THE FIRST RETURN TIME TEST OF PSEUDORANDOM NUMBERS

DONG HAN KIM

ABSTRACT. An algorithm for obtaining the probability distribution of the first return time \( R_n \) for the initial \( n \)-block with overlapping is presented and used to test pseudorandom number generators. The averages and the standard deviations of \( \log R_n \) are computed theoretically and the \( Z \)-test is applied.

1. INTRODUCTION

We introduce a new method of testing PRNGs based on the first return time of some fixed length blocks in a randomly generated binary sequence. The first return time is closely related to entropy, which is the central idea in the information theory founded by C. Shannon[13]. For a binary source it is defined to be the limit of \( -\frac{1}{n} \sum_{i=1}^{2^n} p_i \log p_i \) as \( n \) increases to infinity where the \( p_i \)'s are the relative frequencies of \( 2^n \) blocks of length \( n \) in a typical binary sequence generated by the source. Entropy measures the information content or the amount of randomness. In data compression the entropy measures the maximum compression rate. If there are more patterns, that is, less randomness in a given sequence, then it has smaller entropy and can be compressed more.

In this article the first return time in a random binary sequence is investigated. Consider a stationary ergodic binary process on the space of infinite sequence \( \{0,1\}^\infty, \mu \), where \( \mu \) is the shift invariant ergodic probability measure on the \( \sigma \)-field generated by finite dimensional cylinders. For each sample sequence \( x \) define the first return time by

\[
R_n(x) = \min\{j \geq 1 : x_1 \ldots x_n = x_{j+1} \ldots x_{j+n}\}.
\]

A.D. Wyner and Ziv[15] proved that \( \frac{1}{n} \log R_n(x) \) converges to the entropy of the sequence in measure and Ornstein and Weiss[9] showed that the convergence is pointwise. Later, A.J. Wyner[16] discovered that for a stationary aperiodic Markov chain with entropy \( h \) the random variable \( \frac{1}{n} \log R_n \) has approximately a normal distribution with mean
DONG HAN KIM

$h$ in a suitable sense. For a sharp estimate of the convergence of the average of $\frac{1}{n} \log R_n$, see [2].

Kac's lemma[4] implies that $E[R_n|x^n = a^n] = 1/\mu(a^n)$ for any finite string $a^n$ with $\mu(a^n) > 0$. Since Kac's lemma implies $E[R_n] = \#\{a^n \in A^n : \mu(a^n) > 0\}$, we have

$$\lim_{n \to \infty} \frac{1}{n} \log E[R_n] = \text{topological entropy}.$$  

This implies that the algorithm using $\log R_n$ is much more efficient than the algorithm of $R_n$.

Since Maurer's work[8], the nonoverlapping first return time which corresponds to

$$R_{(n)}(x) = \min\{j \geq 1 : x_1 \ldots x_n = x_{jn+1} \ldots x_{jn+n}\},$$

has been investigated to be used in cryptography or in testing PRNGs. Nonoverlapping algorithm is relatively easier to analyze but overlapping method is more efficient and natural.

2. The Probability Distribution of the First Return Time

A block is a finite sequence of elements of $A$ and an $n$-block is a block of length $n$. For an $n$-block $B = b_1b_2 \cdots b_n$, we write $B^i_j = b_i b_{i+1} \cdots b_j, 1 \leq i \leq j \leq n$. Since the distribution of return time is different from block to block. We classify the blocks to each set of blocks have the same return time distribution.

**Definition 1.** Let $B$ be an $n$-block. Suppose $m$ satisfies $1 \leq m < n$ and

$$B^n_{m+1} = B^{n-m}_1.$$

The smallest such $m$ is denoted by $\lambda_1(B)$, and the next smallest such $m$ which is not a multiple of $\lambda_1(B)$ is called $\lambda_2(B)$, and we define $\lambda_k(B)$ by the smallest such $m$ which is not a multiple of $\lambda_i(B)$ for every $i < k$. Let $\Lambda(B) = \{\lambda_1(B), \lambda_2(B), \ldots\}$.  

