Journal of The Korea Society of Computer and Information
Vol. 18, No. 2, February 2013
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2013.18.2.019

무작위수생성을 위한 부 페레즈 함수

배 성 일*

The Sub-Peres Functions for Random Number Generation

Sung-il Pae*

요 약

이 논문은 무작위수 생성을 위한 페레즈 함수와 같이 재귀적으로 정의된 부 페레즈 함수에 대하여 논한다. 페레즈 함수와 같이 두 개의 인자함수를 쓰는 대신, 부 페레즈 함수들은 한 개의 인자함수만을 이용하여 정의된다. 따라서 당연히 그 출력효율은 페레즈 함수에 비하여 낮고, 점근적으로 최적효율을 내지도 않는다. 그러나, $O(n\log n)$의 시간복잡도를 갖는 페레즈 함수에 비하여, 부 페레즈 함수들은 선형시간, 즉 $O(n)$의 시간에 실행된다. 더구나, 이 함수들은 하나의 인자함수를 쓰기 때문만이 아니라 꼬리재귀함수로서 간단한 반복수행에 의해 구현되어 페레즈 함수보다 더 적은 메모리로 구현될 수 있다. 그럼에도, 이 함수들은 널리 알려진 선형시간 알고리즘인 폰 노이만 방법보다 출력효율이 최대 두 배 이상 높다. 따라서, 이 방법들은 모바일 기기와 같은 제한된 계산 자원을 가진 환경에서 폰 노이만 방법 대신 이용될 수 있다. 이 논문에서는 이러한 부 페레즈 함수들의 실행시간과 정확한 출력효율을 분석하여, 페레즈 함수를 비롯한 다른 무작위수 생성을 위한 방법들과 비교한다. 그리고, 부 페레즈 함수들의 구현에 대하여 논한다.

▸ Keywords : 무작위수 생성, 무작위화 함수, 페레즈 함수, 부 페레즈 함수, 폰 노이만 방법

Abstract

We study sub-Peres functions that are defined recursively as Peres function for random number generation. Instead of using two parameter functions as in Peres function, the sub-Peres functions uses only one parameter function. Naturally, these functions produce less random bits, hence are not asymptotically optimal. However, the sub-Peres functions runs in linear time, i.e., in $O(n)$ time rather than $O(n\log n)$ as in Peres's case. Moreover, the implementation is even simpler

• 제1저자 : 배성일
• 투고일 : 2013. 1. 29, 심사일 : 2013. 2. 9, 게재확정일 : 2013. 2. 23.
* 홍익대학교 컴퓨터공학과(Dept. of Computer Engineering, Hongik University)
• 이 논문은 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2009-0077288).

than Peres function not only because they use only one parameter function but because they are tail recursive, hence run in a simple iterative manner rather than by a recursion, eliminating the usage of stack and thus further reducing the memory requirement of Peres's method. And yet, the output rate of the sub-Peres function is more than twice as much as that of von Neumann's method which is widely known linear-time method. So, these methods can be used, instead of von Neumann's method, in an environment with limited computational resources like mobile devices. We report the analyses of the sub-Peres functions regarding their running time and the exact output rates in comparison with Peres function and other known methods for random number generation. Also, we discuss how these sub-Peres function can be implemented.

▸ Keywords : random number generation, randomizing function, Peres function, sub-Peres function, von Neumann method

I. 서 론

1. 폰 노이만의 방법

앞뒤가 완벽하게 같은 이상적인(ideal) 동전이 있다고 가정하자. 이 동전을 던졌을 때, 직관적으로 예상할 수 있는 결과는 아마도 앞면이 나올 확률과 뒷면이 나올 확률이 같다는 것일 것이다. 2004년, Diaconis 등은 이와 같은 예상에 반하게도, 앞면을 위로 해서 던졌다면 그 결과로 앞면이 나올 확률이 0.51보다 크다는 사실을 동전던지기의 역학적(mechanical) 과정을 연구하여 수학적으로 엄밀하게 증명하였다[1, 2]. 이 사실은 진성무작위수 공급원의 통계적 성질을 제어하기 힘들다는 사실을 웅변적으로 말해주는 예이다.

편향이 있는 무작위수를 보정하는 가장 오래되고 유명한 방법은 폰 노이만(von Neumann)에 의해 제안된 다음의 방법이다[3]. 앞면이 나올 사건을 H라 하고 그 확률이 p이고, 뒷면이 나올 사건을 T 라 하고 확률이 $q=1-p$인 동전을 생각하자. 이때 동전을 두 번 던져서 생기는 확률적 사건들은, 앞면이 두 번 연속으로 나오는 HH, 앞면과 뒷면이 이어 나오는 HT, 뒷면과앞면이 이어 나오는 TH 그리고 뒷면이 두번 연속으로 나오는 TT로 네 가지가 있다. 이 중 HH나 TT의 경우 이 사건들을 무시하고 다시 동전을 두번 던지고, HT인 경우 이를 0으로 간주하고, TH가 나오는 경우 1로 간주하자. 이를 사상으로 표현하면, 출력이 없는 경우를 λ로 표현할 때,

$$HH\mapsto\lambda,\ HT\mapsto 0,\ TH\mapsto 1,\ TT\mapsto\lambda,$$

로 나타낼 수 있다. 그러면

$$\Pr[0\text{이 나올 사건}]=\Pr[HT]=pq,$$
$$\Pr[1\text{이 나올 사건}]=\Pr[TH]=pq$$

이 되어, 위 방법이 0을 출력할 확률과 1을 출력할 확률이 pq로 같음을 알 수 있다. 따라서 주어진 동전의 편향 p의 값에 상관없이 이 방법에 의해 출력되는 0과 1은 항상 같은 확률로 나타난다.

2. 진성무작위수와 편향의 교정

무작위수는 그 생성(generation)의 관점에서 크게 유사무작위수(pseudorandom number)와 진성무작위수(true random number)로 나눈다. 유사무작위수는 정해진 알고리즘에 의해 생성되어, 그 통계적 성질이 미리 잘 연구되어져서 그 쓰임새에 적합하도록 설계할 수 있고, 또한 컴퓨터의 속도만큼 빠르게 대량으로 생성할 수 있고, 무엇보다도 재생성(reproduce)할 수 있다는 것이 장점이다. 반면, 진성무작위수는 생성자가 제어할 수 없는 공급원(source)으로부터 만들어 진다. 예컨대, 우주공간으로 부터 감지되는 방사선, 전기저항의 열잡음(thermal noise), 그리고 동전이나 주사위 던지기도 진성무작위수의 공급원으로 볼 수 있다.

진성무작위수는 재생성이 가능하지 않고, 유사무작위수에 비해 대량으로 생성하기는 쉽지 않다. 그럼에도 진성무작위수를 반드시 필요로 하는 응용분야가 있는데, 그 중 대표적인 것이 암호이다. 예를 들어 암호화를 위한 키생성에 무작위수를 필요로 하는데, 유사무작위수의 예측가능성은 암호시스

템에 치명적이다.

