu) \neq \mathbf{O}\}|$$

for $\mathbf{x} \in \mathcal{X}_\mathcal{S}$.

Proof. The lemma holds trivially for \mathcal{S} with $\delta_\mathcal{S} = 0$ and $\delta_\mathcal{S} = 1$. We proceed by induction on $\delta_\mathcal{S}$. Suppose the lemma holds for \mathcal{S} with $\delta_\mathcal{S} \leq m-1$ for some $m \geq 2$. Consider a status tree \mathcal{S} with $\delta_\mathcal{S} = m$. Let $\mathcal{S}^{(L)} = \mathcal{S}|_{\Delta(\nu_{10})}$ and $\mathcal{S}^{(R)} = \mathcal{S}|_{\Delta(\nu_{11})}$, both of depth $\leq m-1$. For $\mathbf{x} \in \mathcal{X}_\mathcal{S}$, we have

$$|\mathbf{x}| = 2|\mathbf{I}_\mathbf{x}(\nu_{10})| + \mathbf{1}(\mathcal{S}(\nu_1) \neq \mathbf{O}), \quad (3.9)$$

$$|\mathbf{I}_\mathbf{x}(\nu_{10})| = |\mathbf{I}_\mathbf{x}(\nu_{11})| + |\mathcal{A}_{\nu N}(\mathbf{x})|. \quad (3.10)$$

By the induction hypothesis applied to $\mathcal{S}^{(L)}$ and $\mathcal{S}^{(R)}$, we have

$$|\mathbf{I}_\mathbf{x}(\nu_{10})| - |\mathcal{A}_P(\mathbf{I}_\mathbf{x}(\nu_{10}))| = |\{\nu : \mathcal{S}^{(L)}(\nu) \neq \mathbf{O}\}|, \quad (3.11)$$

$$|\mathbf{I}_\mathbf{x}(\nu_{11})| - |\mathcal{A}_P(\mathbf{I}_\mathbf{x}(\nu_{11}))| = |\{\nu : \mathcal{S}^{(R)}(\nu) \neq \mathbf{O}\}|. \quad (3.12)$$

Adding the equations (3.9)–(3.12) yields

$$|\mathbf{x}| - |\mathcal{A}_P(\mathbf{I}_\mathbf{x}(\nu_{10}))| - |\mathcal{A}_P(\mathbf{I}_\mathbf{x}(\nu_{11}))| = |\{\nu : \mathcal{S}(\nu) \neq \mathbf{O}\}| + |\mathcal{A}_{\nu N}(\mathbf{x})|,$$

implying that

$$|\mathbf{x}| - |\mathcal{A}_P(\mathbf{x})| = |\{\nu : \mathcal{S}(\nu) \neq \mathbf{O}\}|.$$

The proof is complete. □



By Lemma 3.7, a status tree \mathcal{S} satisfies

$$n_{\mathcal{S}} - \ell_{\mathcal{S}} = |\{\nu : \mathcal{S}(\nu) \neq \mathbf{O}\}|.$$

It is convenient to introduce the infinite-dimensional vector

$$D_{n,k} = (a_n^{(n,k)}, a_{n-1}^{(n,k)}, \dots, a_1^{(n,k)}, a_0^{(n,k)}, 0, 0, \dots).$$

In other words, the i -th element of $D_{n,k}$ ($i \leq n + 1$) is the number of status trees \mathcal{S} with $n_{\mathcal{S}} = n$, $k_{\mathcal{S}} = k$ and $|\{\nu : \mathcal{S}(\nu) \neq \mathbf{O}\}| = i - 1$ (implying that $\ell_{\mathcal{S}} = n_{\mathcal{S}} - i + 1$). Let \mathbb{N}^{∞} be the set of infinite-dimensional vectors of non-negative integers with finitely many non-zero elements. For $D = (d_1, d_2, \dots)$ and $D' = (d'_1, d'_2, \dots) \in \mathbb{N}^{\infty}$, define

$$D + D' = (d_1 + d'_1, d_2 + d'_2, \dots)$$

and

$$\begin{aligned} DD' &= \left(\sum_i d_i \mathbf{e}_i \right) \left(\sum_j d'_j \mathbf{e}_j \right) \\ &= \sum_{i,j} d_i d'_j \mathbf{e}_i \mathbf{e}_j \\ &= \sum_{i,j} d_i d'_j \mathbf{e}_{i+j-1}, \end{aligned}$$

where $\mathbf{e}_i = (0, \dots, 0, 1, 0, \dots)$ is the infinite-dimensional vector with the i th element 1



and all the other elements zeros.

