Journal of The Korea Society of Computer and Information
Vol. 17, No. 12, December 2012

www.ksci.re.kr
http://dx.doi.org/10.9708/jksci/2012.17.12.149

일라이어스와 페레즈의 방식에 기반한 하이브리드 무작위화 함수

배 성 일*, 김 민 수*

A Hybrid Randomizing Function Based on Elias and Peres Method

Sung-il Pae*, Min-su Kim*

요 약

본 논문에서는 점근적으로 최적인 두가지의 무작위화 함수인 일라이어스(Elias) 함수와 페레즈(Peres) 함수의 장단점을 고려한 하이브리드 무작위화 함수를 제안한다. 무작위화 함수는 편향성이 있는 무작위수의 공급원으로부터 균등한 무작위수를 생성하는데 쓰이는 알고리즘을 수학적으로 추상화한 것이다. 일라이어스 함수와 페레즈 함수는 입력의 길이가 무한으로 증가함에 따라 그 출력효율성이 정보론적 한계치에 다가간다. 특히, 일라이어스 함수는 주어진 (유한의) 입력길이에 대해 최적인 무작위화 함수이다. 그러나 그 계산은 간단하지 않고, 주어진 입력길이에 의존한다. 반면, 페레즈 함수는 정해진 입력의 길이에 대해 출력효율이 최적이지는 않으나, 점근적으로는 최적이고, 간단한 재귀식에 의해 정의되어서 그 계산이 매우 간단하고 적은 메모리를 필요로 한다. 이러한 계산복잡도와 출력효율에 대한 두가지 무작위화 함수의 장단점에 주목하여, 각각의 장점을 고려한 하이브리드 무작위화 함수를 제안하고 이를 분석한다.

▸ Keywords : 무작위화함수, 일라이어스 함수, 페레즈 함수, 무작위수 생성

Abstract

Proposed is a hybrid randomizing function using two asymptotically optimal randomizing functions: Elias function and Peres function. Randomizing function is an mathematical abstraction of producing a uniform random bits from a source of randomness with bias. It is known that the output rate of Elias function and Peres function approaches to the information-theoretic upper bound. Especially, for each fixed input length, Elias function is optimal. However, its computation is relatively

•제1저자 : 배성일 •교신저자 : 배성일
•투고일 : 2012. 10. 23, 심사일 : 2012. 10. 30, 게재확정일 : 2012. 11. 8.
* 홍익대학교 컴퓨터공학과(Dept. of Computer Engineering, Hongik University)
•이 논문은 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2009-0077288).

complicated and depends on input lengths. On the contrary, Peres function is defined by a simple recursion. So its computation is much simpler, uniform over the input lengths, and runs on a small footprint. In view of this tradeoff between computational complexity and output efficiency, we propose a hybrid randomizing function that has strengths of the two randomizing functions and analyze it.

▸ Keywords : randomizing function, Elias function, Peres function, random number generation

I. 서 론

무작위수(random number)는 시뮬레이션, 암호, 컴퓨터 알고리즘, 수치해석 등, 정보화된 현대사회의 여러 분야에 중요하게 쓰인다. 무작위수는 컴퓨터가 광범위하게 쓰이기 전에도, 예컨대, 동전이나 주사위던지기의 형태로 생성되어 쓰였지만, 컴퓨터에 의한 고속의 기계적 계산이 가능해지고 따라서 컴퓨터가 인류의 여러가지 지적, 문화경제적 생활에 필수불가결하게 쓰임에 따라서 새로운 쓰임새가 계속 생겨나고 있고, 그에 걸맞게 고속, 대량으로 생성되어 소비되고 있다.

무작위수는 그 생성(generation)의 관점에서 크게 유사무작위수(pseudorandom number)와 진성무작위수(true random number)로 나눈다. 유사무작위수는 "무작위수"라는 용어에 이율배반적일 수 있으나, 정해진 알고리즘에 의해 생성된다는 점 때문에, 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[1]. 그래서 유사무작위수는 주로 시뮬레이션이나 무작위 알고리즘(randomized algorithm) 연구등에 주로 쓰인다. 유사무작위수에 대한 관심과 그 연구는 현대적 컴퓨터의 발명에 즈음하여 활발해졌는데, 이는 컴퓨터에 의한 고속의 계산과 정보처리가 가능해짐에 따라 이러한 용도에 무작위수가 대량으로 필요하게 되었다는 점을 생각하면 자연스런 일이다.

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 공급원이 될 수 있다. 진성무작위수는 재생성이 가능하지 않고, 유사무작위수에 비해 대량으로 생성하기 쉽지 않다. 따라서 무작위수를 필요로 하는 시뮬레이션 등에는 쓰기가 좋지 않다. 특히 통계적 특성을 제어하기 쉽지 않기 때문에 확률통계적 특성이 중요한 몬테카를로 시뮬레이션과 같은 분야에는 쓰기가 좋지 않다.