진성무작위수는 그 생성법의 특성상 통계적 성질을 제어하기 쉽지 않다. 예를 들면, 데스크탑 컴퓨터에 장착할 수 있게 만들어진 인텔의 무작위수 생성기는 전기저항의 온도에 따른 미시적 가변성에 기반하여 0과 1의 열(sequence)을 생성하는데, 그 비율이 같지 않아서 보정을 해 주어야 한다[4]. 또한 무작위수의 균등성(uniformity)은 암호의 보안성에도 중요한데, 만약 암호에 쓰이는 무작위수가 편향되어(biased) 있다면, 이 성질이 그 암호체계를 공격하는데 쓰일 수 있다. 매우 간단하고 오래된 방법이지만, 폰 노이만의 방법은 실제로 인텔의 무작위수 생성기를 비롯한 여러 진성무작위수 생성에서 그 공급원의 편향성을 보정하기 위해 널리 쓰인다[4]. 보편화 되어가는 모바일 계산환경에서도 암호는 여러 측면에서 중요하게 쓰이고, 따라서 계산자원이 제한된 모바일 환경에서 작동하는 간단하고 효율적인 진성무작위수의 생성은 중요하다. 이때, 주목할 측면은 시간복잡도(time complexity)와 공간복잡도(space complexity), 그리고 주어진 공급원에 대해 얼마나 많은 무작위수를 생성하는가 하는 출력효율에 관한 측면인데 아래에서는 이러한 측면에서 폰 노이만의 방법과 또 다른 방법인 페레즈의 방법에 대해 논한다.

3. 페레즈의 방법

진성무작위수의 생성에서 출력효율(output rate)은 공급원의 입력에 대한 평균출력의 비율이다. 예컨대, 앞에 예를 든 폰 노이만 방법에서 동전 던지기의 횟수에 대하여 얻게 되는 평균 출력 비트의 비율이다. 실제로 진성무작위수의 공급원이 제한된 자원일 수 있고, 또한 많은 출력을 필요로 하는 경우 출력효율은 진성무작위수 생성법 성능의 중요한 잣대가 된다. 폰 노이만의 방법을 이용하여 편향이 p인 동전을 보정하는 경우, 하나의 출력을 얻기 위해 평균적으로 $1/pq$번의 동전던지기를 해야 함을 쉽게 보일 수 있다.

$$\frac{1}{pq}=\frac{1}{p(1-p)}\geq 4$$

이므로, 하나의 출력을 얻기 위해 적어도 평균 4번의 동전던지기를 해야 함을 알 수 있다. 이는 편향 p와 무관하게 성립하고, 편향 p가 0.5에서 멀어질 수록 더 많은 수의 동전을 던져야 한다. 이 사실을 출력효율로 바꾸어 말하면, 한 번의 동전던지기에 대해 얻게 되는 평균 비트수는 0.25보다 편향과 상관없이 적다.

편향된 무작위수 보정법의 효율성은 그 공급원의 엔트로피에 의해 제한되는데, 이를 엔트로피 한계치(entropy bound)라 부르자. 여기서 엔트로피는 클로드 샤논(Claude Shannon)에 의해 정의된 엔트로피를 말한다 [5, 6]. 편향 p인 동전의 경우 그 엔트로피는 $-p\log_2 p - q\log_2 q$이고, 어떤 보정법의 출력효율도 이 엔트로피보다 작다는 것이 알려져 있다. 만약 주어진 동전의 편향 p가 0.3이라면, 이에 대한 엔트로피는 대략 0.881이고, 이 경우 폰 노이만 방법의 출력효율은 $0.3 \cdot 0.7 = 0.21$로서 엔트로피 한계치보다 네 배 이상 나쁨을 알 수 있다.

그러므로 폰노이만 방법보다 더 효율적인 방법이 있는가 하는 질문이 자연스럽게 생긴다. 여러 사람들이 이 질문에 대한 답으로서, 폰 노이만의 방법보다 더 효율적인 방법을 제시하였는데, 실제로 효율성이 엔트로피 한계치에 임의로 가까와지는 보정법들이 존재한다[7, 8]. 다만 이 방법들은 계산적으로 더 복잡하고, 효율성이 엔트로피 한계치에 가까운 방법일수록 한 번에 많은 수의 동전을 던져야 한다. 예를 들어, 표 1에 의해 정해지는 함수 Ψ_2는 한 번에 네 번의 동전을 입력으로 받아들여서 최대 두 개의 0 또는 1을 출력한다. 이 경우 출력효율은

$$1 \cdot 2p^2q^2 + 2 \cdot 4p^2q^2 + 2 \cdot 4p^3q + 2 \cdot 4qp^3$$

인데, 대략 폰 노이만의 방법보다 두 배 가량 더 크다.

주어진 입력의 편향성(bias)과 상관없이 출력의 확률분포가 일정하게 되는 함수를 무작위화함수(randomizing function)이라 부르자[9, 10]. 위에서 언급한 보정법들은, 함수로 볼 때, 모두 무작위화 함수들이다. 무작위화 함수의 조합론적 성질을 연구하면 출력효율이 최적인(optimal) 무작위화함수를 찾을 수 있다. 이에 의하면 폰 노이만의 방법은 입력을 한 번에 두 개의 입력을 받아들이는 무작위화 함수 중 최적임을 알 수 있다. (실제로는, 폰 노이만의 방법이, 0과 1을 바꾸어서 출력하는 경우를 같은 방법으로 간주할 때, 한 번에 두 개의 입력을 받아들이는 유일한 무작위화 함수이다.) 따라서 더 효율적인 보정을 위해서는 한 번에 받아들이는 입력의 길이가 길어져야 함을 알 수 있다. 그리고 위에서 제시한 Ψ_2는 한 번에 입력을 4개 받아들이는 무작위화 함수중 최적임을 증명할 수 있다.

표 1. 페레즈 함수 Ψ_2
Table 1. Peres Function Ψ_2

x	$\Psi_2(x)$	x	$\Psi_2(x)$
HHHH	λ	HHHT	11
TTTT	λ	HHTH	01
HHTT	1	HTHH	10
HTHT	11	THHH	00
HTTH	10	HTTT	10
THHT	01	THTT	00
THTH	00	TTHT	11
TTHH	0	TTTH	01

페레즈 함수는 출력효율이 엔트로피 한계치에 다가가는, 알려진 점근적 최적 무작위화 함수중 계산복잡도가 가장 낮은 함수로서 입력길이 n에 대하여 $O(n\log n)$의 시간복잡도를 갖는다. 페레즈함수는 시간복잡도가 낮을 뿐 아니라, 구현이 간단하고 적은 메모리를 필요로 한다는 장점이 있는데, 이 사실의 주된 이유는 아주 간단한 재귀식에 의해 정의된다는 데 있다. 이 논문에서는 페레즈 함수와 비슷한 방식으로 재귀적으로 정의하여, 부 페레즈(sub-Peres) 함수라고 부르는 무작위화 함수에 대하여 논한다.