Lemma 3.8. *We have $D_{n,k} = D_{n,n-k}$.*

Proof. For $\mathbf{x} = (x_1, \dots, x_n) \in \{\mathbf{H}, \mathbf{T}\}^n$, let $\tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_n) \in \{\mathbf{H}, \mathbf{T}\}^n$ be such that $\tilde{x}_i \neq x_i, i = 1, \dots, n$. We have $\mathbf{x} \in S_{n,k}$ if and only if $\tilde{\mathbf{x}} \in S_{n,n-k}$. It is readily seen that $\mathcal{S}_{\mathbf{x}}$ and $\mathcal{S}_{\tilde{\mathbf{x}}}$ satisfy that $\mathcal{S}_{\mathbf{x}}(\nu) = \mathcal{S}_{\tilde{\mathbf{x}}}(\nu)$ for $\nu \neq \nu_{(2^i-1)}, i = 1, 2, \dots$ and that for $\nu = \nu_{(2^i-1)}$, either $\mathcal{S}_{\mathbf{x}}(\nu) = \mathcal{S}_{\tilde{\mathbf{x}}}(\nu) = \mathbf{O}$ or $\mathcal{S}_{\mathbf{x}}(\nu), \mathcal{S}_{\tilde{\mathbf{x}}}(\nu) \in \{\mathbf{H}, \mathbf{T}\}$ and $\mathcal{S}_{\mathbf{x}}(\nu) \neq \mathcal{S}_{\tilde{\mathbf{x}}}(\nu)$. In particular, $|\{\nu : \mathcal{S}_{\mathbf{x}}(\nu) \neq \mathbf{O}\}| = |\{\nu : \mathcal{S}_{\tilde{\mathbf{x}}}(\nu) \neq \mathbf{O}\}|$. It follows from Lemma 3.7 that $a_{\ell}^{(n,k)} = a_{\ell}^{(n,n-k)}$. The proof is complete. \square

Note that $D_{0,0} = (1, 0, 0, \dots) = \mathbf{e}_1$ and $D_{1,0} = D_{1,1} = (0, 1, 0, \dots) = \mathbf{e}_2$.

Proposition 3.5. *Let $n \geq 2$ and $k \geq 1$. Then*

(i) *for even $n \geq 2$ and $k \leq n/2$,*

$$D_{n,k} = \sum_{i=0}^{\lfloor k/2 \rfloor} D_{\frac{n}{2}, k-2i} D_{\frac{n}{2}-k+2i, i},$$

(ii) *for odd $n \geq 3$ and $k < n/2$,*

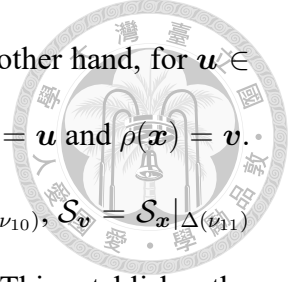
$$D_{n,k} = D_{1,0}(D_{n-1,k} + D_{n-1,k-1}).$$

Proof. To prove (i), let $n \geq 2$ be even and $0 \leq k \leq n/2$. For $\mathbf{x} \in S_{n,k}$, we have

$$\lambda(\mathbf{x}) \in S_{\frac{n}{2}, k-2i} \quad \text{and} \quad \rho(\mathbf{x}) \in S_{\frac{n}{2}-k+2i, i} \quad \text{for some } 0 \leq i \leq \lfloor k/2 \rfloor.$$

Moreover, $\mathcal{S}_{\mathbf{x}}(\nu_1) = \mathbf{O}$,

$$\mathcal{S}_{\mathbf{x}}|_{\Delta(\nu_{10})} = \mathcal{S}_{\lambda(\mathbf{x})} \quad \text{and} \quad \mathcal{S}_{\mathbf{x}}|_{\Delta(\nu_{11})} = \mathcal{S}_{\rho(\mathbf{x})},$$