하지만 이러한 유사무작위수의 장점에도 불구하고 유사무작위수를 쓸 수 없는 응용분야도 있다. 그 중 가장 중요한 것이 암호이다. 현재 쓰이는 거의 모든 암호 프로토콜은 무작위수를 사용하는데, 예를 들면 암호화를 위한 키생성에 무작위수를 필요로 한다. 유사무작위수는 고정된 알고리즘에 의해 생성된다는 사실때문에 예측이 가능하고 이는 암호시스템에 치명적이다. 또한 진성무작위수는 유사무작위수를 컴퓨터에 의해 대량으로 생성할 때 그 씨드(seed)로써 쓰인다.

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[2]. 또한, 그 비율이 같지 않을 뿐만 아니라, 온도 등의 물리적 상황이 변함에 따라서 그 비율 자체도 일정치 않을 수 있음을 예상할 수 있다. 무작위수의 균등성(uniformity)은 암호와 같은 응용분야에서도 중요한데, 만약 암호에 쓰이는 무작위수가 편향되어(biased) 있다면, 이 성질이 그 암호체계를 공격하는데 쓰일 수 있다.

무작위화함수는 이러한 진성무작위수 공급원의 수학적 모델중 가장 간단한 베르누이 공급원(Bernoulli source)으로부터 균등한 분포를 가지는 무작위수를 생성하는 알고리즘의 수학적 추상화이다. 이 논문에서는 출력효율의 입장에서 점근적으로 최적(asymptotically optimal)인 일라이어스 함수와 페레즈 함수의 장단점을 고려한 하이브리드 무작위화 함수를 제안하고 분석한다.

II장에서는 무작위화 함수를 소개하고, 이 논문에서 다루게 되는 두 무작위화 함수인 일라이어스 함수와 페레즈 함수에 대한 기본적 사실들을 설명한다. III장에서는 위 두 함수를 이용한 하이브리드 함수를 제안하고, 제안한 함수가 무작

위화 함수임을 보인다. 또한, 출력효율을 계산하기 위한 재귀식을 도출하고 이를 이용하여 입력길이에 대한 정확한 출력효율을 계산하여 일라이어스 함수와 페레즈 함수의 출력효율과 비교한 결과를 제시한다.

II. 배경 및 관련 연구

1. 무작위화함수

폰노이만(von Neumann)은 1951년에 편향(bias)이 있는 동전으로부터 편향이 없는(unbiased) 동전을 얻는, 즉 각각의 확률이 1/2인 무작위 비트(random bits)를 생성하는 아주 간단한 방법을 제시한다[3]. 그 방법은 다음과 같다. 먼저 던지기를 1회 시행하면 나올 수 있는 사건은 앞면(H)과 뒷면(T)이고, 각각의 확률은 p와 $q=1-p$라 하자. 이 동전을 연속으로 두 번 던지는 독립 시행을 한다면 다음과 같은 네 가지 사건이 생길 수 있고 각각의 확률은 다음과 같다.

$$\Pr[HH]=p^2, \Pr[HT]=\Pr[TH]=pq, \Pr[TT]=q^2 \quad (1)$$

연속으로 앞면(HH)이나 뒷면(TT)이 나오면 그 결과를 무시한다. 그리고 앞면과 뒷면이 번갈아 (HT 또는 TH) 나오면 그 각각의 출력을 0과 1로 한다. 이 때 출력 0과 1은 발생 확률이 pq로 같아 편향없는 무작위수가 된다.

이 아이디어를 일반화하고 추상화하여, 무작위화 함수 $f:\{0,1\}^n\rightarrow\{0,1\}^*$는 길이 n인 입력을 편향이 있고 독립인 공급원으로부터 받아서 각 비트가 편향없고 독립인 이진수열(binary sequence)을 그 함수값으로 가진다[4, 5]. 예를 들어, 함수 $\Psi_1:\{0,1\}^2\rightarrow\{0,1\}^*$을 다음과 같이 정의하자:

$$\Psi_1(00)=\Psi_1(11)=\lambda;\ \Psi_1(01)=1;\ \Psi_1(10)=0.$$

여기서 λ는 길이가 0인 문자열, 즉 빈 문자열(empty string)이다. 그러면 이 함수는 무작위화 함수인데, 위에 설명한 폰노이만의 방법에 대응한다. 이 함수를 폰노이만 함수라 부르기로 하자.

위에서 가정한 무작위수의 공급원, 즉 개별시행이 독립이고 각 시행에서의 편향이 동일한 공급원을 베르누이 공급원(Bernoulli source)이라 한다. 아래의 논의에서는 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 함수의 효율(rate)은 한 개의 입력을 받았을 때 평균출력의 길이라고 정하자. 이 효율은 정보론적 한계치를 가지는 것으로 알려져 있는데, 베르누이 공급원의 편향 p에 대하여 항상 샤논 엔트로피(Shannon entropy) $H(p)$보다 작고, 이를 무작위수 생성에서의 엔트로피 한계치(entropy bound)라고 부른다[6, 7]. 편향 $p=1/3$인 경우, 폰노이만 함수의 효율은 $pq=2/9$로서 대략 0.22이다. 엔트로피 $H(1/3)$의 값이 약 0.92인 것을 고려하면 그 차이가 상당히 크다.