If $\lambda \in \Lambda(B)$ then

$$(B_{1}^{\lambda}B_{1}^{\lambda} \ldots B_{1}^{\lambda})_1^n = B,$$

or

$$B = [b_1 \ldots b_\lambda]_n [b_1 \ldots b_\lambda]_n \ldots [b_1 \ldots b_\lambda]_n [b_1 \ldots b_j]_n.$$
THE FIRST RETURN TIME TEST OF PSEUDORANDOM NUMBERS

Table 1. The Expectation of $R_n$ and $\log R_n$

<table>
<thead>
<tr>
<th>Block</th>
<th>$\Lambda(B)$</th>
<th>$E[R_n]$</th>
<th>$E[\log R_n]$</th>
<th>$\text{Var} \log R_n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>1</td>
<td>256</td>
<td>4.122127</td>
<td>18.37019</td>
</tr>
<tr>
<td>00000001</td>
<td>0</td>
<td>256</td>
<td>7.299403</td>
<td>2.441935</td>
</tr>
<tr>
<td>00000010</td>
<td>7</td>
<td>256</td>
<td>7.273498</td>
<td>2.589157</td>
</tr>
<tr>
<td>00000100</td>
<td>6.7</td>
<td>256</td>
<td>7.219351</td>
<td>2.905512</td>
</tr>
<tr>
<td>00001000</td>
<td>5.6,7</td>
<td>256</td>
<td>7.106875</td>
<td>3.576236</td>
</tr>
<tr>
<td>00010001</td>
<td>4</td>
<td>256</td>
<td>7.055111</td>
<td>3.986235</td>
</tr>
<tr>
<td>00100001</td>
<td>5</td>
<td>256</td>
<td>7.183896</td>
<td>3.147559</td>
</tr>
<tr>
<td>00100010</td>
<td>4.7</td>
<td>256</td>
<td>7.031221</td>
<td>4.110117</td>
</tr>
<tr>
<td>00100100</td>
<td>3.7</td>
<td>256</td>
<td>6.717126</td>
<td>6.102838</td>
</tr>
<tr>
<td>01000001</td>
<td>6</td>
<td>256</td>
<td>7.244771</td>
<td>2.763759</td>
</tr>
<tr>
<td>01000010</td>
<td>5.7</td>
<td>256</td>
<td>7.158986</td>
<td>3.283393</td>
</tr>
<tr>
<td>01001001</td>
<td>3</td>
<td>256</td>
<td>6.738698</td>
<td>6.005312</td>
</tr>
<tr>
<td>01010101</td>
<td>2</td>
<td>256</td>
<td>6.015615</td>
<td>10.32028</td>
</tr>
</tbody>
</table>

for some $1 \leq j \leq \lambda$.

Table 2 shows $\Lambda(B)$ for some 8-blocks. For more of the definition of $\Lambda(B)$, see [1].

Lemma 2 ([1], Lemma 3). If $\lambda_1(B) \leq n/2$, then $\lambda_i(B) > n - \lambda_1(B)$ for every $i \geq 2$ if $\lambda_i(B)$ exists.

Definition 3. For a given $n$-block $B$, let $\mathcal{F}(B, k)$ be the set of $k$-blocks, $k \geq n$ defined by

$$\mathcal{F}(B, k) = \{C : C^n_1 = B, C_i^{i+n} \neq B \text{ for any } i \geq 1\},$$

and let $\mathcal{S}(B, k)$ be the set of $k$-blocks, $k \geq 1$ defined by

$$\mathcal{S}(B, k) = \{C : (CB)^n_1 = B, (CB)_{i+1}^{i+n} \neq B \text{ for any } i, 1 \leq i < k\}.$$

Example 4. Consider the case of $B = '010'$ and $k = 6$. The 6-block '010001' is not elements in $\mathcal{S}('010',6)$ but in $\mathcal{F}('010',6)$, since the 9-block '010001010' has three '010' blocks (e.g. '010001010'). Hence

$$\mathcal{F}('010',6) = \{010000, 010001, 010011, 010110, 010111\},$$

$$\mathcal{S}('010',6) = \{010000, 010011, 010110, 010111\}.$$
The following shows the relation between $\mathcal{F}(B, k)$ and $S(B, k)$. Note that for $k \geq n$, $S(B, k) \subset \mathcal{F}(B, k)$.