implying that \mathcal{S}_x is uniquely determined by $\mathcal{S}_{\lambda(x)}$ and $\mathcal{S}_{\rho(x)}$. On the other hand, for $\mathbf{u} \in S_{\frac{n}{2}, k-2i}$ and $\mathbf{v} \in S_{\frac{n}{2}-k+2i, i}$, there is a unique $\mathbf{x} \in S_{n,k}$ such that $\lambda(\mathbf{x}) = \mathbf{u}$ and $\rho(\mathbf{x}) = \mathbf{v}$. Furthermore, the three status trees $\mathcal{S}_x, \mathcal{S}_u$ and \mathcal{S}_v satisfy $\mathcal{S}_u = \mathcal{S}_x|_{\Delta(\nu_{10})}, \mathcal{S}_v = \mathcal{S}_x|_{\Delta(\nu_{11})}$ and $|\{\nu : \mathcal{S}_x(\nu) \neq \mathbf{O}\}| = |\{\nu : \mathcal{S}_u(\nu) \neq \mathbf{O}\}| + |\{\nu : \mathcal{S}_v(\nu) \neq \mathbf{O}\}|$. This establishes the formula in (i). To prove (ii), note that

$$S_{n,k} = \{\mathbf{x} * \mathbf{T} : \mathbf{x} \in S_{n-1,k}\} \cup \{\mathbf{x} * \mathbf{H} : \mathbf{x} \in S_{n-1,k-1}\}$$

A status tree \mathcal{S}_x with $\mathbf{x} \in S_{n,k}$ and $\mathcal{S}_x(\nu_1) = \mathbf{T}$ induces the status tree $\mathcal{S}_{x_{-n}}$ with $\mathcal{S}_{x_{-n}}(\nu_1) = \mathbf{O}$ and $\mathcal{S}_{x_{-n}}(\nu) = \mathcal{S}_x(\nu)$ for $\nu \neq \nu_1$, where $x_{-n} \in S_{n-1,k}$ is \mathbf{x} with the last term deleted. Similarly, a status tree \mathcal{S}_x with $\mathbf{x} \in S_{n,k}$ and $\mathcal{S}_x(\nu_1) = \mathbf{H}$ induces the status tree $\mathcal{S}_{x_{-n}}$ with $\mathcal{S}_{x_{-n}}(\nu_1) = \mathbf{O}$ and $\mathcal{S}_{x_{-n}}(\nu) = \mathcal{S}_x(\nu)$ for $\nu \neq \nu_1$, where $x_{-n} \in S_{n-1,k-1}$. Conversely, a status tree \mathcal{S}_x with $\mathbf{x} \in S_{n-1,k}$ induces the status tree $\mathcal{S}_{x*\mathbf{T}}$ with $\mathbf{x} * \mathbf{T} \in S_{n,k}$ and $\mathcal{S}_{x*\mathbf{T}}(\nu_1) = \mathbf{T}$ and $\mathcal{S}_{x*\mathbf{T}}(\nu) = \mathcal{S}_x(\nu)$ for $\nu \neq \nu_1$. A status tree \mathcal{S}_x with $\mathbf{x} \in S_{n-1,k-1}$ induces the status tree $\mathcal{S}_{x*\mathbf{H}}$ with $\mathbf{x} * \mathbf{H} \in S_{n,k}$ and $\mathcal{S}_{x*\mathbf{H}}(\nu_1) = \mathbf{H}$ and $\mathcal{S}_{x*\mathbf{H}}(\nu) = \mathcal{S}_x(\nu)$ for $\nu \neq \nu_1$. This establishes the formula in (ii). The proof is complete. \square





Chapter 4 Concluding remarks

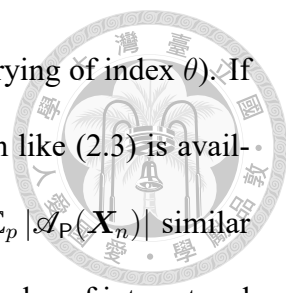
In this dissertation, we have studied the asymptotic behavior of Peres' algorithm and introduced and analyzed its streaming versions. Specifically, by exploiting the recursion in (2.3), we derived for $p = \frac{1}{2}$

$$\lim_{n \rightarrow \infty} \frac{\log(nh(p) - \mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|)}{\log n} = \theta = \log\left(\frac{1 + \sqrt{5}}{2}\right), \quad (4.1)$$