엔트로피 한계치에 수렴하는 효율을 가진 무작위수 생성법 두 가지가 각각 일라이어스(Elias)와 페레즈(Peres)에 의해 제안 되었는데, 아래에서 자세히 설명한다.

무작위화 함수의 자세한 이론은 [5]를 참조한다.

2. 일라이어스 함수

2.1 일라이어스 함수와 최적효율

일라이어스(Elias)는 1972년에 아래에서 설명하는 무작위화함수 $E_n:\{0,1\}^n\rightarrow\{0,1\}^*$을 제안했는데[8], 이 함수는 각 입력길이 n에 대하여 아래의 정리 1에 주어진 최대효율을 가진다.

길이가 n인 이진수열의 집합 $\{0,1\}^n$의 원소들 중 1의 개수가 k인 것들의 부분집합을 $S_{n,k}$라 하자. 만약 $x\in S_{n,k}$라면, $\Pr(x)=p^{n-k}q^k$이고, $\{0,1\}^n$는 $n+1$개의 같은 확률을 가진 이진수열들의 집합으로 다음과 같이 나누어진다:

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

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. 예를 들어,

$$|S_{10,3}|=\binom{10}{3}=120=2^6+2^5+2^4+2^3$$

이다. 이때, 집합 $S_{n,k}$를 다시 각각 크기가 2^{n_i}인 부분집합 $S_{n,k}^{(i)}$들의 합으로 다음과 같이 나눌 수 있다:

$$S_{n,k} = S_{n,k}^{(1)} \cup S_{n,k}^{(2)} \cup \cdots \cup S_{n,k}^{(l)} \quad (2)$$

따라서, 주어진 입력 $x \in \{0,1\}^n$에 대하여 x가 포함되는 유일한 부분집합 $S_{n,k}^{(i)}$가 정해진다. 이때, 일라이어스 함수의 출력값 $E_n(x)$는 $\{0,1\}^{n_i}$의 원소중 하나로 유일하게 정해진다.

위와 같이 함수 E_n을 정의할 때, 식 (2)의 분할은, 다시 말해 $S_{n,k}$의 어떤 원소가 어떤 $S_{n,k}^{(i)}$에 포함되는 지는, 정하지 않음에 유의하자. 실제 구현에서는, 예를 들어, 사전순서에 의해 이 분할을 정할 수 있다. 또한 $S_{n,k}^{(i)}$에 포함되는 x에 대하여 함수값 $E_n(x)$를 어떤 식으로 길이 n_i의 비트열로 유일하게 정하는 지도 정하지 않는데, 이 또한 실제 구현에서는 사전순서를 이용할 수 있다. 이 두 가지 임의성은 일라이어스함수가 무작위화 함수라는 사실과 주어진 입력의 길이에 대하여 항상 최적이라는 사실에는 영향을 미치지 않지만, 일라이어스의 구현과 그 계산복잡도에는 영향을 미치게 되는데, 사전순서에 의하지 않고 더 나은 계산복잡도를 가지게 되는 방법이 있는가 하는 문제는 일라이어스 함수에 관한 중요한 미해결 문제이다.

이제, 위와 같이 주어졌을 때, 다음과 같이 함수 $A(n,k)$를 정의하자.

$$A(n,k) = n_1 2^{n_1} + n_2 2^{n_2} + \cdots + n_i 2^{n_i} \quad (E1)$$

그러면 $A(n,k)$는 $S_{n,k}$의 모든 원소에 대한 일라이어스 함수의 출력크기의 합이고, 임의의 무작위화 함수 $f : \{0,1\}^n \rightarrow \{0,1\}^*$에 대하여, 다음의 사실이 성립함이 알려져 있다[4, 5].

$$\sum_{x \in S_{n,k}} |f(x)| \le A(n,k).$$

그러므로 다음과 같은 결론을 얻을 수 있다.

정리 1 [4, 5] 입력길이가 n인 무작위화 함수의 최대 효율은 $\frac{1}{n}\sum_{k=0}^{n} A(n,k) p^{n-k} q^k$이다.

무작위화 함수 E_n은 이와 같이 각 입력길이 n에 대하여, 위의 정리 1의 최대효율을 가질 뿐만 아니라 입력길이 n이 무한대로 갈 때 그 효율이 엔트로피 한계치에 접근함이 알려져 있다[8].