**Lemma 5.** For all $n$-block $B$

$$S(B, k) = \mathcal{F}(B, k) \setminus \bigcup_{\lambda \in \Lambda(B)} \{C \in \mathcal{F}(B, k) : C_{k-\lambda+1}^{k} = B_{1}^{\lambda}\},$$

where the union is disjoint union. For $\lambda \in \Lambda(B)$ and $\ell(\lambda) = \max\{j \in \mathbb{N} : j\lambda < n\}$ we have

$$\{C \in \mathcal{F}(B, k) : C_{k-\lambda+1}^{k} = B_{1}^{\lambda}\} = \bigcup_{j=1}^{\ell(\lambda)} \{C_{j} B_{j}^{\lambda} \cdots B_{1}^{\lambda} : C \in S(B, k-j\lambda)\}.$$  

**Proof.** See [1], Lemma 6. \(\square\)

Note that for any $n$-block $B$, we have $\lambda_{i}(B) > n/2$, $i \geq 2$ and $\ell = 1$ except $\lambda = \lambda_{1}(B)$.

**Definition 6.** Define $r_{k}(B)$ and $s_{k}(B)$ by

$$r_{k}(B) = \Pr(x_{1}\ldots x_{k} \in \mathcal{F}(B, k)), \quad k \geq n,$$

$$s_{k}(B) = \Pr(x_{1}\ldots x_{k} \in S(B, k)), \quad k \geq 1.$$  

From now on we consider i.i.d. processes. For i.i.d. processes we have

$$\Pr(x_{1}\ldots x_{n} = B, R_{n}(x) > k - n) = r_{k}(B), \quad k \leq n$$

$$\Pr(x_{1}\ldots x_{n} = B, R_{n}(x) = k) = s_{k}(B)\mu(B), \quad k \geq 1.$$  

We can calculate the distribution of the first return time from the followings:

**Proposition 7.** For i.i.d. processes, if $k > n$, we have

$$r_{k}(B) = r_{k-1}(B) - \mu(B)s_{k-n}(B).$$  

Let $m = |\Lambda(B)|$ and $\ell = \max\{i : i \cdot \lambda_{1}(B) < n\}$. For $k \geq n$

$$s_{k}(B) = r_{k}(B) - \sum_{i=1}^{\ell} \mu(B_{1}^{\lambda_{i}(B)})s_{k-i\lambda_{i}(B)}(B) - \sum_{i=2}^{m} \mu(B_{1}^{\lambda_{i}(B)})s_{k-\lambda_{i}(B)}(B).$$

For initial seeds, we have

$$r_{n}(B) = \mu(B)$$
THE FIRST RETURN TIME TEST OF PSEUDORANDOM NUMBERS

and for $i < n$

$$s_i(B) = \begin{cases} 
0 & \text{if } i \notin \Lambda(B), \\
\mu(B_i) & \text{if } i \in \Lambda(B). 
\end{cases}$$

Proof. For $k > n$

$$r_k(B) = \Pr(x_1^n = B, R_n(x) > k - n)$$

$$= \Pr(x_1^n = B, R_n(x) > k - n - 1) - \Pr(x_1^n = B, R_n(x) = k - n)$$

$$= r_{k-1}(B) - \mu(B)s_{k-n}(B).$$

For $k \geq n$ by Lemma 5 we have

$$s_k(B) = r_k(B) - \sum_{\lambda \in \Lambda(B)} \sum_{j=1}^{\ell(\lambda)} \Pr(x_1^{k-j\lambda} \in S(B, k-j\lambda), x_k^{k-j\lambda+1} = B_1^{\lambda} \ldots B_1^{\lambda}).$$

By Lemma 2, for $i \geq 2$, $\ell(\lambda_i) = 1$. Hence

$$s_k(B) = r_k(B) - \sum_{i=1}^{\ell} s_{k-\lambda_i(B)}(B) \mu(B_1^{\lambda_i(B)})^i - \sum_{i=2}^{m} s_{k-\lambda_i(B)}(B) \mu(B_1^{\lambda_i(B)}).$$

The computation of $s_k(B)$ for every $n$-block $B$ is necessary for the application in later sections and it is done recursively on computers. To save time we use the fact that the blocks with the same $\Lambda(B)$ have the same pattern as far as the first return time is concerned. Thus we classify all the $n$-blocks using $\Lambda(B)$ and compute $s_k$ for each block $B$ from different classes.