페레즈 함수와 같이 두 개의 인자함수를 쓰는 대신, 부 페레즈 함수들은 한 개의 인자함수만을 이용하여 정의된다. 따라서 당연히 그 출력효율은 페레즈 함수에 비하여 낮고, 점근적으로 최적효율을 내지도 않는다. 그러나, 자매논문인 [11]에서 논의한 바와 같이, 무작위수 생성에서의 출력효율과 시간복잡도 사이의 트레이드오프(tradeoff)가 페레즈함수와 부 페레즈 함수 사이에도 성립하는데, $O(n\log n)$의 시간복잡도를 갖는 페레즈 함수에 비하여, 이 부 페레즈 함수들은 선형시간, 즉 $O(n)$의 시간에 실행된다. 더구나, 이 함수들은 꼬리재귀함수(tail recursive function)로서 간단한 반복수행에 의해 구현되어 이미 적은 메모리를 요구하는 페레즈 함수보다 더 적은 메모리로 구현 될 수 있다. 그럼에도, 이 함수들은 널리 알려진 선형시간 알고리즘인 폰 노이만 방법보다 출력효율이 최대 두 배 이상 높다. 따라서, 이 방법들은 모바일 기기와 같은 제한된 계산 자원을 가진 환경에서 폰 노이만 방법 대신 이용될 수 있다. 이 논문에서는 이러한 부 페레즈 함수들의 실행시간과 정확한 출력효율을 분석하고 그 구현에 대하여 논한다.

II장에서는 무작위화 함수에 대한 이론과 페레즈 함수의 정의, 출력효율, 시간복잡도등을 개략적으로 설명한다. III장에서는 부 페레즈 함수를 정의하고, 시간복잡도와 출력효율을 분석하여 페레즈 함수와 다른 무작위화 함수에 관련한 주요 지표와 비교한다. 특히 출력효율을 분석하는 방법론에 대하여 자세히 다룬다. IV장에서는 위의 분석결과를 바탕으로, 부 페레즈 함수의 응용에 대하여 논한다.

II. 배경 및 관련 연구

1. 무작위화함수

개별시행이 독립이고 각 시행에서의 편향이 동일한 무작위수 공급원을 베르누이 공급원(Bernoulli source)이라 한다. 이 논문에서는 항상 베르누이 공급원을 가정하고, 이진수열 $x\in\{0,1\}^n$는 베르누이 공급원으로부터 n개의 비트를 독립적으로 뽑은 것으로 가정한다. 그리고 따로 정하지 않는 한, 우리가 논의하는 베르누이 공급원의 편향은 0이 나오는 확률 p라고 쓰고, 1이 나오는 확률을 $q=1-p$로 정하기로 하자.

무작위화 함수 $f:\{0,1\}^n\rightarrow\{0,1\}^*$는 베르누이 공급원으로부터 길이 n인 입력을 받아서, 그 함수값으로서 각 비트가 편향없고 독립인 이진수열(binary sequence)을 출력한다. 정확한 정의를 위해 x는 입력, $z=f(x)$로서 함수 f의 출력이라 하자. 이진수열 z에 대해 $z[l]$은 z의 l번째 비트를 나타내고, $z[1,l]$은 처음 l개의 비트로 이루어지는 이진수열을 뜻한다. 그러면 무작위화 함수는 다음과 같이 정의할 수 있다.

정의 1 [7, 10] 함수 $f:\{0,1\}^n\rightarrow\{0,1\}^*$는, 각 l과 길이 $l-1$인 이진수열 w에 대하여, 그리고 모든 편향값 p에 대하여, 다음의 조건을 만족할 때 무작위화 함수라 부른다.

$$\Pr[z[l]=0\,|\,z[1,l-1]=w] = \Pr[z[l]=1\,|\,z[1,l-1]=w].$$

특히, 입력의 길이 n을 강조할 필요가 있을 때, n-무작위화 함수라고 부른다.

길이가 n인 이진수열의 집합 $\{0,1\}^n$의 원소들 중 1의 개수가 k인 것들의 부분집합을 $S_{n,k}$라 하면, $\{0,1\}^n$는 $n+1$개의 같은 확률을 가진 이진수열들의 집합으로 다음과 같이 나누어 진다:

$$\{0,1\}^n = S_{n,0}\cup S_{n,1}\cup\cdots\cup S_{n,n}$$

이제 $S_{n,k}$의 크기, $\binom{n}{k}$의 이진수전개를 생각하자:

$$|S_{n,k}| = \binom{n}{k} = 2^{n_1} + 2^{n_2} + \cdots + 2^{n_l}$$

이고, 이때 $n_1 > n_2 > \cdots > n_l \geq 0$ 이다. 이와 같이 주어졌을 때, 다음과 같이 함수 $A(n,k)$를 정의하자.

$$A(n,k) = n_1 2^{n_1} + n_2 2^{n_2} + \cdots + n_i 2^{n_l}.$$

그러면, n-무작위화 함수의 최대출력효율에 대한 다음의 사실이 성립한다.

정리 1 [9, 10] 입력의 길이가 n인 무작위화 함수의 최대효율은 $\frac{1}{n}\sum_{k=0}^{n} A(n,k)p^{n-k}q^k$이다.

위에 주어진 최대효율을 만족하는 함수는 일라이어스[7]에 의해 처음 제안 되었고, [10]에서 최적 n-무작위화 함수임이 증명되었다.

우리가 이 논문에서 다루게 되는 무작위화 함수는 각 출력길이 d에 대해 $\{0,1\}^d$의 모든 원소에 대해 같은 확률의 출력을 갖는데, 이런 무작위화 함수를 추출함수라 부르고, 다음과 같이 정의한다.

정의 2 [8, 10] 함수 $f:\{0,1\}^n \to \{0,1\}^*$가, 모든 편향 p에 대하여, 길이가 같은 이진수열 z_1과 z_2에 대하여

$$\Pr[f(x) = z_1] = \Pr[f(x) = z_2]$$

이면, f를 추출함수라 부른다.

함수 f가 추출적이라는 것은, 동확률 부분집합 $S_{n,k}$에서 각 출력길이 d에 대해 $\{0,1\}^d$의 모든 원소를 같은 횟수로 출력한다는 것과 동치조건이다. 추출함수는 무작위 함수이고, 우리가 논할 페레즈 함수와 부 페레즈 함수는 추출함수이다.