2.2 일라이어스 함수의 계산

위 2.1절에서 설명한 바와 같이, 식 (2)의 분할을 정하는 방법과 각 부분집합 $S_{n,k}^{(i)}$에서의 함수값(출력)을 정하는 방법에 따라 일라이어스 함수의 계산은 달라질 수 있다. 각각 사전순서에 의한 구현과 이에 대한 계산복잡도가 알려져 있다[5, 9]. 예를 들어, $x = 0101111$이라 하자. 그러면 $n = 7$이고 $k = 5$이다. 사전순서에 의하면 x는, 0부터 세기 시작하여, $S_{7,5}$의 7번째 원소이다. $S_{7,5}$의 크기는 $\binom{7}{5} = 21 = 2^4 + 2^2 + 2^0$ 이고, 이 이진수전개에 따라서 x는 크기가 2^4인 $S_{7,5}^{(1)}$에 속하고, 역시 사전순서에 의하여 $\{0,1\}^4$에서 7번째 원소인 0111이 $E_7(x)$의 값이 된다.

이와 같은 방식으로 계산할 때의 계산복잡도는 주어진 입력x가 $S_{n,k}$에서의 사전순서에 의한 순위계산의 복잡도에 의해 정해지는데 이는 $O(nk\mu(n))$시간에 가능함이 알려져 있다[5]. 여기서 $k = O(n)$이므로 이 방식을 이용한 일라이어스 함수의 계산 복잡도는 $O(n^2\mu(n))$이다. Ryabko와 Machikina[9]에 의한 $O(n \log^3 n \log\log n)$의 복잡도를 가지는 알고리즘이 알려져 있으나, 실제에 사용하기는 매우 복잡한 방법이다. 일라이어스 함수의 계산과 계산복잡도에 대하여는 [5]를 참조한다.

3. 페레즈 함수

3.1 페레즈 함수의 정의

페레즈(Peres)가 1992년에 제안한 페레즈 함수 Ψ_ν는 폰노이만 함수와 두 가지 특별한 함수 u와 v를 이용하여 재귀적으로 정의한 무작위화 함수이다[10]. 페레즈 함수 Ψ_ν는 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,

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

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. 예를 들어,

$$\begin{aligned}\Psi_3(001011011)&=\Psi_1(00101101)\\&\quad *\Psi_2(u(00101101))*\Psi_2(v(00101101))\\&=1*\Psi_2(0101)*\Psi_2(00)\\&=1*(\Psi_1(0101)*\Psi_1(11)*\Psi_1(\lambda))\\&\quad *(\Psi_1(00)*\Psi_1(0)*\Psi_1(0))\\&=1*11*\lambda\\&=111\end{aligned}$$

이다.

페레즈에 의하면, 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.

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

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[10]. 그러나 정해진 ν와 유한인 입력의 길이에 대하여는 그 효율이 일라이어스 함수와 같이 최적이 아님을 알 수 있다. 이 사실은 심지어 재귀호출의 깊이 ν가 무한히 커질 때에도 마찬가지 인데, 다음의 3.3절에서 자세히 다룬다.

3.2 페레즈 함수의 시간복잡도

페레즈함수를 분석하기 위해, 본질적으로 위의 정의와 같으나 재귀호출의 깊이가 제한되지 않는 다음의 함수를 생각하자.

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

이 함수는 Ψ_ν와 같이 모든 입력길이에 대하여 무작위화 함수이다. 다만 재귀호출의 깊이가 제한되지 않기 때문에 출력 효율은 임의의 Ψ_ν보다 더 크다. 이 함수 또한 페레즈 함수로 부르기로 하자. 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 Ψ의 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.

3.3 페레즈 함수의 효율

페레즈 함수의 점근적 효율성은 위에서 살펴본 바와 같고, 각 입력길이에 대한 정확한 효율은 [11]에 알려져 있다. 일라이어스 함수와 페레즈 함수는 둘 다 점근적으로 최적이기는 하지만, 페레즈 함수는 일라이어스 함수와 같이 각 입력길이에 대하여 최적이 아니다. 짝수길이의 입력에 대해서만 생각할 때, 페레즈 함수 Ψ의 효율은 입력길이 2와 4에 대해서는 최적이다. 그러나 입력길이 6부터는 일라이어스 함수보다 효율이 낮다.

III. 하이브리드 무작위화 함수

1. 정의

위에 설명한 일라이어스 함수와 페레즈 함수의 장점을 이용하면서 단점을 보완하는 다음과 같은 함수를 생각하자.

$$F_t(x)=\begin{cases}E(x) & |x|\le t\text{일 때},\\ \Psi_1(x)*F_t(u(x))*F_t(v(x)) & |x|>t\text{일 때}.\end{cases}$$

여기서 $E(x)$는 일라이어스 함수를 모든 입력길이에 대해 확장한 함수, 즉 $E(x)=E_{|x|}(x)$이다. 입력의 길이가 t보다 작거나 같으면 일라이어스 함수를 적용하고, 그렇지 않으

면 페레즈 함수와 같은 방식으로 폰노이만 함수를 적용하고, $u(x)$와 $v(x)$에 재귀적으로 F_t를 적용한다. 이 재귀호출에서 $u(x)$또는 $v(x)$의 길이가 t보다 작거나 같으면 일라이어스 함수를 적용하여 $E(u(x))$또는 $E(v(x))$를 계산하고 재귀호출은 끝나게 된다.