where $\mathbf{X}_n = (X_1, \dots, X_n)$ is the input sequence of the outcomes of n coin tosses with bias p , $|\mathcal{A}_P(\mathbf{X}_n)|$ is the number of unbiased output bits generated by Peres' algorithm \mathcal{A}_P applied to \mathbf{X}_n , and $h(p) = -p \log p - (1 - p) \log(1 - p)$ is the Shannon entropy of each X_i . For $p = \frac{1}{2}$, the coin is unbiased and the input sequence consists of n iid unbiased observations, so that $nh(p) - \mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)| = n - \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|$ may be referred to as the cost incurred by \mathcal{A}_P when not knowing $p = \frac{1}{2}$. It is a challenging task to obtain more refined results beyond (4.1). A positive sequence $\{L(n)\}$ is said to be *regularly varying* of index θ if

$$\lim_{n \rightarrow \infty} L(\lfloor \alpha n \rfloor) / L(n) = \alpha^\theta \quad \text{for all } \alpha > 0.$$

(See Bojanic and Seneta [3] for a unified theory of regularly varying sequences.) Figures 2.3 and 2.4 suggest that $n - \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|$ may be a regularly varying sequence. Furthermore, it is of interest to see if $(n - \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|) / n^\theta$ converges to a positive



constant (which would imply that $n - \mathbb{E}_{1/2} |\mathcal{A}_P(\mathbf{X}_n)|$ is regularly varying of index θ). If so, how can this constant be characterized? For $p \neq \frac{1}{2}$, no recursion like (2.3) is available. It seems difficult to derive an asymptotic result on $nh(p) - \mathbb{E}_p |\mathcal{A}_P(\mathbf{X}_n)|$ similar to (4.1). Furthermore, $\text{Var}_p(|\mathcal{A}_P(\mathbf{X}_n)|)$, the variance of $|\mathcal{A}_P(\mathbf{X}_n)|$, is also of interest and importance. Even for $p = \frac{1}{2}$, it seems challenging to derive the asymptotic behavior of $\text{Var}_{1/2}(|\mathcal{A}_P(\mathbf{X}_n)|)$ as $n \rightarrow \infty$.

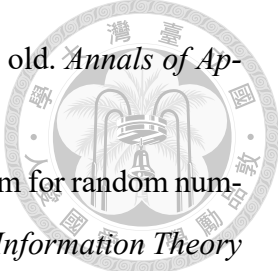
The original Peres' algorithm \mathcal{A}_P is not streaming in the sense that some of the output bits in $\mathcal{A}_P(\mathbf{X}_n)$ may be placed after the output bits induced by X_{n+1} . We introduced a binary tree representation of \mathcal{A}_P , based on which we further introduced a class of streaming versions of \mathcal{A}_P in terms of orderings of the nodes of the binary tree. We showed by example that in general a streaming version of \mathcal{A}_P is not unbiased. By establishing some useful properties of status trees, we showed that a particular streaming version of \mathcal{A}_P (denoted \mathcal{A}_S) is unbiased. We also showed that a delayed version of \mathcal{A}_S proposed by Zhou and Bruck [30] is unbiased.

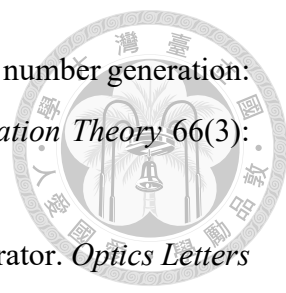
The algorithms considered in this dissertation are in the fixed-to-variable length regime where the length of the input sequence is fixed while the number of output bits is random. In practical applications, the variable-to-fixed length regime may be more relevant where a (fixed) number of (unbiased) output bits is required and the length of the input sequence is random, in which case a stopping time is involved (i.e. stopping at the first time when the number of output bits meets the requirement). Variants of Peres' algorithm in the variable-to-fixed length regime are worth further investigation.