2. 페레즈 함수

2.1 페레즈 함수의 정의

페레즈 함수 Ψ_ν는 폰노이만 함수와 두 가지 특별한 인자함수 u와 v를 이용하여 재귀적으로 정의한 무작위화 함수인데, 다음과 같이 정의된다.

$$\Psi_\nu(x) = \Psi_1(x) * \Psi_{\nu-1}(u(x)) * \Psi_{\nu-1}(v(x))$$

여기서 *는 연결(concatenation) 연산이고, Ψ_1은 폰노이만 함수이다. 함수 u와 v는 다음과 같이 정의된다.

$$u(00) = u(11) = 0;\ u(01) = u(10) = 1;$$
$$v(00) = 0;\ v(11) = 1;\ v(01) = v(10) = \lambda.$$

함수 u는 XOR(exclusive-or) 연산이고, 함수 v는 폰노이만 함수의 보함수(complement)로 볼 수 있다. 이 세 개의 함수는 모두 길이가 2인 입력에 대하여 정의되어 있으나, 다음과 같은 식으로 임의의 이진수열에 대하여 확장할 수 있다. 우선 길이가 짝수인 경우,

$$\Psi_1(x_1x_2\cdots x_{2n}) = \Psi_1(x_1x_2) * \cdots * \Psi_1(x_{2n-1}x_{2n})$$

로 정의하고, 입력의 길이가 홀수인 경우에는 마지막 비트를 버리고 남은 입력을 취한다. 함수 u와 v도 비슷한 식으로 확장한다. 그러면 위의 페레즈함수는 임의의 길이의 입력에 대하여 정의 된다.

본질적으로 위의 정의와 같으나 재귀호출의 깊이가 제한되지 않는 다음의 함수를 생각하자.

$$\Psi(x) = \Psi_1(x) * \Psi(u(x)) * \Psi(v(x))$$

이 함수 역시 Ψ_ν와 같이 모든 입력길이에 대하여 정의되고 역시 페레즈 함수라고 부른다. 재귀호출의 깊이가 제한되지 않기 때문에 출력효율은 임의의 Ψ_ν보다 더 크다.

페레즈 함수가 무작위화 함수임을 논하기 위해, 우선 입력의 길이가 $2n$인 경우를 생각하자. 모든 입력중 1의 개수가 k인 입력을 모아 놓은 집합 $S_{2n,k}$는 폰노이만 함수에 대한 출

력의 길이가 (1) k가 홀수인 경우, 1부터 k까지 사이의 홀수들이 나올 수 있고, (2) k가 짝수인 경우, 0부터 k까지 사이의 짝수들이 나올 수 있다. 다음의 부분집합

$$C_l = \{x \in S_{2n,k} | |\Psi_1(x)| = l\}$$

을 생각하면, $S_{2n,k}$를 다시 폰노이만 함수의 출력길이에 따라 다음과 같이 나눌 수 있다.

$$S_{2n,k} = \begin{cases} C_0 \cup C_2 \cup \cdots \cup C_k & k\text{가 짝수일 때,} \\ C_1 \cup C_3 \cup \cdots \cup C_k & k\text{가 홀수일 때.} \end{cases} \quad \text{(P)}$$

이제 부분집합 C_l은 다음과 같은 중요한 성질을 가진다.

보조정리 S [12] 매핑 $\Phi : x \mapsto (\Psi_1(x), u(x), v(x))$는 C_l과 $\{0,1\}^l \times S_{n,l} \times S_{n-1,(k-l)/2}$사이의 일대일 대응이다.

$$C_l = \{x \in S_{2n,k} | |\Psi_1(x)| = l\}$$

이 정리의 결과로서, x가 베르누이 공급원의 변수일 때, $\Psi_1(x)$, $u(x)$, $v(x)$가 서로 확률적으로 독립이라는 사실 때문에 페레즈 함수는 무작위화 함수이다. 자세한 증명은 [12]를 참조한다.

따름정리 R [8, 12] 페레즈 함수 Ψ, 그리고 $\nu \geq 1$인 자연수에 대하여 Ψ_ν는 추출함수이고, 따라서 무작위화 함수이다.

2.2 페레즈 함수의 출력효율과 시간복잡도

보조정리 S는 페레즈 함수의 조합론적인 구조를 밝힘으로써, 페레즈 함수가 무작위화 함수임을 보이는데 중요한 역할을 할 뿐 아니라 그 정확한 출력효율을 계산할 수 있도록 해준다. 자세한 계산은 [12]을 참조하고, 편향 p에 대한 함수 Ψ의 출력효율 $r(p)$는 이 논문의 주제인 부 페레즈 함수의 출력효율과 함께 그림 1에서 볼 수 있다.

정확한 출력효율 대신, 각각의 ν에 대하여 이 함수의 점근적 효율

$$r_\nu(p) = \lim_{n \to \infty} \mathrm{E}(|\Psi_\nu(x)|)/n$$

이다. $r_\nu(p)$는 다음과 같은 재귀식을 만족한다.

$$r_\nu(p) = r_1(p) + \frac{1}{2} r_{\nu-1}(p^2+q^2) + \frac{1}{2}(p^2+q^2) r_{\nu-1}(p^2/(p^2+q^2)).$$

주어진 편향 p에 대하여 이 재귀식을 풀면 입력이 무한히 커질 때 ν번째 페레즈함수의 효율의 극한이 되고, 이 값은 다시 재귀호출의 깊이에 해당하는 ν가 무한대로 다가갈 때 엔트로피 한계치에 다가간다[9]. 그러나 정해진 ν와 유한인 입력의 길이에 대하여는 그 효율이 최적이 아님을 알 수 있다.

함수 Ψ의 시간복잡도를 생각하자. 주어진 길이 n의 입력에 대하여 함수 Ψ_1, u, v는 각각 $O(n)$의 시간에 계산이 가능함을 쉽게 알 수 있다. 그리고 $u(x), v(x)$의 길이는 각각 $n/2$보다 크지 않기 때문에 다음 레벨의 재귀호출의 입력의 길이는 기껏해야 $n/2$가 된다. 따라서 페레즈 함수 Ψ 의 시간복잡도를 $T(n)$이라 하면 $T(n)$은 다음의 식을 만족한다.

$$T(n) = O(n) + T(n/2) + T(n/2)$$

따라서 $T(n) = O(n \log n)$임을 알 수 있다. 이 시간복잡도는 페레즈 함수 Ψ_ν에도 그대로 적용된다.

III. 부 페레즈(Sub-Peres) 함수

1. 정의

다음과 같이 정의되는 두 함수를 생각하자:

$$\Psi'(x) = \Psi_1(x) * \Psi'(u(x)),$$
$$\Psi''(x) = \Psi_1(x) * \Psi''(v(x)).$$

이 함수들은 페레즈 함수의 인자함수를 하나씩만 이용하여 역시 재귀적으로 정의되는 함수들이다. 페레즈 함수와 같은 식으로 짝수 길이의 이진수열에서 정의하고, 홀수 길이의 이진수열에 대해서는 마지막 비트를 무시함으로써 임의의 길이의 이진 수열에 대하여 정의된다. 이 함수들이 무작위화 함수가 됨은 페레즈 함수의 경우와 같이, 베르누이 공급원의 입력변수 x에 대하여, $\Psi_1(x)$, $u(x)$, $v(x)$가 확률적으로 독립이

라는 사실(보조정리 S)로부터 도출된다.