예 $x=10110100011001$, $t=10$인 경우를 생각하자.

$$\Psi_1(10110100011001)=01101,$$

$$u(10110100011001)=1010111,$$

$$v(10110100011001)=10$$

이므로

$$F_{10}(10110100011001)=01101*F_{10}(1010111)*F_{10}(10)$$
$$=01101*E(1010111)*E(10).$$

II.2.2절에서 정한 방식으로 일라이어스 함수를 계산한다고 하면

$$E(1010111)=E_7(1010111)=0111,$$

$$E(10)=E_2(10)=\Psi_1(10)=0.$$

따라서

$$F_{10}(10110100011001)=01101*0111*0=0110101110.$$

함수 F_t가 무작위화 함수임을 보이기 위해 다음을 생각하자. 우선 입력의 길이가 $2n$인 경우를 생각하자. 모든 입력 중 1의 개수가 k인 입력을 모아 놓은 집합 $S_{2n,k}$는 폰노이만 함수에 대한 출력의 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.

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

이제 부분집합 C_l은 다음과 같은 중요한 성질을 가진다.

보조정리 S [11] 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.

함수 $f:\{0,1\}^n\to\{0,1\}^*$가 각 출력길이 d에 대해 $\{0,1\}^d$의 모든 원소에 대해 같은 확률의 출력을 할 때 f를 추출함수라 부르고, 또 f를 추출적(extracting)이라 부르기도 한다. 이 경우 추출함수 f는 동확률 부분집합 $S_{n,k}$에서 각 출력길이 d에 대해 $\{0,1\}^d$의 모든 원소를 같은 횟수로 출력하고, 이 조건은 f가 추출적이라는 것과 동치조건이다. 추출함수는 무작위 함수이다. 그러나 모든 무작위화 함수가 추출함수인 것은 아니다[8, 5]. 일라이어스 함수와 페레즈 함수는 추출함수이다. 그리고 이 논문에서 제안한 함수 F_t도 추출함수인데, 우리는 F_t가 추출함수임을 보임으로써 무작위화 함수임을 보이겠다.

정리 2 함수 F_t는 무작위화 함수이다.

증명: 입력의 길이에 대한 귀납법을 쓴다. 입력의 길이가 t보다 작거나 같으면 일라이어스 함수가 추출함수이므로 성립한다. 이 사실을 귀납법의 초기 조건으로 삼는다. 입력의 길이가 2^m이하 일 때, 이 명제가 참이라 가정하자. 입력의 길이 $2n=2^{m+1}$ 이라 하자. 이 명제는 각각의 동확률 부분집합 $S_{2n,k}$에 대해서만 보이면 충분하고, 나아가 $S_{2n,k}$의 부분집합들인 C_l에 대해서 보이면 충분하다. 보조정리 S에 의하여 $\Psi_1(x)$, $u(x)$, $v(x)$는 각각 확률적으로 독립이고, $\Psi_1(C_l)=\{0,1\}^l$. 그리고 $|u(x)|=2^m$, $|v(x)|\le 2^m$이고, $u(C_l)=S_{n,l}$, $v(C_l)=S_{n-l,(k-l)/2}$이므로, 귀납법의 가정에 의하여 $F_t(u(x))$와 $F_t(v(x))$는 균등분포하게 되고, 따라서

$$\Psi_1(x)*F_t(u(x))*F_t(v(x))$$

는 균등분포한다. 따라서 F_t는 추출적이고, 그러므로 무작위화 함수이다. [증명끝]

위의 II.3.2절에서 페레즈 함수의 시간복잡도를 증명한 것과 같은 식으로 t의 값에 상관없이 F_t의 시간복잡도는 $O(n\log n)$임을 보일 수 있다.