References

1. Bernard, J. & Letac, G. (1973). Construction d'événements équiprobables et coefficients multinomiaux modulo p^n . *Illinois Journal of Mathematics* 17(2): 317–332.
2. Blum, M. (1986). Independent unbiased coin flips from a correlated biased source—a finite state Markov chain. *Combinatorica* 6(2): 97–108.
3. Bojanic, R. & Seneta, E. (1973). A unified theory of regularly varying sequences. *Mathematische Zeitschrift* 134(2): 91–106.
4. Dwass, M. (1972). Unbiased coin tossing with discrete random variables. *Annals of Mathematical Statistics* 43(3): 860–864.
5. Elias, P. (1972). The efficient construction of an unbiased random sequence. *Annals of Mathematical Statistics* 43(3): 865–870.
6. Han, T. S. & Hoshi, M. (1997). Interval algorithm for random number generation. *IEEE Transactions on Information Theory* 43(2): 599–611.
7. Hoeffding, W. & Simons, G. (1970). Unbiased coin tossing with a biased coin. *Annals of Mathematical Statistics* 41(2): 341–352.
8. Jacquet, P. & Szpankowski, W. (1999). Entropy computations via analytic depoissonization. *IEEE Transactions on Information Theory* 45(4): 1072–1081.
9. Kallenberg, O. (2002). *Foundations of Modern Probability*. New York: Springer.
10. Keane, M. S. & O'Brien, G. L. (1994). A Bernoulli factory. *ACM Transactions on Modeling and Computer Simulation* 4(2): 213–219.
11. Knuth, D. & Yao, A. (1976). The complexity of nonuniform random number generation. *Algorithms and Complexity: New Directions and Recent Results*. Ed. by J. Traub. Academic Press.
12. Lim, Z. G., Liao, C.-T., & Yao, Y.-C. (2020). Asymptotic analysis of Peres' algorithm for random number generation. *Probability in the Engineering and Informational Sciences*, to appear.
13. Lim, Z. G. & Yao, Y.-C. (2020). Some streaming versions of Peres' algorithm for random number generation. Unpublished manuscript.

- 
14. Nacu, Ş. & Peres, Y. (2005). Fast simulation of new coins from old. *Annals of Applied Probability* 15 (1A): 93–115.
15. Oohama, Y. (2011). Performance analysis of the interval algorithm for random number generation based on number systems. *IEEE Transactions on Information Theory* 57(3): 1177–1185.
16. Pae, S.-i. (2013). Exact output rate of Peres’s algorithm for random number generation. *Information Processing Letters* 113(5): 160–164.
17. Pae, S.-i. (2015). A generalization of Peres’s algorithm for generating random bits from loaded dice. *IEEE Transactions on Information Theory* 61(2): 751–757.
18. Pae, S.-i. (2020). Binarization trees and random number generation. *IEEE Transactions on Information Theory* 66(4): 2581–2587.
19. Pae, S.-i. & Loui, M. C. (2006). Randomizing functions: simulation of a discrete probability distribution using a source of unknown distribution. *IEEE Transactions on Information Theory* 52(11): 4965–4976.
20. Peres, Y. (1992). Iterating von Neumann’s procedure for extracting random bits. *The Annals of Statistics* 20(1): 590–597.
21. Prasitsupparote, A., Konno, N., & Shikata, J. (2018). Numerical and non-asymptotic analysis of Elias’s and Peres’s extractors with finite input sequences. *Entropy* 20(10): article #729, 19 pages.
22. Ryabko, B. & Matchikina, E. (2000). Fast and efficient construction of an unbiased random sequence. *IEEE Transactions on Information Theory* 46(3): 1090–1093.
23. Samuelson, P. A. (1968). Constructing an unbiased random sequence. *Journal of the American Statistical Association* 63(324): 1526–1527.
24. Seroussi, G. & Weinberger, M. J. (2015). Optimal algorithms for universal random number generation from finite memory sources. *IEEE Transactions on Information Theory* 61(3): 1277–1297.
25. Stout, Q. F. & Warren, B. (1984). Tree algorithms for unbiased coin tossing with a biased coin. *Annals of Probability* 12(1): 212–222.
26. von Neumann, J. (1951). Various techniques used in connection with random digits. *National Bureau of Standards Applied Math Series* 12: 36–38.

- 
27. Watanabe, S. & Han, T. S. (2020). Interval algorithm for random number generation: information spectrum approach. *IEEE Transactions on Information Theory* 66(3): 1691–1701.
28. Wei, W. & Guo, H. (2009). Bias-free true random-number generator. *Optics Letters* 34(12): 1876–1878.
29. Zhou, H. & Bruck, J. (2012a). Efficient generation of random bits from finite state Markov chains. *IEEE Transactions on Information Theory* 58(4): 2490–2506.
30. Zhou, H. & Bruck, J. (2012b). Streaming algorithms for optimal generation of random bits. arXiv: [1209.0730 \[cs, math\]](https://arxiv.org/abs/1209.0730). URL: <http://arxiv.org/abs/1209.0730>.