정리 R′ 부 페레즈 함수 Ψ' 와 Ψ'' 는 무작위화 함수이다.

페레즈 함수와 비교할 때, 이 함수들의 출력효율은 페레즈 함수보다 낮으리라는 것을 쉽게 알 수 있다. 다른 흥미로운 측면은 이 함수들의 계산복잡도가 페레즈 함수와는 달리 선형 시간복잡도를 가진다는 사실이다.

2. 시간복잡도

입력길이 n에 대한 함수 Ψ'의 시간복잡도를 $A(n)$이라 하자. 그러면 Ψ_1과 u는 각각 선형시간에 실행되므로 길이 n인 입력 x에 대하여 $\Psi'(x)$를 계산하기 위해 $\Psi_1(x)$와 $u(x)$를 계산하기 위한 실행시간은 $O(n)+O(n)=O(n)$이다. 그리고 $|u(x)|=n/2$이므로, 재귀호출 $\Psi'(u(x))$를 계산하기 위한 시간은 $A(n/2)$가 된다. 따라서, 다음의 식을 얻는다:

$$A(n)=O(n)+A(n/2).$$

이 식에서 우변을 계속 전개해 나가면,

$$\begin{aligned} A(n) &= O(n)+O(n/2)+A(n/4) \\ &= O(n)+O(n/2)+O(n/4)+\cdots \\ &= O(n). \end{aligned}$$

따라서 Ψ'의 시간복잡도는 입력길이 n에 대하여 선형으로 증가함을 알 수 있고, 함수 Ψ''의 시간복잡도 역시 비슷한 식으로, v가 선형시간에 계산되고 $|v(x)|\le n/2$임을 이용하여, 선형 시간복잡도를 가짐을 보일 수 있다.

3. 출력효율과 점근적 효율

3.1 입력길이에 대한 정확한 출력효율

위에 정의한 두 함수 Ψ'와 Ψ''의 출력효율은 페레즈 함수의 경우 [12], 또는 하이브리드 무작위화 함수의 경우 [11]와 비슷한 방식으로 계산할 수 있다. 다만 그 계산을 위한 재귀식이 약간 다르고, 다음과 같이 계산할 수 있다. 입력길이 $2n$에 대하여 동확률 부분집합 $S_{2n,k}$에서 함수 Ψ'의 총출력길이를 다음과 쓰기로 하자.

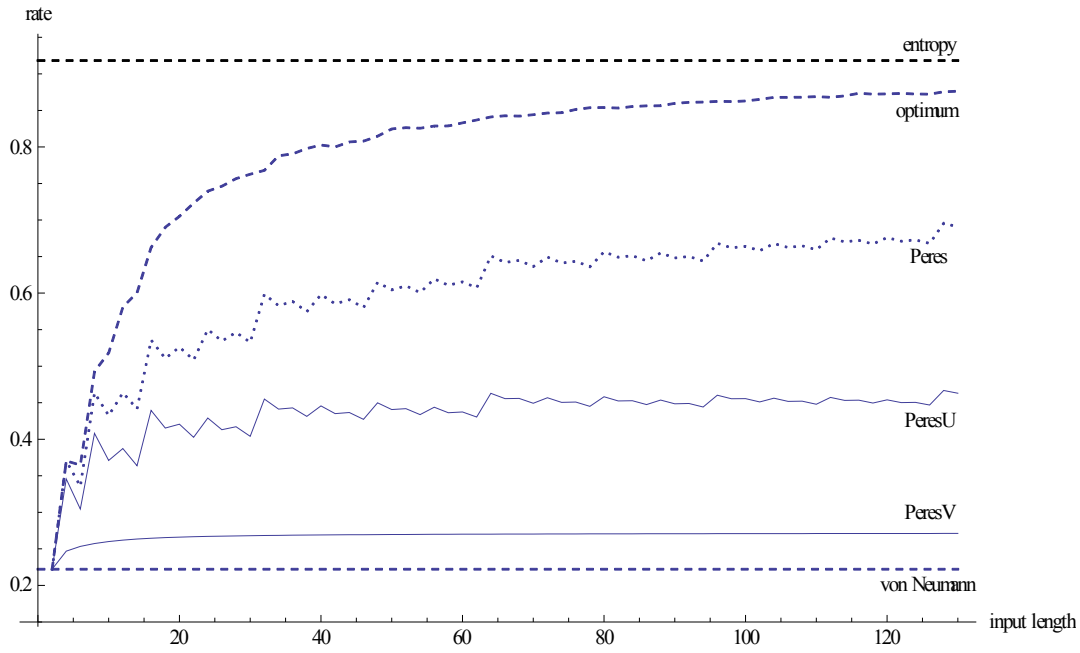

그림 1. 부 페레즈 함수들의 출력효율
Fig. 1. Output Rates of Sub-Peres Functions

$$P'(2n,k) = \sum_{x \in S_{2n,k}} |\Psi'(x)|.$$

그러면 Ψ'의 효율은 다음과 같이 쓸 수 있다:

$$r'(2n,p) = \frac{1}{2n} \sum_{k=0}^{2n} P'(2n,k)\, p^{2n-k} q^k.$$

따라서, 페레즈 함수의 경우와 같이, 효율 $r'(2n,p)$의 계산은 $P'(2n,k)$의 계산으로 귀결되고, 다음의 재귀식을 얻게 된다:

$$P'(2n,k) = 2^l \binom{n}{l} \binom{n-l}{\frac{k-l}{2}} l + 2^l \binom{n-l}{\frac{k-l}{2}} P'(n,l). \quad \text{(R1}'\text{)}$$

이제, (R1′)의 우변을 보면, 좌변에서 입력길이가 $2n$임에도, 홀수 길이의 입력에 대한 호출이 이루어 질 수 있다. 따라서, 홀수길이의 입력 x를 받으면, Ψ는 함수 Ψ_1, u가 먼저 적용되므로, x의 가장 마지막 비트를 버리고 남는 비트열 x'가 적용된 것과 마찬가지 이다. 이 과정은 $S_{n,k}$와 $S_{n-1,k} \cup S_{n-1,k-1}$사이의 일대일 대응을 주고, 우리의 경우 다음의 식이 만족함을 알 수 있다.

$$P'(n,k) = P'(n-1,k) + P'(n-1,k-1) \quad \text{(R2}'\text{)}$$

이 식을 이용하면, 홀수길이의 입력에 대한 $P'(n,k)$를 계산할 수 있다.