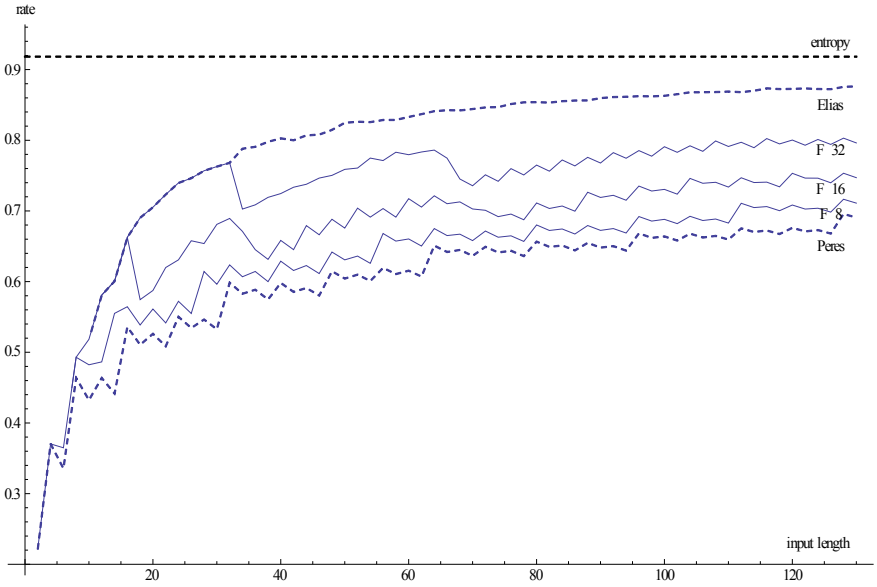


그림 1. 입력길이에 대한 출력효율 비교 (편향 $p=1/3$)
Fig. 1. Comparison of output rates with respect to input length (bias $p=1/3$)

2. 효율계산

입력길이 $2n$에 대하여 동확률 부분집합 $S_{2n,k}$에서 함수 F_t의 총출력길이를 다음과 쓰기로 하자.

$$Q_t(2n,k)=\sum_{x\in S_{2n,k}}|F_t(x)|.$$

그러면 F_t의 효율은 다음과 같이 쓸 수 있다:

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

따라서 F_t의 효율계산은 $Q_t(2n,k)$의 계산으로 귀결된다. 함수 $Q_t(2n,k)$를 효율적으로 계산하기 위해, 이 함수의 재귀식을 구하고자 한다. 그러면, 예를 들어 다이내믹 프로그래밍을 이용하여 $Q_t(2n,k)$의 값을 구할 수 있다. 물론 F_t의 효율을 구하기 위해서 $\{0,1\}^{2n}$의 모든 원소에 대하여 F_t를 계산하여 출력길이를 구할 수 있으나, 이 방법은 입력길이가 조금만 커지면, 예를 들어 32 이상이면, 실질적으로 불가능하다.

정리 R 함수 $Q_t(2n,k)$는 다음과 같이 주어진 재귀식을 만족한다.

$$Q_t(2n,k)=\begin{cases}A(2n,k) & 2n\le t\text{일 때}\\ \sum_l 2^l S(n,k,l) & 2n>t\text{일 때}\end{cases}\qquad\text{(R1)}$$

여기서, 두번째 경우의 합은 위의 식 (P)에서의 l들에 대한 합이고, $S(n,l,k)$는 다음과 같다:

$$\binom{n}{l}\binom{n-l}{\frac{k-l}{2}}l+\binom{n-l}{\frac{k-l}{2}}Q_t(n,l)+\binom{n}{l}Q_t(n-l,\frac{k-l}{2})\qquad\text{(F)}$$

증명 우선, 입력길이 $2n\le t$ 일 때는 F_t의 정의에 의해서 $Q_t(2n,k)=A(2n,k)$이다. 입력길이 $2n>t$인 경우를 생각하자. 보조정리 S에 의하여,

$$|C_l|=2^l\binom{n}{l}\binom{n-l}{\frac{k-l}{2}}.$$

입력 x에 대한 F_t의 출력길이는

$$|F_t(x)|=|\Psi_1(x)|+|F_t(u(x))|+|F_t(v(x))|$$

임을 염두에 두고, 식 (P)에 주어진 $S_{2n,k}$의 분할을 이용하여, 각 C_l에서의 출력길이의 합을 구한다. 첫째, $x\in C_l$이면 $|\Psi_1(x)|=l$이므로,

$$\sum_{x\in C_l}|\Psi_1(x)| = 2^l\binom{n}{l}\binom{n-l}{\frac{k-l}{2}}l. \qquad \text{(A)}$$

두번째로, 역시 보조정리 S의 일대일 대응관계에 의하여, 각 $z\in S_{n,l}$에 대하여, $u(x)=z$인 x가 $2^l\binom{n-l}{\frac{k-l}{2}}$개 있고, $\sum_{x\in S_{n,l}}|F_t(x)| = Q_t(n,l)$이므로,

$$\sum_{x\in C_l}|F_t(u(x))| = 2^l\binom{n-l}{\frac{k-l}{2}}Q_t(n,l). \qquad \text{(B)}$$

세번째로, 비슷한 식으로 각 $w\in S_{n-l,(k-l)/2}$에 대하여, $v(x)=w$인 x가 $2^l\binom{n}{l}$개 있으므로,

$$\sum_{x\in C_l}|F_t(v(x))| = 2^l\binom{n}{l}Q_t(n-l,\frac{k-l}{2}). \qquad \text{(C)}$$