3. APPLICATION: TEST FOR THE PSEUDORANDOM NUMBER GENERATORS

We calculate $\Pr(R_n = k)$ for every integer $k \geq 1$ numerically by using the formula in the previous section. The averages and the standard deviations of the logarithm of the first return time are computed and the deviation of the experimental data from the theoretical prediction is used to test PRNGs. We apply the standard $Z$-test for sample mean of $\frac{1}{n} \log R_n$ of each block. In each case the sample size is 100,000. The test result is shown in Table 1. Test A is to consider the $Z$-values for each $14$-block. Since the $Z$-values among each blocks are highly correlated, we need to reduce the correlation among the
Table 2. Theoretical values for (1/2,1/2)-Bernoulli process

<table>
<thead>
<tr>
<th>$n$</th>
<th>$E[R_n]$</th>
<th>$E[\log R_n]/n$</th>
<th>$\sigma(\log R_n)/n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>4</td>
<td>0.7687192</td>
<td>0.5783033</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>0.8005356</td>
<td>0.4596296</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>0.8278062</td>
<td>0.3838731</td>
</tr>
<tr>
<td>5</td>
<td>32</td>
<td>0.8506853</td>
<td>0.3289105</td>
</tr>
<tr>
<td>6</td>
<td>64</td>
<td>0.8696175</td>
<td>0.2864550</td>
</tr>
<tr>
<td>7</td>
<td>128</td>
<td>0.8841787</td>
<td>0.2525540</td>
</tr>
<tr>
<td>8</td>
<td>256</td>
<td>0.8979582</td>
<td>0.2249535</td>
</tr>
<tr>
<td>9</td>
<td>512</td>
<td>0.9084916</td>
<td>0.2021856</td>
</tr>
<tr>
<td>10</td>
<td>1024</td>
<td>0.9172324</td>
<td>0.1832064</td>
</tr>
<tr>
<td>11</td>
<td>2048</td>
<td>0.9245484</td>
<td>0.1672354</td>
</tr>
<tr>
<td>12</td>
<td>4094</td>
<td>0.9307304</td>
<td>0.1536739</td>
</tr>
<tr>
<td>13</td>
<td>8192</td>
<td>0.9360054</td>
<td>0.1420567</td>
</tr>
<tr>
<td>14</td>
<td>16384</td>
<td>0.9405495</td>
<td>0.1320201</td>
</tr>
<tr>
<td>15</td>
<td>32768</td>
<td>0.9444992</td>
<td>0.1232780</td>
</tr>
<tr>
<td>16</td>
<td>65536</td>
<td>0.9479612</td>
<td>0.1156049</td>
</tr>
<tr>
<td>17</td>
<td>131072</td>
<td>0.9510188</td>
<td>0.1088215</td>
</tr>
</tbody>
</table>

Z-values. A binary block can be regarded as an integer in binary expansion. We say that $B$ is of

- type I if $B = (110101)_2^k + (101001)_2$,
- type II if $B = (1011001)_2^k + (111011)_2$,
- type III if $B = (1001001)_2^k + (111111)_2$,
- type IV if $B = (1100101)_2^k + (111101)_2$,

for some integer $k \geq 0$.