또한, $n < k$인 경우,

$$P'(2n,k) = P'(2n,2n-k). \quad \text{(R3}'\text{)}$$

그리고, 입력의 길이가 1인 경우 출력이 없으므로, 점화식의 초기조건으로서

$$P'(1,k) = 0, \quad k = 0,1. \quad \text{(R4}'\text{)}$$

을 더함으로써, $P'(n,k)$의 재귀적 정의가 식 (R1')—(R4')에 의해 완성된다.

같은 식으로, 동확률 부분집합 $S_{2n,k}$에서 함수 Ψ''의 총출력길이를

$$P''(2n,k) = \sum_{x \in S_{2n,k}} |\Psi''(x)|$$

라 하면, 역시 재귀식

$$P''(2n,k) = 2^l \binom{n}{l} \binom{n-l}{\frac{k-l}{2}} l + 2^l \binom{n}{l} P''(n-l, \frac{k-l}{2}) \quad \text{(R1}''\text{)}$$

을 얻고, (R2′′)—(R4′′)도 (R2′)—(R4′)와 비슷하게 얻는다.

이 재귀식들을 이용, 다이내믹 프로그래밍을 통해 두 함수 P'과 P''의 값을 계산할 수 있는데, 예를 들어,

$$P'(100,50) = 4\ 91831\ 30897\ 32339\ 62695\ 10071\ 82816,$$

$$P''(100,50) = 3\ 43721\ 26197\ 29060\ 93771\ 81435\ 85000.$$

이 값들과 비교하기 위해 페레즈 함수의 경우,

$$\sum_{x \in S_{100,50}} |\Psi(x)| = 7\ 37069\ 15585\ 63057\ 93374\ 77205\ 29592$$

이다.

두 함수 P'과 P''의 값을 이용하여 Ψ'와 Ψ''의 정확한 출력효율을 계산한 결과를 그림 1에 제시하였다. 페레즈 함수는 계산복잡도가 $O(n \log n)$인 것에 비하여 출력효율이 꾸준히 증가하여 엔트로피 한계치에 다가가지만, Ψ'와 Ψ''는 계산복잡도가 $O(n)$인 장점이 있는 반면, 출력효율은 비교적 짧은 입력 길이에서부터 아주 완만히 증가하여 거의 정체함을 관찰할 수 있다. 이는 [11]에서 논의한 바 있는 출력효율과 계산복잡도 사이의 트레이드오프(tradeoff)가 이 경우에도 성립함을 보여준다.

3.2 점근적 출력효율

위 결과에서, $p = 1/3$의 경우, 두 함수 Ψ'와 Ψ''의 출력효율이 각각 0.5와 0.3을 넘지 않으리라는 것을 예상할 수 있으나, 좀 더 정확한 출력효율의 극한은 다음과 같이 계산할 수 있다. 우선, Ψ'와 Ψ''의 점근적 출력효율을 다음과 같이 정의하자:

$$r'(p)=\lim_{n\to\infty} r'(n,p),$$

$$r''(p)=\lim_{n\to\infty} r''(n,p).$$

이제, 재귀호출의 횟수를 ν로 제한하는 함수를 다음과 같이 생각하자:

$$\Psi_\nu'(x)=\Psi_1(x)*\Psi_{\nu-1}'(u(x)),$$

$$\Psi_\nu''(x)=\Psi_1(x)*\Psi_{\nu-1}''(v(x)).$$

이 함수들은 재귀호출의 횟수가 제한되므로, 각각 Ψ'와 Ψ''에 비하여 출력효율이 낮으나, 다음과 같이, 그 점근적 효율 $r'_\nu(p)$과 $r''_\nu(p)$이 재귀식으로 표현되어 그 계산이 가능하게 되어, 우리가 애초 알고자 했던 $r'(p)$과 $r''(p)$의 분석에 이용할 수 있다. 점근적 효율 $r'(p)$과 $r''(p)$는 각각, ν가 무한대로 갈 때, $r'_\nu(p)$과 $r''_\nu(p)$의 극한과 같다. 이 사실에 주목하여, $r'_\nu(p)$과 $r''_\nu(p)$의 값은 다음과 같이 계산할 수 있다. 우선 편향 p에 대한 Ψ_ν'의 점근적 효율 $r'_\nu(p)$는 다음의 재귀식을 만족한다:

$$r_1'(p)=pq,$$

$$r_\nu'(p)=pq+\frac{1}{2}r'_{\nu-1}(p^2+q^2).$$

첫 번째 식은 Ψ_1'이 폰노이만 함수이므로 그 효율은 pq이고, 두 번째 식은 x가 편향 p의 베르누이 변수일 때 $u(x)$는 편향 p^2+q^2의 베르누이 변수이고, $|u(x)|=|x|/2$이라는 사실로부터 나온다. 이 재귀식을 풀면, 예를 들어

$$r_2'(p)=pq+\frac{pq}{p^2+q^2}$$

이 된다. 비슷한 식으로, $r''_\nu(p)$에 대한 재귀식을 다음과 같이 얻을 수 있다:

$$r_1''(p)=pq,$$

$$r_\nu''(p)=pq+\frac{1}{2}(p^2+q^2)\,r''_{\nu-1}\left(\frac{pq}{p^2+q^2}\right).$$

표 2는 $p=1/3$이고 ν의 값이 1부터 20일 때, 위의 재귀식을 이용하여 $r'_\nu(1/3)$과 $r''_\nu(1/3)$의 값을 각각 계산하여 표로 나타낸 것이다. 주어진 편향 p에 대하여 이 두 값들은 ν에 대하여 단조증가하는 무한급수이다. 그러므로 수렴하고, 표2의 계산결과에 의하여 그 수렴값의 근사치를 알 수 있는데, 각각

$$\lim_{n\to\infty} r'(n,1/3)=\lim_{v\to\infty} r_\nu'(1/3)=0.470662,$$

$$\lim_{n\to\infty} r''(n,1/3)=\lim_{v\to\infty} r_\nu''(1/3)=0.272059$$

이다.