이다. 식 (A), (B), (C)을 합치면, 원하는 식을 얻는다. 〔증명끝〕

식 (F)에서, $Q_t(n,l)$과 $Q_t(n-l,(k-l)/2)$에 대한 재귀호출이 이루어지는데, 이때 홀수 길이의 입력에 대한 호출이 이루어 질 수 있다. 홀수길이의 입력 x를 받으면, F_t는 함수 Ψ_1, u, v가 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 경우에 다음의 식이 만족함을 알 수 있다.

$$Q_t(n,k) = Q_t(n-1,k) + Q_t(n-1,k-1) \qquad \text{(R2)}$$

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

보조정리 C $n<k$인 경우,

$$Q_t(2n,k) = Q_t(2n,2n-k). \qquad \text{(R3)}$$

증명 이진수열 x에서 0과 1을 서로 교환해서 생기는 이진수열을 $\overline{x}$라 하자. 그러면,

$$\begin{aligned} Q_t(2n,k) &= \sum_{x\in S_{2n,k}}|F_t(x)| \\ &= \sum_{x\in S_{2n,2n-k}}|F_t(\overline{x})| \\ &= \sum_{x\in S_{2n,2n-k}}\left(|\Psi_1(\overline{x})|+|F_t(u(\overline{x}))|+|F_t(v(\overline{x}))|\right) \end{aligned}$$

인데, $\Psi_1(\overline{x}) = \overline{\Psi_1(x)}$, $u(\overline{x}) = u(x)$이므로

$$\sum_{x\in S_{2n,2n-k}}|\Psi_1(\overline{x})| = \sum_{x\in S_{2n,2n-k}}|\overline{\Psi_1(x)}| = \sum_{x\in S_{2n,2n-k}}|\Psi_1(x)|$$

이고,

$$\sum_{x\in S_{2n,2n-k}}|F_t(u(\overline{x}))| = \sum_{x\in S_{2n,2n-k}}|F_t(u(x))|$$

이다. 그러므로,

$$\sum_{x\in S_{2n,2n-k}}|F_t(v(\overline{x}))| = \sum_{x\in S_{2n,2n-k}}|F_t(v(x))|\,.$$

를 보이면, (R3)을 보이게 된다. 증명의 나머지에서는 이 등식을 입력의 길이에 대한 귀납법으로 보이겠다.

우선, 귀납적 증명의 기본단계로서, 동확률 부분집합집합 $S_{m,j}$에서 일라이어스 함수의 총 출력길이 합을 생각하면, 다음이 성립한다:

$$\sum_{x\in S_{m,j}}|E(x)| = \sum_{x\in S_{m,m-j}}|E(x)|$$

이제, $2n-k<n$이므로, 식 (P)의 분할과 v의 성질 $v(\overline{x}) = \overline{v(x)}$에 의해서

$$\begin{aligned} \sum_{x\in S_{2n,2n-k}}|F_t(v(\overline{x}))| &= \sum_l\sum_{x\in C_l}|F_t(v(\overline{x}))| \\ &= \sum_l\sum_{x\in C_l}|F_t(\overline{v(x)})|. \end{aligned} \qquad \text{(D)}$$

보조정리 S에 의해 $v(C_l)$은 정확히 $2^l\binom{n}{l}$개의 $S_{n-l,(2n-k-l)/2}$이다. 따라서 $\overline{v(C_l)}$는 $2^l\binom{n}{l}$개의 $S_{n-l,(k-l)/2}$이 된다. 귀납법의 가정에 따라,

$$\sum_{x\in S_{n-l,(k-l)/2}}|F_t(x)| = \sum_{x\in S_{n-l,(k-l)/2}}|F_t(\overline{x})|$$

이고, 이를 이용하여 (D)의 과정을 거꾸로 하여

$$\sum_{x\in S_{2n,2n-k}}|F_t(v(\overline{x}))| = \sum_{x\in S_{2n,2n-k}}|F_t(v(x))|$$

를 얻는다. 〔증명끝〕

식 (R2)는 $Q_t(2n,k)$를 계산하기 위해 반드시 필요한 것은 아니나, 다이내믹 프로그래밍에 의해 $Q_t(2n,k)$를 계산할 때 (R2)를 사용하지 않을 때에 비하여 계산해야 하는 항의 개수를 절반으로 줄여준다. 마지막으로, 입력의 길이가 1인 경우 출력이 없으므로, 점화식의 초기조건으로서

$$Q_t(1,k)=0, \quad k=0,1. \tag{R4}$$

을 더함으로써, $Q_t(n,k)$의 재귀적 정의가 식 (R1)--(R4)에 의해 완성된다.