Test B-I, II, III, IV is the variance test of the $Z$-values over the blocks of type I, II, III, IV. The experiments show that correlations among the blocks of type I, II, III and IV are negligible. The symbols $\triangle$ and $\times$ denote the cases when the corresponding generators fail the test with statistical confidence of 95% and 99%, respectively.
4. Pseudorandom number generators

The following is a list of pseudorandom number generators tested in Section 3. We generate binary sequences using the algorithms listed in Table 4. A linear congruential generator $LCG(M,a,b)$ means the algorithm given by

$$X_{n+1} \equiv aX_n + b \pmod{M}.$$ 

Randu is an outdated generator developed by IBM in the sixties. ANSI and Microsoft are the generators used in C libraries by American National Standard Institute and Microsoft, respectively. For a prime $p$, the inversive congruential generator $ICG(p,a,b)$ is that

$$X_{n+1} \equiv a\overline{X_n} + b \pmod{p},$$

where $\overline{X}$ is the multiplicative inverse of $x$ modulo $p$. The generators Ran0, Ran1, Ran2 and Ran3 are from [11]. Ran0 is the linear congruential generator by Park and Miller[10]. Ran1 is Ran0 with Bays-Durham shuffle. Ran2 is L’Ecuyer’s generator[7] made up of

$LCG(2147483563, 40014, 0)$ and $LCG(2147483399, 40692, 0)$

with Bays-Durham shuffle. Ran3 is a subtractive lagged Fibonacci sequences. The subtract with borrow generator (SWB) is the form of $X_n \equiv X_{n-s} - X_{n-r} - b \pmod{M}$, where
### Table 4. The tested random number generators

<table>
<thead>
<tr>
<th>Name</th>
<th>Generator</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Randu</td>
<td>$LCG(2^{31}, 65539, 0)$</td>
<td>$2^{29}$</td>
</tr>
<tr>
<td>ANSI</td>
<td>$LCG(2^{31}, 1103515245, 12345)$</td>
<td>$2^{31}$</td>
</tr>
<tr>
<td>Microsoft</td>
<td>$LCG(2^{31}, 214013, 2531011)$</td>
<td>$2^{31}$</td>
</tr>
<tr>
<td>Fishman</td>
<td>$LCG(2^{31} - 1, 950706376, 0)$</td>
<td>$2^{31} - 2$</td>
</tr>
<tr>
<td>ICG</td>
<td>$ICG(2^{31} - 1, 1, 1)$</td>
<td>$2^{31} - 1$</td>
</tr>
<tr>
<td>Ran0</td>
<td>$LCG(2^{31} - 1, 16807, 0)$</td>
<td>$2^{31} - 2$</td>
</tr>
<tr>
<td>Ran1</td>
<td>Ran0 with shuffle</td>
<td>$&gt; 2^{31} - 2$</td>
</tr>
<tr>
<td>Ran2</td>
<td>L’Ecuyer’s algorithm with shuffle</td>
<td>$&gt; 2.3 \times 10^{18}$</td>
</tr>
<tr>
<td>Ran3</td>
<td>$X_n \equiv X_{n-55} - X_{n-24} \pmod{2^{31}}$</td>
<td>$\geq 2^{55} - 1$</td>
</tr>
<tr>
<td>F90</td>
<td>Ran0 combined with shift register</td>
<td>$\sim 3.1 \times 10^{18}$</td>
</tr>
<tr>
<td>SR</td>
<td>$X_n = X_{n-1}(I + L^{13})(I + R^{17})(I + L^{5})$</td>
<td>$\sim 2^{1178}$</td>
</tr>
<tr>
<td>SWB</td>
<td>$X_n \equiv X_{n-24} - X_{n-37} - b \pmod{2^{32}}$</td>
<td>$\sim 2^{1178}$</td>
</tr>
</tbody>
</table>

The borrow $b$ is $-1$ if the previous subtraction is negative and $0$ otherwise. The shift register generator (SR) is the form of $X_n \equiv X_{n-1}(I + L^r)(I + R^s)$ or $X_n \equiv X_{n-1}(I + R^f)(I + L^s)$, where $\oplus$ denotes the binary exclusive-or operation and $L$ (resp. $R$) is the bitwise left-shift (resp. right-shift). F90 is Ran0 combined with SR[12].

### References


THE FIRST RETURN TIME TEST OF PSEUDORANDOM NUMBERS


DEPARTMENT OF MATHEMATICS, KOREA ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY, TAEJON 305-701, KOREA

E-mail address: kim@euclid.kaist.ac.kr