표 2. Ψ_ν'와 Ψ_ν''의 점근적 효율

Table 2. Asymptotic Rates of Ψ_ν' and Ψ_ν''

ν	$r'_\nu(1/3)$	$r''_\nu(1/3)$
1	0.222222	0.222222
2	0.345679	0.266667
3	0.408169	0.271895
4	0.439419	0.272058
5	0.455044	0.272059
6	0.462857	0.272059
7	0.466763	0.272059
8	0.468716	0.272059
9	0.469693	0.272059
10	0.470181	0.272059
11	0.470425	0.272059
12	0.470547	0.272059
13	0.470608	0.272059
14	0.470639	0.272059
15	0.470654	0.272059
16	0.470662	0.272059
17	0.470666	0.272059
18	0.470668	0.272059
19	0.470669	0.272059
20	0.470669	0.272059

4. 인자함수 u와 v의 확률적 특성

입력 x에 대한 함수 Ψ의 첫 번째 재귀호출은 $\Psi_1(u(x))*\Psi'(u(u(x)))$이고, 따라서 $\Psi_1(u(x))$를 출력하고, 다음 재귀호출을 한다. $u^{(m)}(x)$를 x에 u를 m번 적

용한 값이라고 하면, 입력 x에 대한 함수 Ψ의 i번째 재귀호출은 $\Psi_1(u^{(i)}(x))*\Psi(u^{(i+1)}(x))$이고, $\Psi_1(u^{(i)}(x))$를 출력하고 다음 재귀호출을 한다. 따라서,

$$\Psi(x)=\Psi_1(x)*\cdots*\Psi_1(u^{(i)}(x))*\cdots.$$

같은 식으로

$$\Psi''(x)=\Psi_1(x)*\cdots*\Psi_1(v^{(i)}(x))*\cdots.$$

그러므로, x가 편향 p인 베르누이 공급원으로부터 뽑은 확률변수일 때 각각 $u^{(i)}(x)$와 $v^{(i)}(x)$의 확률적 특성이 함수 Ψ와 Ψ''의 출력 특성을 결정한다. 우선, 편향 p가 0에서 1/2까지 변할 때 폰노이만 함수 Ψ_1의 효율 pq는 다음 그림에서 보듯 p가 1/2로 다가갈 때 가장 높고, 0과 1에 가까울 때는 0으로 다가간다.

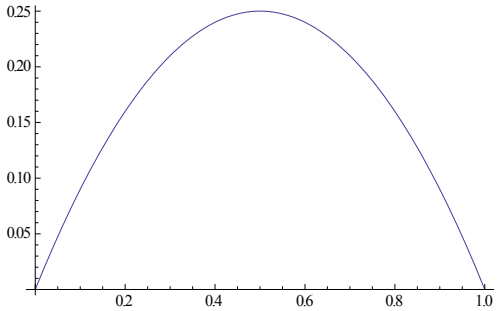

그림 2 편향에 따른 폰노이만 함수의 효율
Fig. 2 Rate of von Neumann function

이제 입력 x의 편향이 p일 때 u와 v를 반복적으로 적용한 결과인 $u^{(i)}(x)$와 $v^{(i)}(x)$의 편향이 어떻게 변해가는 지 살펴보자. 우선 $u(x)$의 편향은 위의 식 (P)의 도출과정에서 살펴본 바와 같이, p^2+q^2이다. 따라서 u를 한번 적용할 때 마다 편향은 다음의 사상을 적용한 것과 같다:

$$p\mapsto p^2+q^2.$$

같은 식으로 v를 한번 적용할 때, 편향은 다음의 사상에 따라 변하게 된다:

$$p\mapsto \frac{pq}{p^2+q^2}.$$

이 사상들에 의해 계산된 $u^{(i)}(x)$와 $v^{(i)}(x)$의 편향은 다음의 그림 3과 4에 나타난 바와 같다—시작하는 편향 값은 0.5부터 0.1사이의 값이고, 각각 u와 v를 반복적으로 적용할 때 그 결과로 나오는 확률변수의 편향을 나타내었다. 인자함수 u의 경우에는 $p=1.0$로 시작하는 경우를 제외하고는 모두 편향값 0.5로 수렴하고, 인자함수 v의 경우에는 $p=0.5$로 시작하는 경우를 제외하고는 모두 편향값 1.0으로 수렴한다. 시작하는 편향값이 0부터 0.5사이 일 때에도 같은 결과를 갖는다.

그림 2에서 볼 수 있듯이 폰노이만 함수 Ψ_1은 편향이 0.5일 때 그 효율이 0.25로서 가장 높고, 편향이 0이거나 1일 때는, 당연히, 효율이 0이다. 따라서 Ψ의 경우에는 재귀호출을 할 때마다 그 인자 $u^{(i)}(x)$의 편향이 0.5에 가까워 져서 인자의 길이에 비한 효율이 Ψ_1의 최대효율인 0.25로 다가가고, Ψ''의 경우에는 반대로 $v^{(i)}(x)$의 편향이 0이나 1로 다가감으로써 인자의 길이에 비한 효율이 0으로 다가가게 된다. 이 사실로부터, 그림 1이나 표 2에서 관찰할 수 있었듯이 Ψ''의 효율이 Ψ에 비하여 빨리 수렴값에 다가가는 이유를 설명할 수 있다. 다시 말해, $u^{(i)}(x)$는 i가 증가할수록 최대 엔트로피를 갖는 확률변수에 가까워지는 반면, $v^{(i)}(x)$는 최소엔트로피를 갖는 확률변수에 가까워진다. 인자의 길이의 측면에서도, $|u^{(i)}(x)|\geq|u^{(i)}(x)|$ 인 것을 고려하면, 위 사실은 Ψ이 Ψ''에 비하여 높은 출력효율을 갖는 이유를 잘 설명한다.

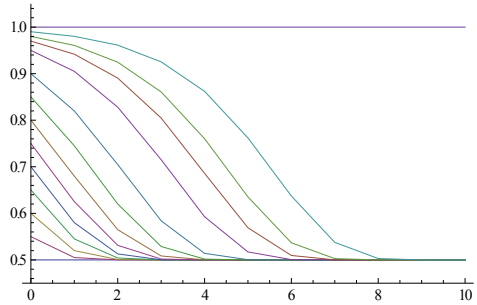

그림 3 $u^{(i)}(x)$의 편향, $i=0,...,10$.
Fig. 3 Bias of $u^{(i)}(x)$, $i=0,...,10$.

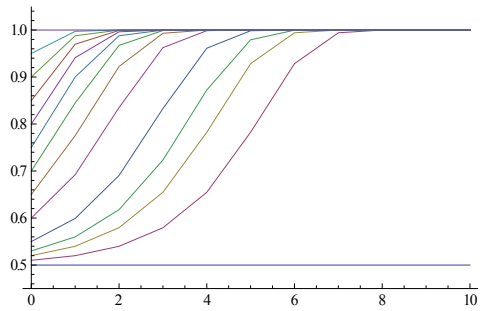

그림 4 $v^{(i)}(x)$의 편향, $i=0,...,10$.
Fig. 4 Bias of $v^{(i)}(x)$, $i=0,...,10$.