3. 계산결과

위의 점화식을 이용하면 예를 들어 다음의 값들을 계산할 수 있다:

$$Q_8(100,50) = 7\ 65397\ 21137\ 16981\ 52689\ 77427\ 04664$$

$$Q_{16}(100,50) = 8\ 15787\ 80048\ 83666\ 41631\ 23946\ 21400$$

$$Q_{32}(100,50) = 8\ 87532\ 14613\ 41264\ 36070\ 33236\ 98008$$

이 값들과 비교하기 위해 페레즈함수의 경우,

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

이고, 일라이어스 함수의 경우,

$$\sum_{x\in S_{100,50}} |E(x)| = 9\ 63297\ 00815\ 24591\ 75552\ 91573\ 74520.$$

이다. 각각, $t=8,16,32$인 경우, $S_{100,50}$에서의 총출력은 페레즈 함수보다는 크고 일라이어스 함수보다는 작음을 볼 수 있다.

그림 1은, 이와 같은 $Q_t(n,k)$의 계산을 이용하여, 역시 $t=8,16,32$인 경우에 대해 하이브리드 무작위화 함수 F_t의 효율을 계산한 결과를 보여준다. 그림에 나타난 모든 알고리즘의 출력효율은 베르누이 공급원의 편향 $p=1/3$에 대한 값이다. 다른 값의 편향에 대하여도 비슷한 결과를 얻는다. 그 정의에서 알 수 있는 바와 같이, 각 t의 값에 대해, F_t의 효율은 입력길이 $n \le t$일 때는 일라이어스 함수와 같고, 입력길이가 t보다 커질 때부터 효율이 차이가 남을 볼 수 있다.

IV. 결 론

이 논문에서는 점근적으로 최적인 두 가지의 무작위화 함수인 일라이어스 함수와 페레즈 함수의 계산적 측면과 출력효율의 측면을 살펴보고, 그 장점을 살리고 단점을 보완하는 새로운 무작위화 함수를 제안하였다. 이 방법은 주어진 인자 t에 대하여, 입력이 t보다 긴 경우에는 계산부하가 적은 페레즈의 방법과 같은 식의 재귀적인 구조를 가지고, 입력이 t보다 짧은 경우에는 출력효율이 최적인 일라이어스의 방법을 이용한다. 이렇게 정의된 함수가 역시 무작위화 함수임을 보였고, 재귀적 구조의 분석을 통하여 정확한 출력효율을 계산하는 점화식을 도출하고 입력길이에 대한 정확한 출력효율을 계산하여 그 결과를 일라이어스 함수와 페레즈 함수와 비교하여 제시하였다.

이 연구결과와 관련한 추후 연구문제로서는 페레즈 함수와 같이 재귀적으로 정의 되어지는 무작위화 함수들에 대한 연구가 있다. 예를 들어, 이러한 함수들 중 출력효율이 최적인 함수는 어떤 것인가 하는 질문을 할 수 있다. 페레즈 함수의 장점들은 주로 이 함수가 간단한 재귀식에 의해 직접적으로 (explicitly) 정의됨에 있는데, 이 질문에 대한 답은 계산적으로 간단하면서 출력효율이 좋은 무작위화 함수에 관한 이해를 도울 것이다.

참고문헌

[1] Donald E. Knuth. The Art of Computer Programming, Seminumerical Algorithms, volume 2. Addison-Wesley, third edition, 1998.

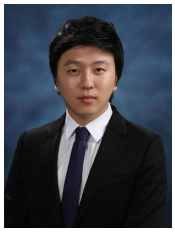
[2] Benjamin Jun and Paul Kocher. The INTEL Random Number Generator, Cryptography Research, Inc. White paper prepared for INTEL Corporation, April 22, 1999.

[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〕 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.
〔5〕 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, 52(11):4965-4976, November 2006.
〔6〕 Claude Elwood Shannon and Warren Weaver. The Mathematical Theory of Communication. The University of Illinois Press, Urbana, 1964.
〔7〕 Thomas M. Cover and Joy A. Thomas. Elements of Information Theory. Wiley Series in Telecommunications. John Wiley & Sons, New York, NY, USA, 1991.
〔8〕 Peter Elias. The efficient construction of an unbiased random sequence. The Annals of Mathematical Statistics, 43(3):865-870, 1972.
〔9〕 B. Y. Ryabko, E. Matchikina, Fast and efficient construction of an unbiased random sequence, IEEE Transactions on Information Theory 46 (3) (2000) 1090-1093.
〔10〕 Yuval Peres. Iterating von Neumann's procedure for extracting random bits. Annals of Statistics, 20(1):590-597, 1992.
〔11〕 Sung-il Pae. Exact Computation of Output Rate of Peres's Algorithm for Random Number Generation, submitted.
〔12〕 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 rbana-Champaign 전산학과(박사)
현재: 홍익대학교 컴퓨터공학과 조교수
관심분야 : 알고리즘, 계산이론
Email: pae@hongik.ac.kr



김 민 수
2010년 홍익대학교 공과대학 컴퓨터공학과(공학사)
2012년 홍익대학교 대학원 컴퓨터공학과(석사)
현재: (주)파수닷컴 개발본부 연구원
관심분야 : 알고리즘, 계산이론
Email: ms.supremus@gmail.com