5. 꼬리재귀(tail recursion)를 이용한 구현

꼬리재귀(tail recursion)는, 재귀적으로 구현된 컴퓨터 프로그램으로사, 재귀호출이 이루어진 뒤 값을 반환하기만 하고 종료하는 것을 말한다. 꼬리재귀는 그 구현을 재귀적으로 하는 대신 반복수행으로 동등한 일을 수행하도록 할 수 있고, 따라서 재귀호출을 위한 스택을 필요로 하지 않는다. 페레즈 함수의 경우에는 재귀호출이 두 번 수행되므로 꼬리재귀가 아니다. 반면, 부 페레즈 함수들 Ψ와 Ψ'는 꼬리재귀로 볼 수 있고, 나아가 반복수행에 의해 동등한 구현을 할 수 있다. 예를 들어, Ψ는 다음과 같이 구현할 수 있다:

```
procedure PeresU
input: x∈{0,1}*

  while |x| ≥ 2
    output Ψ1(x)
    x←u(x)
  end while

end procedure
```

그림 5 반복수행에 의한 Ψ의 구현
Fig. 5 Iterative Implementation of Ψ

이 함수를 재귀적으로 구현하는 경우, 스택을 필요로 하고, 입력의 길이가 n일 때 재귀호출의 깊이가 $\lfloor \log_2 n \rfloor$이므로, 스택의 구현에 $O(\log_2 n)$의 메모리를 필요로 한다. 그러나 위와 같이 반복수행에 의한 구현은 $O(1)$의 메모리로 충분하다. 다른 부 페레즈 함수 Ψ'도 비슷한 식으로 구현할 수 있다. 반면, 페레즈 함수는 간단하기는 하나, 두 번의 재귀호출을 하므로 꼬리재귀로 만들 수 없어서 이와 같이 구현할 수 없다.

IV. 결 론

이 논문에서는 페레즈 함수와 같이 재귀적으로 정의되지만, 페레즈 함수에서 쓰이는 두 개의 인자함수를 각각 하나씩 이용하는 부 페레즈 함수 Ψ와 Ψ'에 대하여 살펴보았다. 이 부 페레즈 함수들은 하나의 인자함수만을 쓰기 때문에 출력효율이 페레즈 함수에 비하여 낮은 대신, 페레즈 함수와 달리 선형시간에 실행할 수 있다는 장점이 있다.

출력효율에 대한 분석은, 첫째, 입력길이에 대한 정확한 출력효율을 계산하기 위해, 페레즈 함수의 경우와 같이 동확률집합에서의 총출력길이를 구하는 재귀식을 구한 후, 다이내믹 프로그래밍을 이용하여 이 값들을 계산함으로써 정확한 출력효율을 제시하였다. 이 결과로부터 부 페레즈 함수들은 비교적 짧은 입력길이에서 출력효율이 수렴해 감을 관찰할 수 있고, 공급원의 편향 $p=1/3$인 경우, 그 점근적 효율이 각각 0.5와 0.3보다 약간 적은 값임을 관찰하였다. 둘째, 정확한 점근적 효율을 계산하는 방법을 제시하고, 그 값을 계산하였다. 부 페레즈 함수들은 입력길이가 증가할 때, 비교적 빠르게 그 효율이 수렴하기 때문에, 이와같이 계산한 점근적 효율을 각각 두 함수 Ψ와 Ψ'의 효율의 대표값으로 간주할 수 있다.

이 분석으로부터, $p=1/3$인 경우, Ψ와 Ψ'의 효율은 대략 0.47과 0.27정도이다. 비슷하게 정의된 이 두 가지 방법의 효율이 이런 차이를 보이는 것을 인자함수 u와 v의 확률적 성질을 분석함으로써 설명할 수 있었다.

실행시간의 측면에서 부 페레즈함수들은 선형시간에 실행될 뿐 아니라, 꼬리재귀로서 반복수행으로 동등한 구현을 할 수 있음으로써, 페레즈 함수의 재귀적 구현에서와 같이 스택을 필요로 하지 않는다.

널리 알려진 폰노이만 함수의 효율은, 편향 $p=1/3$인 경우, 0.21인데, Ψ의 효율은 폰노이만 함수에 비하여 두 배 이상 높은 효율을 가짐을 알 수 있고, 이 사실은 다른 편향 값에 대하여도 성립한다. 선형시간에 실행되고, 위에 언급한 구현에서의 장점 때문에, 부 페레즈 함수 Ψ는 전력이나 계산시간 등 계산자원이 제한된 모바일 기기와 같은 환경에서 폰노이만 함수를 대신하여 사용될 수 있다.

참고문헌

〔1〕 P. Diaconis. The search for randomness. at American Association for the Advancement of Science annual meeting. Feb. 14, 2004. Seattle.P. Diaconis, S. Holmes, and R. Montgomery.

〔2〕 P. Diaconis, S. Holmes, and R. Montgomery, Dynamical bias in the coin toss. SIAM Review, Vol. 49, No. 2, p.211, 2007.

〔3〕 John von Neumann. Various techniques for use in connection with random digits. Notes by G. E. Forsythe. In Monte Carlo Method, Applied Mathematics Series, volume 12, pages 36 - 38. U.S. National Bureau of Standards, Washington D.C., 1951. Reprinted in von Neumann's Collected Works 5 (Pergammon Press, 1963), 768 - 770.

〔4〕 B. Jun and P. Kocher. The Intel random number generator. White paper prepared for Intel Corporation, 1999. Cryptography Research, Inc.

〔5〕 C. E. Shannon and W. Weaver. The Mathematical Theory of Communication. The University of Illinois Press, Urbana, 1964.

〔6〕 T. M. Cover and J. A. Thomas. Elements of Information Theory. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA, 1991.

〔7〕 Peter Elias. The efficient construction of an unbiased random sequence. The Annals of Mathematical Statistics, Vol. 43, No.3, pp. 865 - 870, 1972.

〔8〕 Yuval Peres. Iterating von Neumann's procedure for extracting random bits. Annals of Statistics, Vol. 20, No. 1, pp. 590 - 597, 1992.

〔9〕 Sung-il Pae and Michael C. Loui. Optimal random number generation from a biased coin. In Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pages 1079 - 1088, January 2005.

〔10〕 Sung-il Pae and Michael C. Loui. Randomizing functions: Simulation of discrete probability distribution using a source of unknown distribution. IEEE Transactions on Information Theory, Vol. 52, No. 11, pp. 4965 - 4976, November 2006.

〔11〕 Sung-il Pae and Min-su Kim, A Hybrid Randomizing Function Based on Elias and Peres Method, Journal of The Korea Society of Computer and Information, Vol. 17, No. 12, pp. 149-158, Dec. 2012.

〔12〕 Sung-il Pae. Exact Computation of Output Rate of Peres's Algorithm for Random Number Generation, Information Processing Letters, 2013, to appear.

〔13〕 Min-su Kim, A Hybrid Randomizing Function Using Peres-Elias Method for Efficient Generation of Random Bits, Master's thesis, Hongik University, 2012.

저 자 소 개

배 성 일

1993년 서울대학교 자연대학 수학과 (이학사)

1997년 University of Illinois at Urbana-Champaign 수학과(석사)

2005년 University of Illinois at Urbana-Champaign 전산학과(박사)

현재: 홍익대학교 컴퓨터공학과 조교수

관심분야 : 알고리즘, 계산이론

Email: pae@hongik.ac.kr